

LUIGINO CALVI

INFORMATICA



2023

Una *grande* scoperta risolve un grande problema ma c'è un granello di scoperta nella soluzione di ogni problema.

Il tuo problema può essere modesto; ma se esso sfida la tua curiosità e mette in gioco le tue facoltà inventive, e lo risolfvi con i tuoi mezzi, puoi sperimentare la tensione e godere del trionfo della scoperta.

Questa esperienza ad una età suscettibile può creare un gusto per il lavoro mentale e lasciare un'impronta nella mente e un carattere per tutta la vita.

G. Polya, *How to solve it*

Il materiale presente in questa dispensa è utilizzabile secondo i termini della licenza Creative Commons CC BY-SA 4.0 

Indice

1	Calcolare	1
1.1	I mucchi di sassolini	2
1.2	Una calcolatrice a sassolini	6
1.3	L'abaco.	13
1.4	Macchine calcolatrici	15
1.5	Valutare con una calcolatrice	20
	Esercizi.	24
2	Contare	27
2.1	Contare senza i numeri.	28
2.2	Contare con i numeri	28
2.3	Strategie enumerative di conteggio	29
2.4	Strategie insiemistiche di conteggio	31
2.5	Strategie elementari di conteggio.	32
2.6	Strategie geometriche di conteggio	32
2.7	Strategie combinatorie di conteggio	34
2.8	Contare i percorsi.	37
2.9	Calcolo combinatorio	40
	Esercizi.	43
3	Sommare	49
3.1	Somme di numeri.	50
3.2	Algoritmi per sommare	52
3.3	Progressioni	54
3.4	Somma di tabelle di numeri	56
3.5	Somme infinite.	57
	Esercizi.	61
4	Misurare	63
4.1	Area del triangolo	64
4.2	Area dei quadrilateri	67

4.3	Area dei poligoni	68
4.4	Area del cerchio	68
4.5	Un problema di aree.	69
4.6	Scomposizione di figure	70
4.7	Efficacia delle scomposizioni	72
4.8	Calcoli geometrici	73
4.9	Calcoli algebrici	75
	Esercizi.	76
5	Disegnare	79
5.1	Una tartaruga che disegna	80
5.2	Istruire la tartaruga	81
5.3	Disegnare figure ripetitive	81
5.4	Figure ricorsive	83
5.5	Ripetere e ricorrere	88
5.6	La tartaruga si muove nel piano	90
5.7	La tartaruga come automa con stato	92
5.8	Disegno di un grafo completo	96
5.9	La tartaruga come calcolatore	98
5.10	La grafica della tartaruga in modalità ad oggetti	99
	Esercizi.	101
6	Tracciare	109
6.1	Le entità geometriche	110
6.2	Costruzioni con riga e compasso	110
6.3	Intersezione di linee.	112
6.4	Un linguaggio per le costruzioni geometriche.	114
6.5	Costruzioni geometriche	115
6.6	Numeri costruibili	122
6.7	Costruzioni con il solo compasso	126
6.7.1	Il problema del compasso molle	128
6.7.2	Il problema di Napoleone	130
	Esercizi.	132
7	Punteggiare	135
7.1	I punti e loro denominazione	136
7.2	Punti e figure	137
7.3	Il sistema geometrico di Tarski	137
7.4	Relazioni fra punti	138
7.4.1	Uguaglianza fra punti	138
7.4.2	Interposizione ed allineamento	139

7.4.3	Isometria	140
7.4.4	Parallelismo	140
7.5	Operazioni sui punti	141
7.5.1	Allineamento, parallelismo e perpendicolarità	142
7.5.2	Distanza fra punti	143
7.5.3	Angoli	144
7.5.4	Omotetie	146
7.5.5	Rotazioni	149
7.6	L'algebra dei punti	150
7.7	Proprietà delle operazioni sui punti	153
7.8	Figure	155
7.8.1	Definizione di figure	155
7.8.2	Denotazione di figure	157
7.8.3	Espressione di figure	158
7.9	Il "triangolo"	159
	Esercizi.	161
8	Metrizzare	165
8.1	Il concetto di distanza	166
8.2	Gli spazi metrici	167
8.3	Metrizzare il piano	169
8.4	Metrizzare gli spazi n-dimensionali.	171
8.5	Definizioni in uno spazio metrico	172
8.6	Distanza fra figure di uno spazio metrico	174
8.7	Criteri per definire la distanza	175
8.8	Metriche sulle stringhe.	175
8.9	La distanza di edit	177
8.10	Metriche equivalenti	179
8.11	Prodotto di spazi metrici	180
8.12	Interrogazioni negli spazi metrici	180
	Esercizi.	182
9	Approssimare	187
9.1	Vi presento la $\sqrt{2}$	188
9.2	Alla ricerca della $\sqrt{2}$	189
9.3	Calcolo della $\sqrt{2}$	190
9.4	Calcolo della \sqrt{a}	191
9.5	Frazioni continue	194
9.6	Velocità di convergenza	197
9.7	Calcoli infiniti	198
9.8	Convergenza negli spazi metrici	199

9.9	Procedimenti iterativi indefiniti	200
9.10	Criteri di arresto	201
9.11	L'errore nei processi di calcolo	202
9.12	Computabilità dei numeri	203
	Esercizi.	204
10	Formalizzare	207
10.1	I sistemi formali	208
10.2	Dimostrazioni nei sistemi formali	210
10.3	Sistemi formali e significato	211
10.4	Il sistema formale MU	214
10.5	Giochi come sistemi formali	215
10.6	Gli L-sistemi	216
10.6.1	L-sistemi per figure periodiche	218
10.6.2	L-sistemi per figure ricorsive	219
10.6.3	L-sistemi con ramificazioni	220
10.6.4	L-sistemi che generano piante	221
10.6.5	Una macchina per gli L-sistemi	222
	Esercizi.	223

1

CALCOLARE

L'abaco procura un'esperienza multisensoriale; infatti l'abacista vede muoversi le palline, le sente tintinnare quando urtano una contro l'altra e le percepisce nel loro insieme. Sicuramente non esiste un altro calcolatore digitale che abbia un'attendibilità così alta in proporzione al basso costo di acquisto e di manutenzione.

Martin Gardner, *The Abacus*

L'ideazione, la costruzione e l'uso di strumenti mediante i quali svolgere più agevolmente le proprie attività, specialmente quelle più ripetitive, è sempre stato un tratto caratteristico dell'evoluzione dell'uomo; su questa linea di sviluppo si inserisce il filone del pensiero scientifico e del progresso tecnologico relativo agli strumenti di calcolo. Metodi e strumenti per eseguire velocemente i calcoli aritmetici hanno punteggiato tutto il percorso dell'evoluzione dell'uomo, dalle civiltà antiche fino agli enormi progressi degli ultimi secoli. Se mettessimo in fila, come in un museo virtuale, tutti gli strumenti ideati ed utilizzati dall'uomo per calcolare, delineeremmo una linea che corre parallela all'evoluzione del pensiero scientifico.

Nel recente passato l'informatica è stata spesso associata al concetto di *elaborazione numerica*; gli elaboratori, chiamati per l'appunto *calcolatori*, venivano pensati come delle potenti macchine che ricevevano in ingresso numeri, li elaboravano a grande velocità e producevano numeri; anche se più recentemente hanno acquisito sempre maggiore importanza le elaborazioni di tipo non numerico, i *numeri* costituiscono ancora gli oggetti privilegiati in molti processi di elaborazione di tipo informatico. A conferma della centralità e dell'importanza del calcolare, il *calcolatore* è stato assunto come icona dell'evoluzione tecnologica che ha caratterizzato l'ultimo mezzo secolo.

1.1 I mucchi di sassolini

Il più antico strumento utilizzato dall'uomo per contare e per eseguire dei calcoli è costituito dai sassolini. Il termine *calcolo* deriva proprio dal latino *calculus* che significa sassolino. Useremo il termine *mucchio* per indicare un generico raggruppamento finito di elementi indistinguibili fra loro: sassolini, noccioline,



Figura 1.1: Un mucchio di sassolini

L'unica caratteristica di un mucchio è la numerosità degli elementi che lo compongono. Siccome opereremo su più mucchi, ci servirà denotare i vari mucchi con dei nomi identificativi; utilizzeremo dei nomi che iniziano con un carattere maiuscolo: $A, B, C, \dots, M, N, \dots$. Con il simbolo \circ denoteremo il *mucchio vuoto*, ossia il mucchio composto da nessun elemento. Con la notazione $\bullet \bullet \bullet$ oppure con $\boxed{3}$ indicheremo un mucchio composto da 3 elementi.

Un generico insieme di elementi può essere messo in corrispondenza con un mucchio, costruendo il mucchio, a partire dal mucchio vuoto, passando in rassegna ad uno ad uno gli elementi dell'insieme ed inserendo un elemento nel mucchio in corrispondenza ad ogni elemento considerato. Un mucchio diventa il rappresentante di tutti gli insiemi di oggetti composti dallo stesso numero di elementi.

Le operazioni di base sui mucchi

Per operare su un mucchio ammetteremo le seguenti operazioni di base, in modalità procedurale:

- $\text{isempty}(A)$: test se il mucchio A è vuoto
- $\text{insert}(A)$: inserimento di un elemento nel mucchio A
- $\text{remove}(A)$: eliminazione di un elemento dal mucchio A

Usando queste operazioni di base si possono realizzarne altre; queste agevoleranno, a loro volta, la definizione di operazioni più avanzate.

Esempio 1.1.1 - Per spostare un elemento da un mucchio ad un altro si può procedere come descritto nell'algoritmo 1.

Algoritmo 1 - *sposta(A, B)*

Require: mucchi A e B **Ensure:** mucchio A con 1 elemento in meno e B con 1 elemento in più

```

1: if  $\neg \text{isempty}(A)$  then
2:   remove( $A$ )
3:   insert( $B$ )
4: end if

```

Esempio 1.1.2 - Per svuotare un mucchio, ossia per eliminare tutti gli elementi presenti in esso, si può utilizzare il seguente algoritmo ciclico.

Algoritmo 2 - *svuota(A)*

Require: mucchio A **Ensure:** il mucchio A è vuoto

```

1: while  $\neg \text{isempty}(A)$  do
2:   remove( $A$ )
3: end while

```

Operazioni di spostamento fra mucchi

Consideriamo le operazioni *travasa*, *copia*, *aggiungi* descritte graficamente nella figura 1.2.

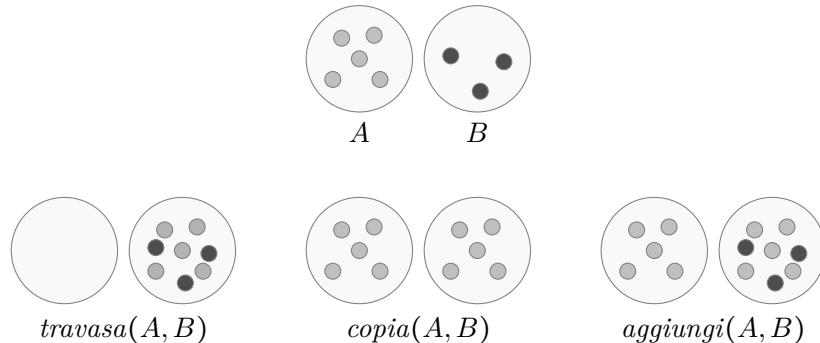


Figura 1.2: Operazioni sui mucchi: a partire dalla situazione descritta nella parte superiore della figura, ciascuna operazione indica porta alla nuova situazione descritta in corrispondenza all'operazione stessa.

Queste tre operazioni, realizzate nei tre algoritmi che seguono, costituiscono una triade che verrà utilizzata per realizzare altre operazioni più avanzate.

Esempio 1.1.3 - Trasferire tutti gli elementi da un mucchio ad un altro:

Algoritmo 3 - *travasa*(A, B)

Require: mucchi A e B

Ensure: il mucchio A è vuoto, B contiene gli elementi di entrambi i mucchi

- 1: **while** $\neg \text{isempty}(A)$ **do**
 - 2: *sposta*(A, B)
 - 3: **end while**
-

*Esempio 1.1.4 - Copiare gli elementi da un mucchio ad un altro, mantenendo inalterato il primo mucchio e senza usare la funzione *clone*:*

Algoritmo 4 - *copia*(A, B)

Require: mucchi A e B

Ensure: il mucchio A rimane inalterato, B contiene gli elementi di A

- 1: $T \leftarrow \emptyset$
 - 2: *svuota*(B)
 - 3: **while** $\neg \text{isempty}(A)$ **do**
 - 4: *insert*(B)
 - 5: *sposta*(A, T)
 - 6: **end while**
 - 7: *travasa*(T, A)
-

Esempio 1.1.5 - Aggiungere ad un mucchio gli elementi presenti in un altro:

Algoritmo 5 - *aggiungi*(A, B)

Require: mucchi A e B

Ensure: il mucchio A rimane inalterato, B contiene gli elementi di A

- 1: $T \leftarrow \emptyset$
 - 2: *copia*(A, T)
 - 3: *travasa*(T, B)
-

Problema 1.1.1 È data un mucchio A ed un mucchio B vuoto. Suddividere equamente i sassolini in modo che in entrambi questi due mucchi ci siano sassolini in un numero uguale o al più diverso di un'unità. Nel caso in cui il numero di elementi presenti in A sia dispari, alla fine A dovrà contenere un sassolino meno di B .

Soluzione. Una soluzione è data dal seguente algoritmo 6.

Algoritmo 6 - *suddividi*(A, B)

Require: mucchio A di elementi, mucchio B vuoto
Ensure: mucchi A e B con gli elementi equamente suddivisi

```

1:  $C \leftarrow \emptyset$ 
2: while  $\neg \text{isempty}(A)$  do
3:   sposta( $A, B$ )
4:   if  $\neg \text{isempty}(A)$  then
5:     sposta( $A, C$ )
6:   end if
7: end while
8: travasa( $C, A$ )

```

Operazioni prenumeriche

Anche se è abbastanza naturale identificare un mucchio di sassolini con un numero (pari al *numero* dei sassolini) ci sono molte interessanti operazioni che non richiedono di pensare ai numeri. Sfruttando le operazioni definite negli esempi precedenti risolviamo alcuni problemi.

Problema 1.1.2 Dimezzare il numero di elementi presenti in un mucchio.

Soluzione. Una soluzione è data dal seguente algoritmo 7 che si fonda direttamente sull'algoritmo *suddividi*.

Algoritmo 7 - *dimezza*(A)

Require: mucchio A
Ensure: A contiene metà degli elementi presenti inizialmente

```

1:  $T \leftarrow \emptyset$ 
2: suddividi( $A, T$ )

```

Problema 1.1.3 Stabilire se due dati mucchi contengono lo stesso numero di elementi.

Soluzione. Per evitare che il controllo sui due mucchi non abbia effetti collaterali modificando lo stato iniziale dei due mucchi, è necessario copiarli in due mucchi ausiliari per poi operare liberamente su queste copie. Una soluzione è data dal seguente algoritmo 8.

Algoritmo 8 - *uguale*(A, B)

Input: mucchi A e B di elementi**Output:** TRUE se e solo se A e B contengono un numero uguale di elementi

```

1: copia( $A, T_1$ )
2: copia( $B, T_2$ )
3: while  $\neg \text{isempty}(T_1) \wedge \neg \text{isempty}(T_2)$  do
4:   remove( $T_1$ )
5:   remove( $T_2$ )
6: end while
7: return  $\text{isempty}(T_1) \wedge \text{isempty}(T_2)$ 
```

Osservazione. Le operazioni *dimezza* e *uguale* sono delle operazioni prenumeriche che, pur richiamando nella loro etimologia delle operazioni aritmetiche (divisione per 2, controllo di uguaglianza =), sono primitive rispetto a qualsiasi interpretazione numerica in quanto si fondano solamente su concetti di comparazione e di numerosità.

1.2 Una calcolatrice a sassolini

Eseguire mediante una calcolatrice un semplice calcolo, ad esempio $4726 * 9817$, è per noi un'operazione così semplice e scontata che ci fa perdere di vista tutto il travaglio di pensiero, creatività tecnica e di sperimentazione che ha portato alla situazione così tecnologicamente avanzata dei nostri giorni. Per ripercorrere questa interessante avventura tecnica e scientifica ci proponiamo di realizzare una calcolatrice a sassolini.

Alan Turing in un famoso articolo del 1950 (*Computing Machinery and Intelligence*) propose una struttura di calcolatore numerico che venne sostanzialmente adottata nella realizzazione dei calcolatori nei decenni che seguirono. Seguendo l'analogia di come l'uomo opera a mano su carta per fare un calcolo, individuò le seguenti componenti di un calcolatore:

1. la *memoria* contenente i dati; corrisponde alla carta del calcolatore umano; la memoria viene suddivisa in sezioni ridotte, ciascuna contenente un'unità di informazione ed univocamente individuata mediante un indirizzo numerico oppure mediante un nome
2. le *operazioni* di base che il calcolatore è in grado di eseguire; queste operazioni caratterizzano ciascuna macchina: potrebbero essere le quattro operazioni aritmetiche fondamentali (addizione, sottrazione, moltiplicazione, divisione) oppure operazioni più elementari del tipo: "scrivi zero", "incrementa di uno"
3. i *controlli* che servono per descrivere la corretta sequenzialità di esecuzione delle operazioni

Seguendo lo schema proposto da Turing, in questo paragrafo realizziamo una semplice calcolatrice a sassolini in grado di eseguire le operazioni aritmetiche fondamentali sui numeri naturali.

Le componenti di una calcolatrice a sassolini

Nelle seguenti sottosezioni definiremo le componenti di base per realizzare una *calcolatrice a sassolini* in grado di eseguire calcoli.

La memoria. Utilizzeremo un mucchio di sassolini come unità fondamentale di memoria, per rappresentare un numero naturale; in pratica un mucchio di sassolini rappresenterà un numero naturale.

Le operazioni di base. Per operare su un mucchio di sassolini ammetteremo le operazioni di base viste per i mucchi.

I controlli sulle azioni. Per realizzare operazioni più articolate abbiamo bisogno di alcuni controlli per organizzare le operazioni di base descritte nella sezione precedente; a questo scopo utilizzeremo i *controlli di base* della programmazione strutturata (sequenziale, condizionale, ciclico). Nello spirito di minimalità che abbiamo intrapreso, al posto di questi controlli si potrebbe utilizzare equivalentemente i controlli di basso livello costituiti dai salti incondizionati (*goto* α) e condizionati (*if* C *goto* α).

Dai mucchi di sassolini ai numeri (e viceversa)

Per passare dal mondo (reale) dei sassolini al mondo (concettuale) dei numeri utilizzeremo le seguenti due funzioni:

$$\begin{aligned} \text{mucchio} : \text{numero } n &\rightarrow \text{mucchio composto da } n \text{ sassolini} \\ \text{numero} : \text{mucchio } M &\rightarrow \text{numero di sassolini di } M \end{aligned}$$

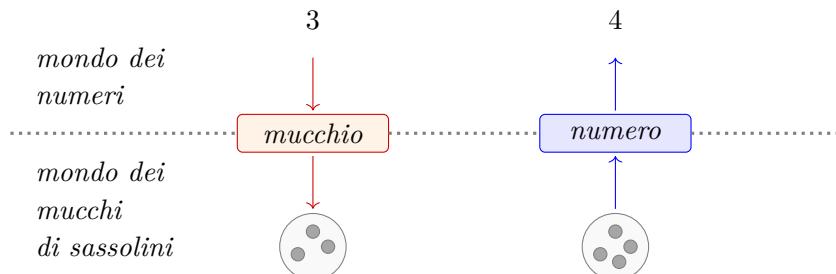


Figura 1.3: Schema delle funzioni *mucchio* e *numero*.

Le funzioni *mucchio* e *numero* costituiscono una corrispondenza fra il mondo dei numeri naturali ed il mondo dei mucchi di sassolini, facendo corrispondere ad un numero naturale n un mucchio composto da n sassolini e ad un mucchio M il numero dei sassolini che lo compongono. Questa corrispondenza può essere interpretata anche nel contesto dell'informatica e dei linguaggi di programmazione, identificando un mucchio M composto da un numero m di sassolini come una variabile n che memorizza il numero m . Queste due funzioni costituiscono delle sorta di periferiche di i/o fra i due mondi, in modo simile alle tradizionali operazioni *read* e *write* di molti linguaggi di programmazione.

Osservazione. Sullo sfondo delle funzioni *mucchio* e *numero*, per noi intuiti-ve e naturali, si cela un lungo percorso di astrazione che ha portato al concetto di numero (naturale) e che si è compiutamente concluso solo agli inizi del XX secolo quando Giuseppe Peano ha definito in modo formale i numeri naturali mediante un coerente sistema di assiomi.

Per ogni numero naturale n ed ogni mucchio M , valgono le seguenti proprietà:

$$\begin{aligned} \text{numero}(\text{mucchio}(n)) &= n \\ \text{mucchio}(\text{numero}(M)) &= M \end{aligned}$$

Indicando con $|M|$ il numero di elementi del mucchio M e con (\underline{n}) un mucchio composto da n elementi, le due precedenti proprietà possono essere scritte in modo sintetico come segue:

$$|(\underline{n})| = n \quad |(\underline{|M|})| = M$$

L’interfaccia di comunicazione fra il mondo dei numeri e quello dei mucchi può essere fondata su pochi e basici concetti corrispondentesi nei due mondi. Fra le tante possibilità una è descritta nella tabella 1.1.

Mondo dei numeri	Mondo dei mucchi di sassolini
numero <i>zero</i> , denotato con 0	mucchio vuoto \circ
numero n	mucchio (\underline{n}) composto da n sassolini
<i>successivo</i> di un numero n , $\text{succ}(n)$	operazione <i>insert</i>
<i>precedente</i> di un numero n , $\text{pred}(n)$	operazione <i>remove</i>
<i>comparazione</i> con il numero 0	operazione <i>isempty</i>
assegnazione $n \leftarrow \text{numero}$	assegnazione $M \leftarrow \text{mucchio}$

Tabella 1.1: Corrispondenza fra le operazioni di base del mondo dei numeri e del mondo dei mucchi di sassolini.

Utilizzando le funzioni di base descritte nella tabella 1.1 realizziamo ora, mediante i due algoritmi 9 e 10, le operazioni *mucchio* e *numero*.

Algoritmo 9 - $\text{numero}(M)$

Input: mucchio M

Output: numero di sassolini di M

- 1: $\text{copia}(A, T)$
 - 2: **while** $\neg \text{isempty}(T)$ **do**
 - 3: $n \leftarrow \text{succ}(n)$
 - 4: $\text{remove}(T)$
 - 5: **end while**
 - 6: **return** n
-

Algoritmo 10 - *mucchio(n)***Input:** numero n **Output:** mucchio di n sassolini

```

1:  $M \leftarrow \emptyset$ 
2:  $k \leftarrow 0$ 
3: while  $k \neq n$  do
4:    $k \leftarrow \text{succ}(k)$ 
5:    $\text{insert}(M)$ 
6: end while
7: return  $M$ 

```

Osservazione. Bisogna fare attenzione a non utilizzare le funzioni *mucchio* e *numero* in modo improprio; ad esempio non si può generare un mucchio contenente un numero di sassolini pari a quelli contenuti in due mucchi A e B mediante l'espressione

$$\text{mucchio}(\text{numero}(A) + \text{numero}(B))$$

Operazioni aritmetiche con i sassolini

Per realizzare le operazioni sui mucchi corrispondentemente a come sono definite le operazioni sui numeri, consideriamo come esempio il processo di addizione $3 + 2$ è descritto nella figura 1.4.

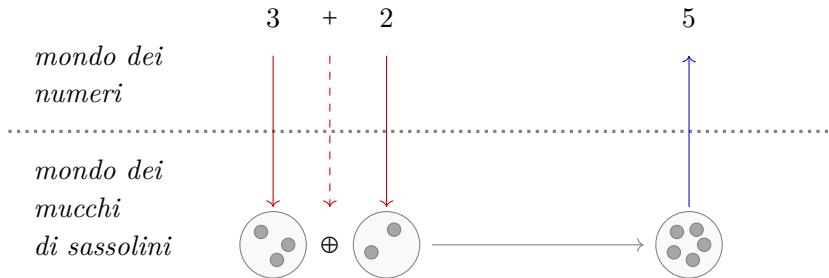


Figura 1.4: Schema del processo di calcolo $3 + 2$: le frecce verso il basso denotano la funzione *mucchio*, la freccia tratteggiata l'algoritmo *unione* di due mucchi e la freccia verso l'alto la funzione *numero*.

Se m ed n sono due numeri naturali, valgono le seguenti due relazioni:

$$m + n = \text{numero}(\text{mucchio}(m) \oplus \text{mucchio}(n))$$

$$M \oplus N = \text{mucchio}(\text{numero}(M) + \text{numero}(N))$$

Realizzeremo a seguire le operazioni fondamentali di addizione, sottrazione, moltiplicazione e divisione fra mucchi di sassolini. Queste operazioni, sfruttando anche la corrispondenza descritta alla sezione precedente, ci con-

sentiranno di realizzare altre operazioni operando disinvoltamente sui mucchi di sassolini come se stessimo operando su dei numeri naturali.

Addizione. Come primo esempio realizziamo l'operazione di addizione fra due mucchi di sassolini, ritornando come risultato un mucchio composto dalla somma dei due dati mucchi. L'impostazione funzionale impone che i due mucchi dati rimangano inalterati. Denoteremo questa operazione con la notazione

$$M \oplus N$$

Algoritmo 11 - Addizione $M \oplus N$ fra due mucchi di sassolini

Input: mucchi M, N

Output: mucchio R contenente la somma dei sassolini di M ed N

- 1: $R \leftarrow \emptyset$
 - 2: $copia(M, R)$
 - 3: $aggiungi(N, R)$
 - 4: **return** R
-

Esempio 1.2.1 - Il risultato r dell'addizione $m + n$ fra due numeri naturali può essere ottenuto direttamente mediante l'espressione

$$\text{numero}(\text{mucchio}(m) \oplus \text{mucchio}(n))$$

oppure mediante la seguente sequenza di assegnazioni:

- 1: $M \leftarrow \text{mucchio}(m)$
- 2: $N \leftarrow \text{mucchio}(n)$
- 3: $R \leftarrow M \oplus N$
- 4: $r \leftarrow \text{numero}(R)$

Sottrazione. Similmente all'addizione, l'operazione di sottrazione viene realizzata mediante il seguente algoritmo.

Algoritmo 12 - Sottrazione $M \ominus N$

Input: mucchi M, N

Output: mucchio R contenente la differenza dei sassolini di M ed N

- 1: $R \leftarrow \emptyset$
 - 2: $T \leftarrow \emptyset$
 - 3: $copia(M, R)$
 - 4: $copia(N, T)$
 - 5: **while** $\neg \text{isempty}(R) \wedge \neg \text{isempty}(T)$ **do**
 - 6: $\text{remove}(R)$
 - 7: $\text{remove}(T)$
 - 8: **end while**
 - 9: **return** R
-

Moltiplicazione. Il seguente algoritmo realizza l'operazione di moltiplicazione mediante una serie di addizioni successive:

Algoritmo 13 - Moltiplicazione $M \otimes N$

Input: mucchi M, N

Output: mucchio R contenente il prodotto dei sassolini di M ed N

```

1:  $R \leftarrow \emptyset$ 
2:  $T \leftarrow \emptyset$ 
3: copia( $N, T$ )
4: while  $\neg \text{isempty}(T)$  do
5:   aggiungi( $M, R$ )
6:   remove( $T$ )
7: end while
8: return  $R$ 
```

Divisione. In modo analogo all'operazione di moltiplicazione, l'operazione di divisione intera può essere realizzata mediante una serie di sottrazioni successive. A seguire è descritto un possibile algoritmo che realizza l'operazione di divisione:

Algoritmo 14 - divisione $M \oslash N$

Input: mucchi M, N

Output: mucchio R contenente la divisione dei sassolini fra M ed N

```

1:  $R \leftarrow \emptyset$ 
2:  $T \leftarrow \emptyset$ 
3: copia( $M, T$ )
4: insert( $T$ )
5: while  $\neg \text{isempty}(T)$  do
6:    $T \leftarrow T \ominus N$ 
7:   insert( $R$ )
8: end while
9: remove( $R$ )
10: return  $R$ 
```

L'istruzione 4:*inserisci* serve per gestire il caso in cui il mucchio T (nella condizione alla linea 5) risulti vuoto, distinguendo il caso in cui il resto della divisione fra M ed N sia 0; l'istruzione 9:*elimina* bilancia l'istruzione 4 correggendo il risultato.

Osservazione. Siamo arrivati a questo punto ad avere a disposizione le 4 operazioni fondamentali $\oplus, \ominus, \otimes, \oslash$ sui mucchi di sassolini; le operazioni *mucchio* e *numero* consentono di convertire numeri naturali in mucchi di sassolini. Abbiamo così raggiunto un livello di astrazione paragonabile a quello dei tradizionali linguaggi di programmazione e possiamo pertanto scrivere istruzioni della forma

$$x \leftarrow 7 + 4 * 3$$

con disinvolta fondata sulla consapevolezza che potranno essere eseguite operando su dei mucchi di sassolini, con una traduzione della forma

- 1: $A \leftarrow \text{mucchio}(7)$
- 2: $B \leftarrow \text{mucchio}(4)$
- 3: $C \leftarrow \text{mucchio}(3)$
- 4: $R \leftarrow A \oplus (B \otimes C)$
- 5: $r \leftarrow \text{numero}(R)$

Nuovi controlli sui mucchi di sassolini

Utilizzando le operazioni di base sui mucchi di sassolini e quelle fin qui realizzate, è possibile realizzare altri controlli, oltre a quelli di base della programmazione strutturata.

Esempio 1.2.2 - Dato un mucchio M di sassolini ed un generico blocco \mathcal{E} di comandi, vogliamo realizzare il controllo descritto nell'algoritmo 15, avente il seguente significato: *Esegui il blocco di istruzioni \mathcal{E} per un numero di volte pari alla cardinalità del mucchio M .*

Algoritmo 15 - Controllo ciclico ripetitivo

- 1: **for** $|M|$ **times**
 - 2: istruzioni \mathcal{E}
 - 3: **end for**
-

avente il seguente significato: *ripeti il blocco \mathcal{E} per un numero di volte pari al numero di sassolini di M .* Utilizzando le operazioni sui mucchi di sassolini questo controllo può essere realizzato come descritto nell'algoritmo 16.

Algoritmo 16 - Controllo *for-times*

Input: mucchio M , blocco \mathcal{E}
Ensure: il blocco \mathcal{E} viene eseguito $|M|$ volte

- 1: $T \leftarrow \emptyset$
- 2: $\text{copia}(M, T)$
- 3: **while** $\neg \text{isempty}(T)$ **do**
- 4: *esegui*(\mathcal{E})
- 5: *remove*(T)
- 6: **end while**

Utilizzando il controllo **for** descritto sopra, si può riscrivere l'operazione di moltiplicazione descritta nell'algoritmo 13 in modo più agevole ed allineato con le forme predisposte nei moderni linguaggi di programmazione (algoritmo 17). È interessante notare la forma dell'algoritmo 17 che ricalca quella di un analogo algoritmo che opera sui numeri naturali: abbiamo così raggiunto, con il nostro "calcolatore a sassolini", un livello di astrazione pari a quello di un moderno computer programmabile mediante un linguaggio di programmazione

Algoritmo 17 - Moltiplicazione $M \otimes N$

Input: mucchi M, N **Output:** mucchio R contenente il prodotto dei sassolini di M ed N

- 1: $R \leftarrow \emptyset$
 - 2: **for** $|N|$ **times**
 - 3: $R \leftarrow R \oplus M$
 - 4: **end for**
 - 5: **return** R
-

1.3 L'abaco

Un abaco, nella sua forma più comune, è una tavoletta di forma rettangolare costituita da una serie di guide (fili, scanalature, ...) parallele, che convenzionalmente indicano le unità, le decine, le centinaia e così via. Lungo ogni guida possono essere spostate delle pietruzze (dette *calcoli*, da cui il termine moderno di *calcolare*) o altri oggetti mobili per eseguire le operazioni aritmetiche.



Figura 1.5: Disegno del XV secolo che rappresenta un abaco a scacchiera, costituito da una tavola suddivisa in riquadri su cui si spostavano delle pedine.

La parola *abaco* viene dal semitico *abaq*, che significa "polvere", "sabbia"; infatti gli abaci più antichi erano tavoli ricoperti da un sottile strato di sabbia sui quali con uno stilo si segnavano i calcoli. I materiali usati per la costruzione degli abaci e la loro foggia costruttiva variano moltissimo a seconda del luogo e dell'epoca storica, però il funzionamento si basa sempre sul principio fondamentale che il valore di una configurazione di calcoli dipende dal posto che occupa, ossia dalla guida su cui è posizionata. Cioè le pietruzze su linee diverse indicano grandezze di ordine diverso, anche frazionarie. Tale principio sarà poi alla base di ogni sistema di numerazione posizionale.

Non si sa quale popolo abbia inventato questo potente strumento di calcolo; si ipotizza i babilonesi; fra gli esemplari di abaco a nostra disposizione (Maya, Egiziani, Cinesi, Romani...) i più antichi hanno più di 2000 anni.

Una delle più antiche forme di abaco fu usato dai greci e successivamente dai romani. L'abaco romano è costituito da una tavola divisa in colonne ove si pongono dei gettoni che si possono far scorrere lungo delle scanalature. Le varie colonne rappresentano le potenze del 10, ed ogni gettone postovi rappresenta una unità di quella potenza (figura 1.6).

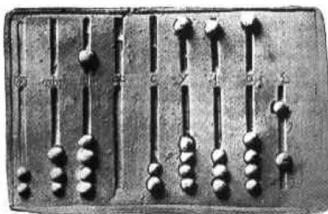


Figura 1.6: Abaco romano conservato al Cabinet des Médailles di Parigi.

L'abaco cinese, detto *suan pan*, è composto da astine sulle quali sono impilate delle perline che si possono spostare. È suddiviso in due parti separate. Ciascuna astina del suan pan ha cinque perline nella parte bassa e due perline nella parte alta. Partendo dalla destra, le perline rappresentano le unità (1, 2, 3, . . .), le successive le decine (10, 20, 30, . . .), poi le centinaia e così via. Le palline sopra l'asta valgono cinque di quelle sotto, ed ogni colonna rappresenta una potenza del dieci. Per segnare i numeri, si spostano le palline verso l'asse centrale (figura 1.7).

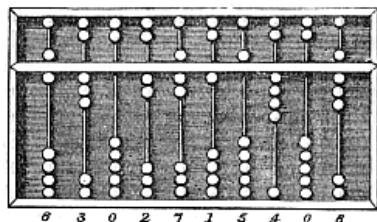


Figura 1.7: Abaco cinese in cui è rappresentato il numero 6302715408.

Una forma più evoluta dell'abaco è il *pallottoliere*, rimasto in uso efficacemente in Russia, Cina e Giappone fin dopo la II guerra mondiale. A sostegno di questa prolungata e convinta tradizione d'uso di questo strumento si narra la storia che, dopo la guerra, in uno scontro di prova fra un contabile giapponese con un pallottoliere ed uno americano con una calcolatrice, il giapponese vinse sia in velocità che in precisione.

Le prime versioni di abaco venivano utilizzate esclusivamente come strumento di conteggio. Successivamente questo strumento è stato utilizzato come supporto per eseguire dei calcoli. Le operazioni facilitate dall'uso dell'abaco non sono soltanto addizioni e sottrazioni, ma anche moltiplicazioni e divisioni, realizzate rispettivamente come addizioni e sottrazioni ripetute. Grazie ad opportune configurazioni fisiche dello strumento e ad opportune tecniche, la velocità di esecuzione dei calcoli può essere raggardevole.

Un abaco non può essere comunque assimilato ad un calcolatore nel senso moderno del termine, in quanto non dispone di meccanismi autonomi: l'operatore deve eseguire manualmente tutte le operazioni, nulla avviene in modo automatico. Il grosso limite degli strumenti come l'abaco o analoghi è costituito dal fatto che essi costituiscono solamente dei meccanismi per rap-

presentare i dati e come supporto per tenere traccia dei risultati intermedi, dei riporti e così via, ma non sono in grado di eseguire operazioni in quanto il processo di calcolo è esterno allo strumento ed è localizzato nella mente dell'uomo che manipola lo strumento.

L'abaco è stato uno degli strumenti di calcolo più utilizzati fino al Rinascimento quando venne via via abbandonato per varie ragioni; alcune di queste sono riconducibili a quanto scrisse Fibonacci nel suo *Liber Abaci*, un famoso trattato di matematica pubblicato nel 1202:

- l'avvento della notazione numerica erroneamente detta *araba*
- l'introduzione del sistema di numerazione decimale per sostituire quello greco e romano
- la diffusione di nuove tecniche e regole di calcolo anche complesso

L'abaco può essere considerato il progenitore di tutte le macchine calcolatrici meccaniche e elettromeccaniche, fino al primo colosso elettronico uscito nel 1946: l'ENIAC. Anche lui, come tutte le macchine calcolatrici precedenti funzionava sul principio dell'abaco e delle palline infilate su astine.

1.4 Macchine calcolatrici

Gli strumenti di calcolo si sono evoluti lungo un binario a due linee. La prima linea è relativa agli strumenti ed alle tecnologie impiegate per la loro costruzione, la seconda relativa ai concetti, alle idee ed ai metodi di uso di questi strumenti. Si è passati, nel corso dei millenni, dall'uso di strumenti naturali (incisioni di tacche sulle ossa di animali, uso di sassolini), alla costruzione di manufatti meccanici di legno o metallo ferro (abaci, ingranaggi con ruote dentate), a strumenti elettromeccanici, a componenti elettronici sempre più potenti e miniaturizzati. Un importante passaggio di discontinuità lungo la seconda linea corrisponde al passaggio dall'uso di strumenti come semplici supporti per la rappresentazione dei numeri alla ideazione e realizzazione di marchingegni in grado di, oltre che di rappresentare i numeri, di eseguire autonomamente delle operazioni su di essi.

Una calcolatrice numerica, nei suoi vari stadi evolutivi dall'abaco ai più moderni elaboratori, rappresenta un minimondo che replica in piccolo l'evoluzione delle tecnologie e dei metodi di calcolo, dal solo uso delle operazioni aritmetiche fondamentali alla programmabilità.

L'idea di calcolatrice di Leonardo

Leonardo da Vinci (1452-1519) fu genio eclettico indiscusso del Romanticismo italiano; si interessò e progettò macchinari di tutti i tipi che documentò nei suoi vari scritti. Fra questi, nel Codice di Madrid I, risalente all'anno 1500, Leonardo riporta dei disegni progettuali di un insieme di ingranaggi costituiti da ruote dentate per un macchinario riconducibile ad una primitiva calcolatrice meccanica (figura 1.8). L'interpretazione di questi disegni non è univoca: secondo alcuni studiosi si tratta di un semplice ingranaggio moltiplicatore, secondo altri di un marchingegno capace di eseguire calcoli.

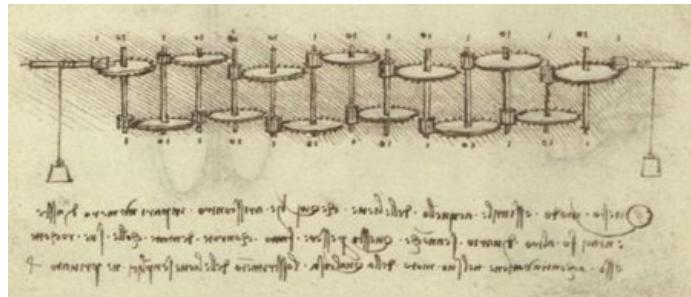


Figura 1.8: Particolare del foglio 36v del Codice di Madrid I di Leonardo.

Negli anni '60 del secolo scorso l'ingegnere italiano Roberto Guatelli (1904-1993), che all'epoca lavorava per la società di computer IBM, basandosi sugli schizzi di Leonardo, realizzò una calcolatrice meccanica; Guatelli dovette comunque adattare pesantemente il progetto di Leonardo, in particolare per risolvere i problemi di attrito; intervenne anche sulle modalità di funzionamento della calcolatrice a tal punto che molti ritengono che abbia forzato e falsato l'idea originale di Leonardo. Dopo oltre un secolo le idee di Leonardo vennero concretizzate da Blaise Pascal.

La pascalina di Pascal

La prima realizzazione di una calcolatrice digitale meccaniche è dovuta al matematico e filosofo francese Blaise Pascal (1623-1662): nel 1642, a 19 anni, con il proposito di aiutare il padre nell'esecuzione di calcoli finanziari, realizzò una calcolatrice digitale meccanica, in seguito chiamata *pascalina*. La pascalina funzionava grazie ad un sistema di ruote dentate ciascuna delle quali rappresentava una cifra decimale. Era in grado di eseguire addizioni e sottrazioni con riporto automatico fino ad otto cifre. Le addizioni venivano eseguite mediante rotazioni delle varie ruote: quando una ruota completa un giro fa avanzare di una posizione la ruota successiva realizzando l'operazione del riporto; le venivano eseguite mediante delle somme, con il metodo del complemento a 10 del secondo operando.

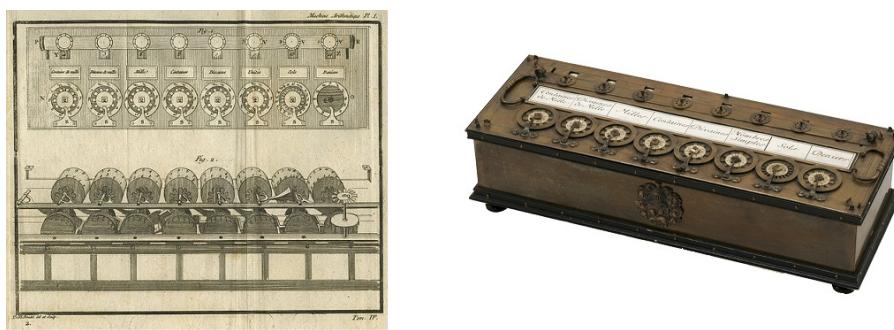


Figura 1.9: Il progetto e la macchina calcolatrice di Pascal.

La pascalina ebbe inizialmente un successo limitato in quanto, a causa degli attriti, i meccanismi si usuravano rapidamente. L'idea di Pascal venne comunque perfezionata e mantenuta in vita fino agli inizi del XX secolo quando, nel 1901, apparve sul mercato statunitense ed ebbe un grande successo di vendite una particolare calcolatrice, denominata "The Calcumeter", realizzata basandosi sull'idea della macchina calcolatrice di Pascal.

Leibniz e la sua macchina calcolatrice

Il matematico e filosofo tedesco Gottfried Wilhelm Leibniz (1646-1716) nel 1671 progettò una macchina calcolatrice che costituiva ingegnoso perfezionamento della macchina calcolatrice di Pascal: grazie a dei meccanismi a pignone con denti di lunghezza proporzionale al valore delle singole cifre era in grado di eseguire le quattro operazioni aritmetiche. In particolare la calcolatrice di Leibniz eseguiva le moltiplicazioni ripetendo delle addizioni grazie ad una tecnica innovativa che consisteva nel "memorizzare", usando un congegno detto "tamburo a denti scalati", uno dei due operandi che veniva utilizzato per eseguire delle addizioni ripetute senza doverlo riscrivere ogni volta. Si scoprì successivamente che la calcolatrice di Leibniz commetteva, in alcune situazioni, degli errori dovuti ad un difetto di un meccanismo di riporto.



Figura 1.10: La macchina calcolatrice di Leibniz.

Leibniz maturò, inoltre, l'originalissima idea, senza riuscire a concretizzarla, di eseguire le quattro operazioni aritmetiche basandosi sulla rappresentazione dei numeri secondo l'aritmetica binaria, rappresentando lo zero e l'uno mediante l'assenza o la presenza di una pallina in particolari punti del marcheggiato.

Le calcolatrici moderne

Nel 1871 l'ingegnere svedese Willgodt Odhner (1845-1905) escogitò una ingegnosa modifica al meccanismo a tamburo di Leibniz che permise di velocizzare il procedimento di addizioni ripetute necessarie per eseguire la moltiplicazione. Su questa idea migliorativa vennero realizzate calcolatrici meccaniche che ebbero un grande successo commerciale e vennero prodotte fino al 1970.

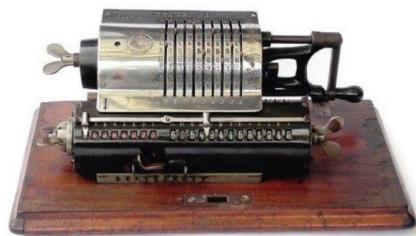


Figura 1.11: La calcolatrice "Odhner" del 1920.

A metà del secolo scorso sono stati sviluppati diversi modelli di calcolatrici, con meccanismi interni di tipo meccanico azionati da motori elettrici.



Figura 1.12: La calcolatrice elettrica Olivetti, modello "Divisumma 24" prodotta a partire del 1956: una calcolatrice completa, in grado di eseguire anche divisioni.

Le calcolatrici elettroniche

Con lo sviluppo della tecnologia elettronica, attraverso diverse fasi che vanno dall'uso di ingombranti valvole termoioniche, passando per l'uso dei transistor fino all'uso del microprocessore, le calcolatrici elettroniche presero il sopravvento e soppiantarono le calcolatrici fondate sulla tecnologia meccanica.

La calcolatrice numerica dotata delle quattro operazioni aritmetiche di addizione, sottrazione, moltiplicazione e divisione fra numeri è stata il primo strumento di calcolo di tipo digitale a diffusione di massa, a partire dall'ultimo quarto del secolo scorso. Successivamente le funzionalità di una calcolatrice sono state ampliate e, grazie alla potenza e versatilità del computer, sono state integrate in esso, diventando uno degli innumerevoli strumenti virtuali che esistono e si usano sullo schermo di un computer (figura 1.13).

Pur nella sua estrema semplicità, una calcolatrice presenta le componenti e le modalità di interazione con un utente tipiche di un sistema di elaborazione moderno:

1. l'utente digita i dati di input
2. i dati vengono memorizzati all'interno della calcolatrice
3. l'utente seleziona le operazioni da eseguire
4. la calcolatrice esegue le operazioni richieste
5. i risultati dei calcoli vengono registrati all'interno della calcolatrice
6. i risultati vengono mostrati all'utente all'esterno

Questo esecutore assolve ad una tipica funzione che si riscontra negli elaboratori: l'utente digita delle *stringhe* di caratteri e l'esecutore le interpreta come dei *numeri* che vengono registrati nella *memoria* e su di essi vengono eseguite delle operazioni. In uscita, al termine dell'elaborazione, la conversione si rovescia: i numeri calcolati dall'esecutore e registrati nella sua memoria vengono presentati all'utente sotto forma di stringhe di caratteri.

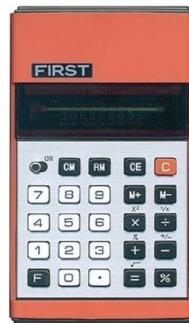


Figura 1.13: Una delle prime calcolatrici numeriche elettroniche (First, 1973).

Osservazione. Una volta realizzate, in qualche modo, delle operazioni di base su uno specifico strumento (abaco, pallottoliere, calcolatrice digitale, calcolatore a sassolini), tutto il resto è una questione di algoritmi, che non hanno più alcuna esigenza di fare riferimento alle caratteristiche, limiti e capacità dell'esecutore che viene utilizzato.

1.5 Valutare con una calcolatrice

Nella sua forma più generale, il problema della valutazione di un'espressione può essere descritto come segue:

Valutare in un ambiente A mediante un esecutore X un'espressione E descritta in notazione N.

Il problema, così impostato, risulta caratterizzato dai dati A, X, E, N . Se l'espressione E non contiene variabili, l'ambiente A risulta ininfluente ai fini della valutazione, e quindi può essere tralasciato. Se come risultato del problema si vuole un procedimento per valutare una generica espressione E in un ambiente A , rimangono come dati del problema le variabili N ed X . Così ricondotto il problema risulta interessante se si ammette che l'esecutore X abbia come capacità di base l'esecuzione di una singola operazione aritmetica elementare fra due numeri. Nel seguito utilizzeremo due modalità di esecuzione: una denominata *MEM* e l'altra *RPN*, corrispondenti a due semplicissime architetture di macchine calcolatrici che saranno descritte nei prossimi paragrafi.

Il problema si riconduce ad analizzare le 4 combinazioni che si ottengono come prodotto cartesiano dei seguenti 2 insiemi:

$$N = \{\text{notazione algebrica, notazione RPN}\}$$

$$X = \{\text{calcolatrice RPN, calcolatrice MEM}\}$$

In pratica si tratta di stabilire dei procedimenti per:

P_1 : valutare un'*espressione algebrica* mediante la calcolatrice *MEM*

P_2 : valutare un'*espressione algebrica* mediante la calcolatrice *RPN*

P_3 : valutare un'*espressione RPN* mediante la calcolatrice *MEM*

P_4 : valutare un'*espressione RPN* mediante la calcolatrice *RPN*

La soluzione di questi problemi fornisce come risultato un algoritmo.

Una *calcolatrice numerica* costituisce lo strumento specializzato per la valutazione di un'espressione; può essere utilizzata per valutare delle espressioni numeriche, come ad esempio le seguenti: $67 + 143, 561 - 51 * 3, (56 + 23) * (542 - (261 - 79))$. Se fosse lecito, come effettivamente avviene in una moderna calcolatrice numerica, digitare i caratteri componenti l'intera espressione, con eventuale uso di parentesi, ottenendo alla fine direttamente il risultato, non ci sarebbe alcun problema per valutare un'espressione. La situazione diventa interessante se la calcolatrice è in grado di eseguire una singola operazione alla volta, in base ai due operandi digitati dall'utente (per il momento analizzeremo solo il caso di operazioni aritmetiche binarie, in pratica le quattro tradizionali operazioni elementari).

Nel caso in cui si debbano valutare espressioni composte da più operazioni è necessario *memorizzare* i risultati intermedi. Consideriamo il caso in cui questo compito di memorizzazione debba essere svolto dall'esecutore stesso. Questo semplice connubio di capacità elaborative e capacità di memorizzazione richiede la definizione di una specifica architettura di elaborazione,

similmente a quanto avviene in un elaboratore moderno. Parallelamente serve che venga definito un linguaggio, coerente con l'architettura di elaborazione dell'esecutore, per descrivere le istruzioni da eseguire.

Una calcolatrice con memoria

Consideriamo una calcolatrice aritmetica con una semplicissima architettura interna di calcolo, in grado di eseguire le quattro operazioni elementari fra due numeri. Per rendere la calcolatrice più versatile la dotiamo di una memoria interna dove vengono registrati i risultati dei calcoli svolti. Una delle forme di memorizzazione che l'uomo usa spesso nella vita quotidiana è quella a pila: stiamo leggendo un libro, suona un amico alla porta ed andiamo ad aprire; mentre stiamo conversando con il nostro amico squilla il telefono e ci precipitiamo a rispondere ... ed alla fine completiamo la lettura del libro al quale avevamo dato la nostra attenzione per *primo*. Anche nella tecnologia e nella programmazione una delle forme più utili di struttura della memoria è organizzata a pila (o stack): i dati (risultati dei calcoli) vengono memorizzati e prelevati con una tecnica LIFO (*Last In - First Out*), ossia l'ultimo valore inserito è il primo ad essere prelevato. Con la calcolatrice descritta, l'utente può gestire direttamente la memoria, accedendo ed estraendo mediante il tasto-funzione M , il valore di testa della pila di memoria, ossia il valore immesso per ultimo nella memoria. L'utente può gestire direttamente la memoria, accedendo ed estraendo mediante M il valore di testa dello stack, ed ha la possibilità di riferirsi, in un'istruzione, ad un risultato ottenuto in una precedente istruzione; in questo modo l'utente (in veste di solutore) può svolgere un calcolo scrivendo dei *programmi*, costituiti da *sequenze di espressioni* che vengono intese come *istruzioni* in quanto vengono interpretate dalla calcolatrice come comando di calcolo con automatico inserimento nella pila di memoria del risultato ottenuto.

Esempio 1.5.1 - Dopo l'esecuzione della sequenza di istruzioni di calcolo $6 + 12$, $78 - 7$, $100/25$ e $6 * 8$ la situazione di memoria è la seguente:

18	71	4	48	\leftrightarrow
----	----	---	----	-------------------

In questa situazione di memoria, eseguendo le istruzioni $M + 1$ e $M - M$ si modifica come descritto nella figura che segue:

18	71	45		
----	----	----	--	--

Esempio 1.5.2 - L'espressione $(6+2)/(7-3)$ viene valutata mediante la sequenza di istruzioni descritta nella figura 1.14. Si noti l'esigenza di valutare prima il secondo operando $7-3$ in quanto l'operatore / di divisione non è commutativo e l'accesso alla memoria viene gestito con la tecnica *FIFO*.

$7 - 3$	<table border="1"><tr><td>4</td><td></td><td></td><td></td><td></td></tr></table>	4				
4						
$6 + 2$	<table border="1"><tr><td>4</td><td>8</td><td></td><td></td><td></td></tr></table>	4	8			
4	8					
M/M	<table border="1"><tr><td>2</td><td></td><td></td><td></td><td></td></tr></table>	2				
2						

Figura 1.14: Processo di calcolo relativo alla valutazione dell'espressione $(6 + 2)/(7 - 3)$ mediante una calcolatrice con memoria a pila.

Esempio 1.5.3 - Consideriamo l'espressione

$$7 - (9 - 3)/(8 - 3 * 2) + 5$$

Questa espressione può essere valutata mediante la sequenza di operazioni elementari descritta nella figura 1.15. Come si può notare, con questa architettura di calcolatrice, la valutazione di questa espressione richiede che il solutore adotti una precisa strategia di calcolo, effettuando preventivamente un'analisi della struttura dell'espressione da valutare per individuare gli operandi costituiti dalle sottoespressioni fra parentesi e per gestire la diversa priorità della valutazione degli operatori additivi rispetto a quelli moltiplicativi.

$3 * 2$	<table border="1"><tr><td>6</td><td></td><td></td><td></td><td></td></tr></table>	6				
6						
$8 - M$	<table border="1"><tr><td>2</td><td></td><td></td><td></td><td></td></tr></table>	2				
2						
$9 - 3$	<table border="1"><tr><td>2</td><td>6</td><td></td><td></td><td></td></tr></table>	2	6			
2	6					
M/M	<table border="1"><tr><td>3</td><td></td><td></td><td></td><td></td></tr></table>	3				
3						
$7 - M$	<table border="1"><tr><td>4</td><td></td><td></td><td></td><td></td></tr></table>	4				
4						
$M + 5$	<table border="1"><tr><td>9</td><td></td><td></td><td></td><td></td></tr></table>	9				
9						

Figura 1.15: Processo di calcolo relativo alla valutazione dell'espressione $7 - (9 - 3)/(8 - 3 * 2) + 5$ mediante una calcolatrice con memoria a pila.

Una calcolatrice RPN

Una calcolatrice RPN accetta in ingresso una sequenza ben formata di elementi, detti *token*, costituiti da *numeri* o da *operatori* che costituiscono un'espressione in notazione RPN. Nella situazione più semplice gli operatori sono binari e sono le tradizionali operazioni aritmetiche $+, -, *, /$. Ricevuta in ingresso la sequenza dei token la calcolatrice fornisce in uscita il risultato.

Risulta interessante analizzare la struttura interna e la modalità di valutazione di una calcolatrice RPN. La calcolatrice è dotata di una memoria a

pila nella quale vengono inseriti gli operandi in sequenza; scandisce sequenzialmente i token dell'espressione e li elabora usando la memoria della pila, seguendo l'algoritmo 18; quando incontra un operatore \circ preleva dalla pila due operandi y e x , ne viene fatta l'operazione $x \circ y$ in base all'operatore incontrato e viene inserito nella pila il risultato; una volta terminata l'elaborazione della sequenza dei token nella pila rimane un unico valore che costituisce il risultato della valutazione dell'intera espressione.

Algoritmo 18 - Valutazione di un'espressione RPN

Input: espressione aritmetica e espressa in notazione RPN

Output: valore dell'espressione e

- ```

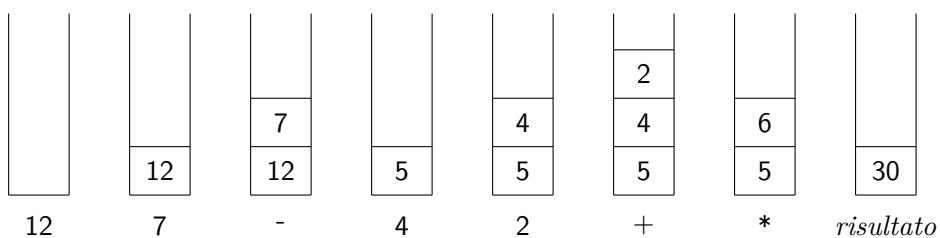
1: inizializza una pila vuota p
2: inizia dal primo token di e
3: while la sequenza dei token non è finita do
4: considera il token corrente x di e
5: if x è un numero then
6: inserisci x nella pila p
7: else
8: preleva un numero y dalla pila p
9: preleva un numero x dalla pila p
10: esegui l'operazione $x \circ y$
11: inserisci il risultato nella pila p
12: end if
13: considera il successivo token di e
14: end while
15: return elemento di testa della pila p

```

**Esempio 1.5.4 -** La figura che segue descrive la situazione della pila nelle varie fasi della valutazione dell'espressione RPN

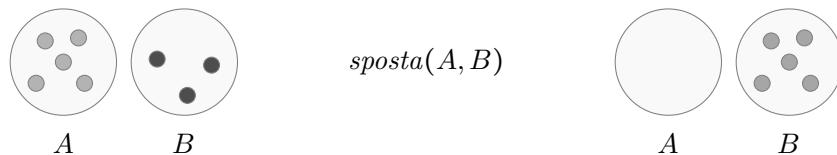
12 7 - 4 2 + \*

mostrando l'evoluzione della pila nei vari passi della valutazione, dalla situazione iniziale di pila vuota alla situazione finale in cui l'unico numero presente nella pila rappresenta il risultato della valutazione.



## ESERCIZI

**1.1** Spostare gli elementi da un mucchio ad un altro, come descritto dalla figura che segue.



**1.2** Usando le operazioni di base sui mucchi di sassolini (*isempty*, *insert*, *remove*):

1. Eguagliare due mucchi di sassolini, aggiungendo sassolini al più piccolo in modo da uguagliare quello più grande.
2. Suddividere equamente gli elementi contenuti in due mucchi, spostando alcuni elementi da uno all'altro, ottenendo che i due mucchi differiscono di al più di un elemento.
3. Eliminare degli elementi in modo da uguagliare due mucchi di elementi al meno numeroso dei due.
4. Inserire degli elementi in modo da uguagliare due mucchi di elementi al più numeroso dei due.
5. Raddoppiare il numero di elementi presenti in un mucchio.
6. Dimezzare il numero di elementi presenti in un mucchio.
7. Stabilire se un mucchio contiene un numero pari di elementi.
8. Raddoppiare gli elementi presenti in un mucchio.
9. Realizzare le operazioni di confronto fra due mucchi.
10. Eliminare da un mucchio i  $\frac{3}{4}$  degli elementi presenti.
11. Stabilire se gli elementi di un mucchio possono essere disposti a formare un quadrato.

**1.3** Spiegare perché l'operazione il risultato dell'operazione  $\otimes$  descritta nell'algoritmo 13 non può essere ottenuto semplicemente con l'espressione

$$\text{mucchio}(\text{numero}(M) * \text{numero}(N))$$

**1.4** Calcolare con le operazioni sui mucchi la parte intera del  $\log_2$  del numero di elementi presenti in un mucchio.

**1.5** Adottando le usuali convenzioni sulle precedenze e sulle associatività, valutare le seguenti espressioni:

1.  $15 + 7 - 8$
2.  $7 - 3 - 2$
3.  $16/8/4$

**1.6** Usando le calcolatrici *MEM*, *RPN*, *ALG*, valutare le seguenti espressioni:

1.  $47 - 12 + 27 + 15$
2.  $19 - 5 * 3$
3.  $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8$
4.  $64/(32/(16/(8/(4/2))))$
5.  $((14 - 9) * 6 - 13) * (19 - 12)$

**1.7** Trasformare in notazione RPN le espressioni riportate nel precedente esercizio e valutare con la calcolatrice RPN, descrivendo la situazione della pila di memoria ad ogni passo della valutazione.

**1.8** Scrivere in notazione funzionale ed in notazione RPN le seguenti espressioni aritmetiche:

1.  $(7 - 4) * 5 - (6 + 3 + 2)$
2.  $9 - 1 - 3 * 2 - (6 + 2 - 4)$

Valutare le espressioni RPN descrivendo lo stato della pila di valutazione subito dopo aver eseguito ciascuna operazione.

**1.9** Definire un'architettura di elaborazione, un linguaggio di comando ed un'interfaccia grafica per una calcolatrice in grado di gestire operatori unari, registri di memoria e l'uso di variabili. Utilizzando la calcolatrice definita scrivere un programma per calcolare la lunghezza della diagonale di un parallelepipedo di dimensioni  $a, b, c$ .



## 2

---

### CONTARE

---

*Tre pescatori di Livorno  
disputarono un anno e un giorno  
per stabilire e sentenziare  
quanti pesci ci sono nel mare.  
Disse il primo: "Ce n'è più di sette,  
senza contare le acciughette".  
Disse il secondo: "Ce n'è più di mille,  
senza contare scampi ed anguille".  
Il terzo disse: "Più di un milione!".  
E tutti e tre avevano ragione.*

Gianni Rodari,  
*Filastrocche in cielo e in terra*

*Contare* è una delle prime attività matematiche che impariamo a svolgere da piccoli e che l'uomo ha esercitato fin dalla preistoria. Nonostante l'apparente semplicità, questa azione nasconde fra le sue pieghe evolutive diversi aspetti e gradi di concettualizzazione a vari livelli. Al processo di contare è strettamente legato il concetto di *numero naturale* in quanto i numeri naturali sono stati ideati proprio per contare e l'operazione di contare utilizza i numeri naturali. Per sottolineare questo legame, al posto del termine "contare" si usa il termine "determinare il numero". Le *operazioni* sono state concepite ed utilizzate successivamente, anche come strumento a supporto del contare.

Nonostante l'apparente semplicità i problemi di conteggio sono profondi, interessanti e sollecitano importanti strategie di *problem solving*. Contare, anche nel caso di insiemi molto numerosi, non è un atto di forza ma richiede l'ideazione di strategie, talvolta raffinate e creative; poi, in definitiva, il contare si riduce all'applicazione di poche operazioni elementari.

## 2.1 Contare senza i numeri

Gli antropologi sono concordi nel ritenere che l'uomo abbia elaborato la *capacità di contare* ben prima del concetto di *numero*. Si ritiene che la necessità di effettuare un qualche tipo di conteggio si sia affermata con l'evolversi di alcune attività umane, come l'allevamento di animali: ad esempio la necessità di verificare che un gregge portato al pascolo rientrasse al completo poteva essere risolto segnando su un bastone una tacca per ogni pecora uscita; al rientro bastava scorrere le tacche del bastone, una per ogni pecora che rientrava, e verificare così di averne *tante quante* erano all'uscita; questo metodo non permette di stabilire *quante* erano le pecore ma risolveva il problema. Questa pratica dell'intaglio è stata in uso presso popolazioni di pastori europei fino in tempi relativamente recenti.

Le prime tracce documentate di qualcosa che si suppone sia un conteggio risalgono a circa 35.000 anni fa; si tratta di ossa di babbuino intagliate con tacche, rinvenute ad Ishango, nel centro dell'Africa, che si presume indichino un qualche tipo di conteggio (forse i giorni dei mesi di un calendario delle fasi lunari).



Figura 2.1: Schema dei segni incisi sull'osso di Ishango.

Negli esempi presentati sopra viene stabilita una *corrispondenza biunivoca* fra oggetti e tacche; e proprio questa corrispondenza sta alla base del *contare*. A questo stadio non c'è ancora il concetto astratto di *numero*, non ci sono le parole per indicare i singoli numeri, né tanto meno dei simboli; c'è solo l'atto pratico del mettere in corrispondenza biunivoca due insiemi.

## 2.2 Contare con i numeri

Gradualmente nel corso della sua evoluzione, l'Uomo ha concepito il concetto di *numero* e *contare* gli elementi di un dato insieme ha assunto il significato di stabilire una corrispondenza fra l'insieme ed un numero naturale in modo tale che il numero rappresenti la numerosità dell'insieme. Nella sua formulazione primordiale il procedimento del contare risolveva un problema della forma "tanti quanti", come quando l'uomo primitivo contava le sue pecore, che al mattino uscivano dal recinto al quale rientravano alla sera, mediante delle tacche incise su un osso di animale o su un bastone. Solo successivamente, con la maturazione del concetto di numero (naturale), il problema si configurò nella forma in cui lo consideriamo oggi, ossia risolvere un problema della forma "quanti sono" che comportava la corrispondenza fra la numerosità di un

insieme di oggetti ed un numero naturale. Non si tratta più della corrispondenza biunivoca fra oggetti di due insiemi (di tacche e di pecore) ma della corrispondenza fra un insieme di oggetti ed il nuovo concetto di numero che fa da termine di paragone fra due insiemi (insieme di pecore uscite dal recinto ed insieme di pecore rientrate).

Quando da piccoli ci hanno insegnato a *recitare* "uno, due, tre, quattro, ..." non avevamo ancora imparato a contare; era soltanto una filastrocca, niente di più che la tiritera "ambarabà cicci coccò tre civette sul comò ...", dove, almeno per noi, avevano qualche significato i termini "civette" e "comò". Quando, poi, la maestra all'asilo ci ha insegnato che il *segno* 1 corrispondeva ad *uno*, che il *segno* 2 corrispondeva a *due* e così via, non avevamo ancora imparato a contare ma semplicemente *denotavamo* dei segni con delle parole. Quando, messi davanti a tre matite, riuscivamo in corrispondenza di ciascuna a recitare "uno-due-tre" stavamo *enumerando* degli oggetti. Solo quando davanti a tre caramelle, alla domanda *Quante sono?* rispondevamo "tre" o scrivevamo 3 sul quaderno, allora avevamo veramente imparato a *contare*. Quando, posti davanti a due mele e tre pere, alla domanda: *Quanti sono i frutti?* scrivevamo sul quaderno  $2+3$  e, in base ad una qualche forma di elaborazione mentale, completavamo la scrittura con  $2+3=5$  avevamo imparato a *codificare* ed a *fare delle operazioni*.

### 2.3 Strategie enumerative di conteggio

Le strategie per contare gli elementi di un insieme dipendono dalla struttura e dalle caratteristiche dell'insieme stesso. Nel caso in cui l'insieme  $A$  non abbia alcuna struttura è necessario passare in rassegna, in qualche modo, gli elementi, conteggiandoli man mano che vengono considerati. In questi procedimenti è sufficiente sfruttare solamente l'"essenza" dei numeri naturali costituita attorno al concetto di "zero" e di "successivo", senza ricorrere ad alcuna altra operazione.

*Esempio 2.3.1* - Se prendiamo in mano un solido a forma di poliedro e vogliamo contarne le facce bisogna segnare quelle già considerate, per non considerarle più volte, ed essere sicuri, alla fine, di averle considerate tutte.

Se gli elementi da contare sono disposti uno dopo l'altro, in fila o anche in ordine circolare, la strategia di conteggio diventa più agevole in quanto la scansione di tutti gli elementi si fonda sul passaggio da un elemento al successivo.

*Esempio 2.3.2* - Per contare i denti di una ruota dentata basta partire da un dente, marcarlo in qualche modo e procedere al conteggio sequenziale degli elementi fino a che si ritorna sul dente dal quale si è partiti.

Spesso gli elementi dell'insieme  $A$  sono disposti linearmente, uno dopo l'altro, ed esiste un *primo* ed un *ultimo* elemento. In queste situazioni, che sono le più frequenti, la strategia di conteggio può essere descritta mediante l'algoritmo 1 che si fonda sul criterio con cui si *passa al successivo elemento*.

---

**Algoritmo 1** - Algoritmo di conteggio
 

---

**Input:** insieme  $A$  di elementi  
**Output:** numero di elementi di  $A$

- 1: considera il primo elemento di  $A$
- 2:  $n \leftarrow 0$
- 3: **while** non si sono esaminati tutti gli elementi di  $A$  **do**
- 4:    $n \leftarrow n + 1$
- 5:   passa al successivo elemento di  $A$
- 6: **end while**
- 7: **return**  $n$

---

**Esempio 2.3.3** - Per contare il numero di giorni che intercorrono fra due date è sufficiente disporre della sola operazione che determina il successivo di un dato giorno.

L'algoritmo 1 ammette una interessante generalizzazione per contare gli elementi soddisfacenti ad una data proprietà, modificando la linea 4 con un'istruzione di controllo per testare se il dato elemento è soddisfacente.

**Esempio 2.3.4** - Per contare i numeri primi minori o uguali ad un numero naturale  $n$  si può adattare l'algoritmo 1 ottenendo il seguente algoritmo 2.

---

**Algoritmo 2** - Conteggio di numeri primi
 

---

**Input:** numero naturale  $n$   
**Output:** numero di numeri primi minori o uguali a  $n$

- 1:  $p \leftarrow 0$
- 2:  $k \leftarrow 1$
- 3: **while**  $k \leq n$  **do**
- 4:   **if**  $k$  è primo **then**
- 5:      $p \leftarrow p + 1$
- 6:   **end if**
- 7:    $k \leftarrow k + 1$
- 8: **end while**
- 9: **return**  $p$

---

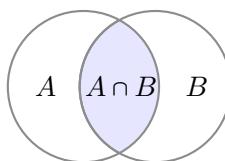
## 2.4 Strategie insiemistiche di conteggio

Alcune situazioni di conteggio possono essere risolte ragionando in termini di insiemi e delle loro cardinalità. Si fa ricorso al seguente

*Principio fondamentale del contare.* Dati due insiemi  $A$  e  $B$ , indicando con  $|\cdot|$  l'operazione fornisce la cardinalità di un insieme, vale la relazione

$$|A \cup B| = |A| + |B| - |A \cap B|$$

La proprietà espressa dalla precedente relazione può essere giustificata basandosi sulla notazione degli insiemi mediante diagrammi di Venn, come descritto nella figura che segue:

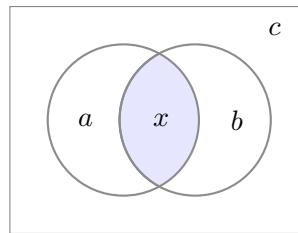


**Problema 2.4.1** In una scuola tutti gli studenti studiano almeno una lingua fra inglese o francese. Sapendo che  $a$  studenti studiano l'inglese,  $b$  studiano il francese e  $c$  studiano due lingue, determinare il numero di studenti che frequentano la scuola.

*Soluzione.* Applicando il principio fondamentale del contare si può dedurre che il numero di studenti che frequentano la scuola è  $a + b - c$ .  $\square$

**Problema 2.4.2** In un gruppo di  $n$  studenti  $a$  praticano l'atletica,  $b$  il basket e  $c$  studenti non praticano alcuno sport. Determinare il numero di studenti che praticano sia atletica che basket.

*Soluzione.* Indicando con  $x$  il numero da trovare, la situazione può essere descritta mediante il seguente diagramma:



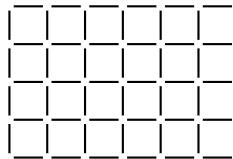
Basandosi su questo diagramma si deduce l'equazione  $n = a + b - x + c$  dalla quale si ricava  $x = a + b + c - n$ .  $\square$

## 2.5 Strategie elementari di conteggio

Spesso l'insieme di cui si vogliono contare gli elementi possiede una struttura organizzativa o operatoria che permette di attivare delle strategie di conteggio degli elementi, senza passare in rassegna uno per uno tutti gli elementi dell'insieme. In molti casi è sufficiente utilizzare le operazioni aritmetiche elementari.

*Esempio 2.5.1* - Per contare il numero dei numeri compresi fra due numeri naturali  $m$  ed  $n$ , ipotizzando che sia  $m \leq n$ , si ricorre all'operazione di sottrazione; il risultato è  $n - m + 1$ .

**Problema 2.5.1** Determinare il numero di segmenti unitari necessari per costruire un reticolo rettangolare di  $m \times n$  quadrati per lato, come descritto nella figura che segue:

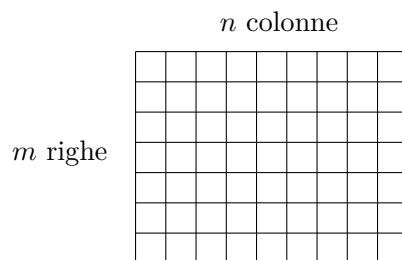


**Soluzione.** Il problema risulta di facile soluzione se si considerano separatamente le linee orizzontali e le linee verticali. La soluzione discende direttamente dalle seguenti considerazioni: ci sono  $m + 1$  segmenti orizzontali di  $n$  tratti ed  $n + 1$  segmenti verticali di  $m$  tratti. Pertanto il numero totale di segmenti unitari è pari a:  $n(m + 1) + m(n + 1) = 2mn + m + n$ .  $\square$

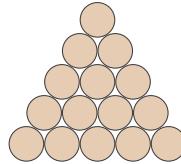
## 2.6 Strategie geometriche di conteggio

Qualora gli elementi da contare presentino direttamente o possano essere ricondotti a degli schemi geometrici, si possono adottare delle strategie che sfruttano la particolare disposizione geometrica degli elementi. Nel caso in cui gli elementi siano disposti in una configurazione piana, il conteggio si basa su considerazioni relative ad aree e conseguentemente comporta l'uso dell'operazione di moltiplicazione.

*Esempio 2.6.1* - Per contare il numero di quadretti di un foglio a quadretti si ricorre alla moltiplicazione fra il numero  $m$  di righe ed il numero  $n$  di colonne.



**Problema 2.6.1** Contare il numero di monete disposte in  $n$  file orizzontali in un triangolo, come descritto nella figura che segue, corrispondente al caso  $n = 5$ .



**Soluzione.** Un possibile approccio al problema consiste nel notare che in ogni fila del triangolo sono disposte monete in numero progressivo, da 1 a  $n$ ; il numero di monete è dunque pari a

$$1 + 2 + \cdots + n$$

Questa linea di azione porta dritta all'algoritmo 3.

---

**Algoritmo 3** - Somma dei primi  $n$  numeri naturali

---

**Input:** numero  $n$  di addendi  
**Output:** somma dei numeri  $1 + 2 + \cdots + n$

```

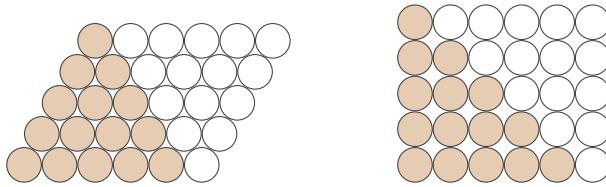
1: $s \leftarrow 0$
2: $k \leftarrow 1$
3: for n times
4: $s \leftarrow s + k$
5: $k \leftarrow k + 1$
6: end for
7: return s
```

---

Pur corretta, la soluzione presentata nell'algoritmo 3 risulta poco efficiente e ciò discende dal fatto di aver trasformato troppo precocemente il problema nel campo aritmetico, perdendo l'utile informazione geometrica della disposizione triangolare delle monete. Torniamo al punto di partenza, cercando un'altra strada. Data la particolare configurazione geometrica delle monete, un embrione di strategia consiste nel pensare a monete di grandezza unitaria e di adottare l'equivalenza fra le seguenti due espressioni:

$$\text{numero di monete} \equiv \text{area del triangolo di monete}$$

L'idea sembra promettente ma perde di forza quando si scontra con il seguente scoglio: Quali sono e quanto valgono la base e l'altezza del triangolo? La situazione si chiarisce se affianchiamo un altro triangolo rovesciato di monete in modo da ottenere un parallelogramma di monete, come evidenziato nella parte sinistra della figura che segue. A questo punto trasformiamo il parallelogramma in un rettangolo, come indicato a destra della stessa figura.



Si ottiene un rettangolo di lati composti da  $n$  e da  $n+1$  monete. Di conseguenza il rettangolo è composto da  $n(n+1)$  monete. Si ricava così che il numero di monete scure del triangolo è uguale a  $\frac{n(n+1)}{2}$ . Questa formula condensa in una sola espressione l'algoritmo 3 e risulta più efficiente in quanto richiede l'esecuzione di sole 3 operazioni rispetto alle  $2n$  addizioni richieste dall'algoritmo.  $\square$

*Osservazione.* Nella soluzione del problema 2.6.1 abbiamo ottenuto come risultato collaterale l'espressione che fornisce la somma dei primi  $n$  numeri naturali:

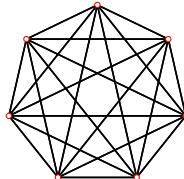
$$1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

Questa relazione è nota anche come *formula di Gauss*.

## 2.7 Strategie combinatorie di conteggio

Molti problemi di conteggio vengono risolti ricorrendo a strategie di conteggio di tipo combinatorio, basate su schemi moltiplicativi o ricorsivi.

**Problema 2.7.1** Nella figura che segue è riportato un *grafo completo* di 7 vertici, generato unendo con dei segmenti, detti *lati*, in tutti i modi possibili, i vertici di un poligono regolare. Determinare il numero dei lati, per un generico numero  $n$  di vertici del poligono.



*Soluzione.* Il problema può essere risolto adottando diverse strategie. La prima strategia consiste nel pensare ai vertici come a dei punti disposti linearmente in fila; in questo caso ogni vertice va congiunto con tutti gli altri che seguono, in ordinamento lineare: il primo vertice viene congiunto a  $n-1$  vertici, il secondo a  $n-2$  vertici, fino ad arrivare al penultimo vertice che va congiunto all'ultimo. Pertanto il numero totale dei lati è pari a

$$(n-1) + (n-2) + \dots + 1 = \frac{n(n-1)}{2}$$

dove nell'ultimo passaggio si fa ricorso alla formula di Gauss. In questo caso il "contare" si avvale del "sommare".

Una soluzione più immediata si basa sui seguenti due passi sequenziali:

1. ogni vertice ( $n$ ) va connesso a tutti gli altri ( $n - 1$ ); quindi  $n(n - 1)$  collegamenti
2. dividere per 2 in quanto ciascun collegamento viene contato due volte

Si ottiene ancora che il numero totale dei lati è pari a  $\frac{n(n-1)}{2}$ .

**Problema 2.7.2** Contare il numero di regioni  $r$  individuate da  $n$  rette a due a due incidenti e non passanti in tre per uno stesso punto. La figura che segue illustra i casi per i primi quattro valori di  $n$ .

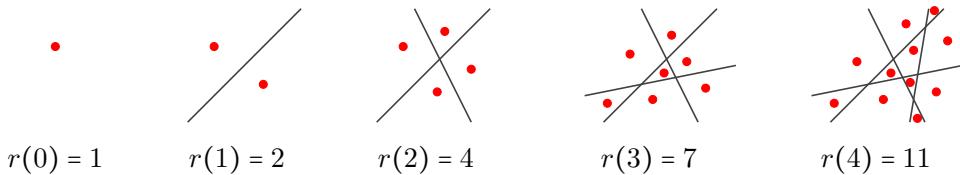


Figura 2.2: Numero di regioni  $r(n)$  determinate da  $n$  rette incidenti a due a due.

**Soluzione.** Dall'analisi della casistica descritta nella figura non risulta evidente alcuna relazione che leggi il numero  $n$  delle rette al numero  $r$  delle regioni. L'idea risolutrice si basa sulla seguente osservazione: ogni nuova  $n$ -esima retta incontra tutte le altre  $n - 1$  rette ed incrementa il numero di regioni di un numero uguale al numero di rette al passo precedente più  $n$ , come descritto nella figura 2.3.

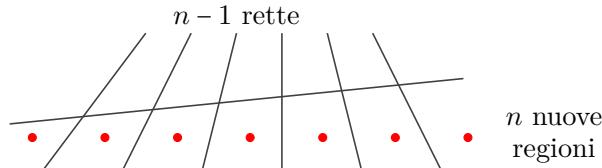


Figura 2.3: L'intersezione fra  $n-1$  rette ed una trasversale genera altre  $n$  nuove regioni.

Questa idea può essere tradotta nell'algoritmo 4.

---

**Algoritmo 4** - Numero di regioni individuate da  $n$  rette intersecantesi

---

**Input:** numero  $n$  di rette intersecantesi

**Output:** numero  $r$  di regioni generate

```

1: r \leftarrow 1
2: for i from 1 to n do
3: $r \leftarrow r + i$
4: end for
5: return r

```

---

L'idea precedente può essere tradotta anche con la seguente relazione ricorsiva:

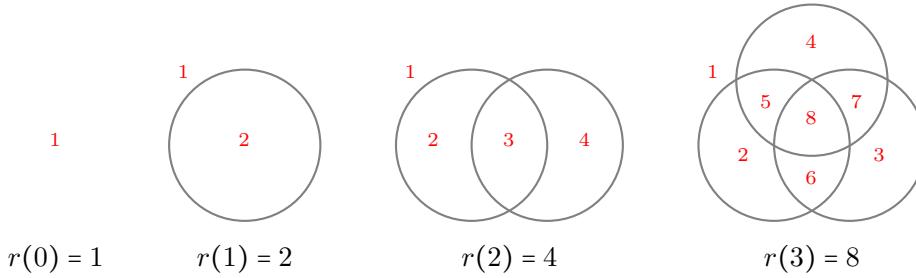
$$r(n) = n + r(n - 1)$$

Da questa relazione si può ricavare una formula chiusa come segue:

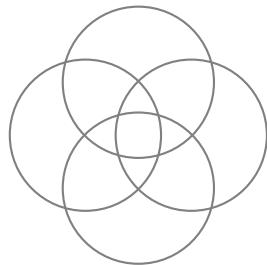
$$\begin{aligned} r(n) &= n + r(n - 1) \\ &= n + (n - 1) + r(n - 2) \\ &= \dots \\ &= n + (n - 1) + \dots + 2 + r(1) \\ &= n + (n - 1) + \dots + 2 + 1 + r(0) \\ &= \frac{n(n + 1)}{2} + 1 \end{aligned}$$

□

**Problema 2.7.3** Determinare il numero di regioni del piano che vengono determinate da  $n$  circoli ciascuno dei quali interseca tutti gli altri  $n - 1$ .



**Soluzione.** I casi illustrati nella precedente figura lasciano presupporre una relazione della forma  $r(n) = 2^n$ . Questa congettura viene però smentita dal caso  $n = 4$  illustrato nella figura che segue, per il quale si ha  $r(4) = 14$ .



Il ragionamento che risolve il problema è il seguente: l' $n$ -esimo circolo interseca i precedenti  $n - 1$  circoli in 2 punti ed ogni volta che interseca un altro circolo genera una nuova regione; pertanto l' $n$ -esimo circolo genera  $2(n - 1)$  nuove regioni. Quindi, sviluppando telescopicamente i calcoli, per  $n \geq 1$ , si ottiene:

$$\begin{aligned}
r(n) &= 2(n-1) + r(n-1) \\
&= 2(n-1) + 2(n-2) + r(n-2) \\
&= \dots \\
&= 2(n-1) + 2(n-2) + \dots + 2 \cdot 3 + 2 \cdot 2 + 2 \cdot 1 + 2 \\
&= 2[(n-1) + (n-2) + \dots + 1] + 2 \\
&= 2 \cdot \frac{n(n-1)}{2} + 2 \\
&= n^2 - n + 2
\end{aligned}$$

□

## 2.8 Contare i percorsi

Molti problemi di conteggio si agganciano a schemi geometrici e riguardano percorsi che connettono dei punti in un grafo.

*Esempio 2.8.1* - Su uno schema geometrico costituito da un reticolo triangolare di punti connessi fra loro con dei segmenti, come si vede nella figura 2.4, consideriamo i percorsi che partono dal vertice in alto e procedono di un passo alla volta verso il basso. Il problema consiste nel determinare il numero dei percorsi che partono dal vertice in alto ed arrivano ad un dato punto del reticolo. Ogni punto del reticolo è individuato da una coppia di coordinate intere, come si vede in figura 2.4 dove sono descritti due (dei tanti) percorsi che congiungono il vertice  $(0,0)$  in alto al vertice  $(8,4)$  in basso. Così posto, il problema dipende solamente dalle coordinate  $(n, k)$  del punto sul quale terminano i percorsi.

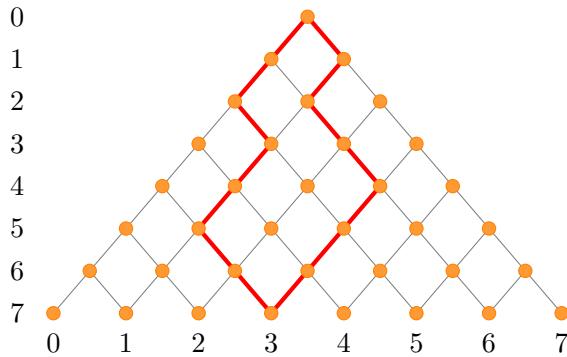


Figura 2.4: Reticolo triangolare di percorsi.

La soluzione del problema emerge osservando che in un punto si può arrivare solamente da due punti adiacenti posti sulla riga superiore; pertanto il numero di percorsi che portano ad un dato punto è dato dalla somma dei numeri corrispondenti ai due punti adiacenti posti sulla riga superiore. Questa sola osservazione permette di costruire un triangolo di numeri corrispondenti a

ciascun punto del reticolo, in cui ogni numero rappresenta il numero di percorsi che portano a quel punto, come si vede nella figura 2.5. E' questa una configurazione di numeri, studiata da Blaise Pascal (1623-1662) e da lui utilizzata nei suoi studi sulla probabilità come descritto nel suo testo "Traité du triangle arithmétique"; e per questo denominata *triangolo di Pascal*. Questo schema numerico era stato studiato 5 secoli prima dal matematico cinese Yang Hui (1238-1298) e nel XVI secolo dal matematico italiano Niccolò Tartaglia (1499-1557).

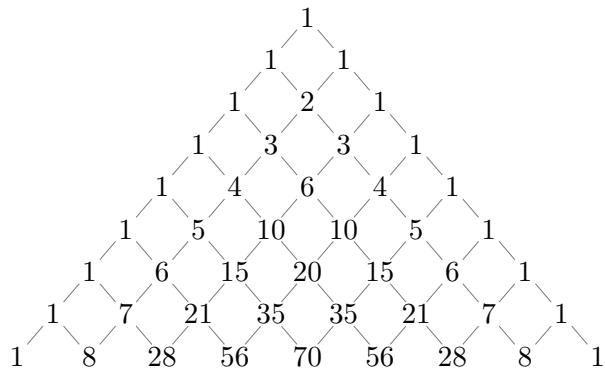


Figura 2.5: Il triangolo di Pascal: partendo dal numero 1 del vertice in alto ciascun numero delle righe successive è la somma dei due numeri adiacenti della riga precedente.

Cerchiamo ora un metodo per calcolare i numeri del triangolo di Pascal. Introduciamo una notazione dovuta ad Eulero: indichiamo con  $\binom{n}{k}$  il numero di percorsi dal vertice  $(0, 0)$  al punto  $(n, k)$ . Con questa notazione il triangolo di Pascal corrisponde al triangolo della figura 2.6.

$$\begin{array}{ccccccccc}
 & & \binom{0}{0} & & & & & & \\
 & & \binom{1}{0} & \binom{1}{1} & & & & & \\
 & & \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & & & \\
 & & \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & & & \\
 & & \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} & & \\
 & & \binom{5}{0} & \binom{5}{1} & \binom{5}{2} & \binom{5}{3} & \binom{5}{4} & \binom{5}{5} & \\
 & & \binom{6}{0} & \binom{6}{1} & \binom{6}{2} & \binom{6}{3} & \binom{6}{4} & \binom{6}{5} & \binom{6}{6} \\
 & & \binom{7}{0} & \binom{7}{1} & \binom{7}{2} & \binom{7}{3} & \binom{7}{4} & \binom{7}{5} & \binom{7}{6} & \binom{7}{7}
 \end{array}$$

Figura 2.6: Coefficienti binomiali del triangolo di Pascal.

Il problema è ora ricondotto al calcolo del valore  $\binom{n}{k}$ . Un semplice metodo potrebbe calcolare in successione, riga per riga, i valori, usando l'osservazione

utilizzata per la costruzione del triangolo di Pascal che si traduce nella formula (detta *formula di ricorrenza del triangolo di Pascal*)

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

E' questa una definizione ricorsiva che ha bisogno, per essere completa, di altre specifiche per la chiusura. Ragionando sulla figura si deduce che c'è un unico un percorso (rettilineo) che porta dal vertice in alto ai punti esterni disposti lungo i lati laterali del triangolo; questa osservazione si traduce nelle uguaglianze

$$\binom{n}{0} = \binom{n}{n} = 1$$

che costituiscono delle *condizioni al contorno*, che, unitamente alla relazione sopra permettono di calcolare i valori  $\binom{n}{k}$ . A questo punto un algoritmo che usa anche una lista di numeri che memorizzano di volta in volta i valori calcolati per la riga precedente di chiudere il problema, ma cerchiamo una soluzione diretta, una formula che fornisce il valore di  $\binom{n}{k}$  in funzione dei valori  $n$  e  $k$ . La formula chiusa per esprimere il binomiale è stata scoperta ancora da Pascal e si esprime come segue:

$$\binom{n}{k} = \frac{n!}{(n-k)! k!}$$

La dimostrazione di Pascal è storicamente rilevante in quanto anticipa uno schema di ragionamento analizzato ed utilizzato nel XX secolo noto con il nome di *induzione matematica*.

Per efficienza di calcolo, questa scrittura può essere equivalentemente scritta nella forma

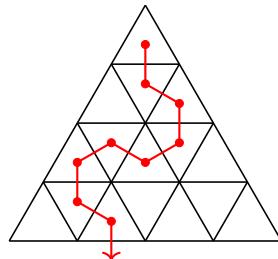
$$\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}$$

Sostituendo nella parte destra della formula sopra si ottiene un'espressione uguale; da ciò discende direttamente la relazione

$$\binom{n}{k} = \binom{n}{n-k}$$

che interpretata geometricamente sul triangolo dei coefficienti binomiali, giustifica la simmetria assiale del triangolo di Pascal rispetto all'asse verticale passante per il vertice del triangolo.

**Problema 2.8.1** È dato un triangolo equilatero con lato di lunghezza  $n$ , suddiviso in triangoli equilateri di lato unitario. Un *percorso* parte dal centro del triangolino alla sommità e passa per il centro di un triangolino adiacente non ancora visitato, della stessa riga o di una inferiore. Il percorso termina uscendo dal lato in basso. La figura che segue descrive un possibile percorso. Determinare il numero dei possibili percorsi, in funzione del numero  $n$  delle suddivisioni di ciascun lato del triangolo.



*Soluzione.* Il numero dei percorsi può essere determinato in base al seguente ragionamento. Ciascun percorso taglia in un unico punto ciascun segmento orizzontale che separa una riga di triangolini dalla riga sottostante, quindi è univocamente individuato da una sequenza di numeri naturali

$$(a_1, a_2, \dots, a_n) \quad 1 \leq a_i \leq i$$

dove  $a_i$  denota il numero del tratto di lato in cui il percorso taglia il segmento che separa dalla riga sottostante; ad esempio, il percorso descritto nella figura sopra è rappresentato dalla sequenza  $(1, 2, 1, 2)$ . C'è un'unica possibilità per la prima riga in alto, due per la seconda e così via, fino ad  $n$  per l'ultimo lato in basso. Quindi il numero totale dei possibili percorsi è

$$1 \cdot 2 \cdot 3 \cdots n = n!$$

□

## 2.9 Calcolo combinatorio

Oggiorno il problema del contare è in stretta connessione con i problemi combinatori che consistono nel determinare quante sono le possibili disposizioni o in quanti modi si possono selezionare degli oggetti compatibilmente con certi vincoli prefissati. Lo studio di questi problemi va sotto il nome di *calcolo combinatorio*. I problemi di conteggio più semplici coinvolgono e si risolvono con le sole quattro operazioni elementari. Ne sono conferma i seguenti esempi.

*Esempio 2.9.1* - Avendo a disposizione  $a$  berretti e  $b$  cappelli diversi, una persona può coprirsi il capo in  $a + b$  modi diversi.

*Esempio 2.9.2* - Avendo a disposizione  $a$  giacche e  $b$  pantaloni diversi, una persona può vestirsi in  $ab$  modi diversi.

Un'interessante tipologia di problemi di conteggio riguarda la determinazione del numero di possibili allineamenti in fila o raggruppamenti formati da elementi presi da un insieme di elementi. L'attributo *semplice* che verrà utilizzato in questo paragrafo indica gli allineamenti ed i raggruppamenti sono costituiti da elementi distinti.

## Permutazioni

Una *permutazione semplice* di  $n$  oggetti distinti è una qualsiasi loro disposizione in fila. Il numero di permutazioni di  $n$  oggetti distinti è

$$P(n) = n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

*Esempio 2.9.3* - Le possibili permutazioni delle lettere della parola ABC sono le seguenti  $1 \cdot 2 \cdot 3 = 6$ : ABC, ACB, BAC, BCA, CAB, CBA.

## Combinazioni

Una *combinazione semplice* di ordine  $k$  di un insieme di  $n$  elementi è un sottinsieme di  $k$  elementi presi dall'insieme. Il numero di queste combinazioni è pari a

$$C(n, k) = \frac{n!}{k!(n-k)!} = \frac{n \cdot (n-1) \cdots (n-k+1)}{k!}$$

Il numero  $C(n, k)$ , denotato anche con  $\binom{n}{k}$ , è detto *coefficiente binomiale*.

*Esempio 2.9.4* - Con l'insieme di lettere {A,B,C,D,E} si possono formare un numero di insiemi di 2 elementi pari a

$$\binom{5}{2} = \frac{5!}{2!3!} = 10$$

## Disposizioni

Una *disposizione semplice* di ordine  $k$  di un insieme di  $n$  elementi è una sequenza di  $k$  elementi distinti. A differenza che in una combinazione, in una disposizione è importante l'ordine con cui compaiono gli elementi.

Il numero di *disposizioni* di  $k$  oggetti distinti, presi da un insieme di  $n$  elementi, è pari a

$$D(n, k) = n \cdot (n-1) \cdots (n-k+1) = \frac{n!}{(n-k)!}$$

*Esempio 2.9.5* - Con le lettere ABCDE si possono formare  $5 \cdot 4 \cdot 3 = 60$  parole composte ciascuna da 3 lettere distinte.

**Problema 2.9.1** Dato un numero naturale  $n$ , determinare il numero delle possibili quaterne  $(a, b, c, d)$  soddisfacenti alla condizione  $0 \leq a < b < c < d \leq n$ .

**Soluzione.** Il problema trova immediata soluzione pensando ai numeri da 0 a  $n$  come ad un insieme di  $n+1$  elementi dal quale vengono estratti 4 numeri. Dunque il numero di quaterne è pari a  $\binom{n+1}{4}$ .  $\square$

**Problema 2.9.2** Dato un numero naturale  $n$ , determinare il numero delle possibili quaterne  $(a, b, c, d)$  soddisfacenti alla condizione  $0 \leq a \leq b \leq c \leq d \leq n$ .

**Soluzione.** Questo problema ha la stessa struttura del problema 2.9.1 con

la complicazione aggiuntiva che le quaterne  $(a, b, c, d)$  possono essere composte da elementi uguali. Il problema può essere ricondotto al problema 2.9.1 notando che c'è una corrispondenza biunivoca fra le possibili quaterne ed i sottoinsiemi di 4 oggetti presi dall'insieme  $\{0, 1, 2, \dots, n+3\}$ , identificando una generica quaterna  $(a, b, c, d)$  con il sottoinsieme  $\{a, b+1, c+2, d+3\}$ . Quindi il numero di possibili quaterne è pari a  $\binom{n+4}{4}$ .  $\square$

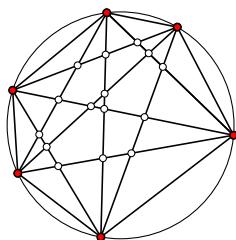
**Problema 2.9.3** Dati due numeri naturali  $n, k$ , con  $k \leq n$ , determinare in quanti modi si può esprimere  $n$  come somma di  $k$  addendi, considerando diverse le configurazioni che hanno gli stessi addendi permutati.

*Soluzione.* Esprimiamo il numero naturale  $n$  come somma di  $n$  unità:

$$n = \underbrace{1 + 1 + 1 + \cdots + 1}_{n \text{ addendi } 1}$$

Scegliendo  $k-1$  degli  $n-1$  segni di addizione  $+$  e sommando in un unico addendo le porzioni di espressioni delimitate dalle operazioni di addizione scelte, si ottengono  $k$  addendi. Pertanto il numero cercato è pari a  $\binom{n-1}{k-1}$ .  $\square$

**Problema 2.9.4** Sono dati  $n$  punti su una circonferenza e tutte le corde che li congiungono. Nessuna terna di corde passa per uno stesso punto. Determinare il numero di intersezioni interne alla circonferenza. Ad esempio, per  $n = 6$  si hanno 15 punti di intersezione, come descritto nella figura che segue.



*Soluzione.* La soluzione si ottiene con il seguente ragionamento: ogni punto è determinato da una quaterna di punti che sono gli estremi dei punti delle 2 corde che si intersecano. Reciprocamente, ogni quaterna di punti individua 2 coppie di corde che si intersecano in un punto interno alla circonferenza. Pertanto il numero di punti di intersezione interni alla circonferenza è pari al numero delle quaterne di punti che si possono estrarre da un insieme di  $n$  punti, e quindi è pari a  $\binom{n}{4}$ .  $\square$

## ESERCIZI

**2.1** Si vogliono contare i petali di una margherita. Analizzare i diversi metodi di conteggio che si possono usare a seconda che si possano o non si possano staccare i petali.

**2.2** Analizzare le differenti strategie di conteggio che vengono usate per contare le mele messe alla rinfusa in un cesto e per contare le mele disposte ordinatamente in fila in una cassetta.

**2.3** Analizzare le differenti strategie di conteggio con cui vengono generalmente svolti a mente da una persona i seguenti conteggi:

1. contare i numeri compresi fra 2398 e 2401
2. contare i numeri compresi fra 1998 e 2173
3. contare i numeri compresi fra 1483 e 7485
4. contare i numeri compresi fra 1689 e 3245

**2.4** Disponendo di un righello con tacche di misura ed utilizzando le quattro operazioni aritmetiche fondamentali, contare:

1. le righe di un foglio a righe
2. i quadretti di un foglio a quadretti

**2.5** Dimostrare che  $m$  mucchietti di  $n$  palline contengono un numero di palline pari a quelle di  $n$  mucchietti di  $m$  palline. Suggerimento: si possono disporre le palline in un reticolo rettangolare; oppure basta prendere una pallina da ciascun mucchietto e formare  $n$  mucchietti di  $m$  palline.

**2.6** Una persona può tenere le mani abbassate, oppure alzare la mano sinistra o destra o entrambe. Contemporaneamente può tenere i piedi appoggiati a terra oppure alzare una delle due gambe. Contare quante configurazioni diverse può assumere la persona.

**2.7** Determinare il numero di multipli di  $k$  compresi fra due dati numeri  $m$  ed  $n$ ,  $m < n$ .

**2.8** Determinare il numero di sfere disposte a piramide a base triangolare, dove ogni pila si adagia all'interno di tre sfere contigue dello strato precedente. Si assume come dato il numero  $n$  delle sfere disposte lungo il triangolo più grande alla base della piramide.

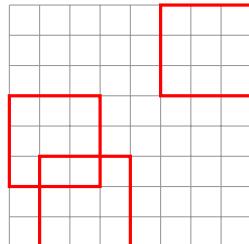
**2.9** Contare il numero di regioni di area finita che si creano mediante  $n$  rette intersecantesi, a tre a tre non passanti per lo stesso punto.

**2.10** Dato un insieme di  $n$  città, determinare il numero di diversi percorsi che collegano tutte le città, senza ripassare su una città già visitata.

**2.11** Contare quante sono le diagonali di un poligono di  $n$  vertici, intendendo con *diagonale* un segmento che unisce due vertici non contigui.

**2.12** Fissato nel piano cartesiano un punto  $C$  di coordinate intere  $(x, y)$  ed una distanza  $r$ , contare il numero dei punti di coordinate intere contenuti nella circonferenza di centro  $C$  e raggio  $r$ .

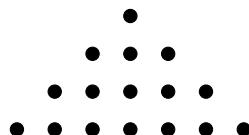
**2.13** Contare il numero di quadrati  $3 \times 3$  che si possono disegnare su una scacchiera  $8 \times 8$ . Generalizzare al caso di quadrati di  $k$  caselle per lato in una scacchiera di  $n \times n$  caselle.



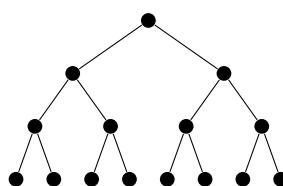
**2.14** Dato un reticolo rettangolare di punti di coordinate intere con i lati paralleli agli assi coordinati, contare il numero delle seguenti tipologie di quadrilateri che si possono ottenere in modo che abbiano i vertici sul reticolo:

- quadrati con i lati paralleli ai lati esterni del reticolo
- rettangoli con i lati paralleli ai lati esterni del reticolo
- quadrati qualsiasi
- rettangoli qualsiasi

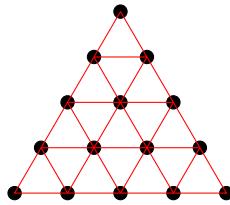
**2.15** Dei pallini sono disposti su  $n$  file secondo lo schema descritto nella figura che segue. Determinare il numero di pallini mediante un algoritmo e mediante una formula chiusa.



**2.16** Determinare il numero di nodi di un albero binario disposti su  $n$  livelli, come descritto nella figura che segue.

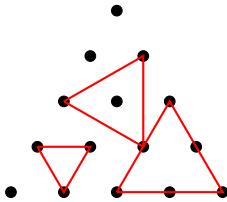


**2.17** Si consideri un reticolo che congiunge i punti disposti in modo regolare a distanze unitarie, in un triangolo equilatero avente  $n$  punti su ciascun lato, come descritto nella figura che segue.

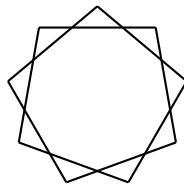


1. Determinare il numero di segmenti unitari costituenti il reticolo.
2. Determinare il numero dei triangolini di lati di lunghezza unitaria.
3. Determinare il numero complessivo dei triangoli equilateri (di tutte le dimensioni) formati sui lati del reticolo.

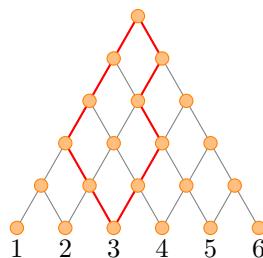
**2.18** Determinare il numero di triangoli equilateri che si possono costruire congiungendo 3 punti di un reticolo di punti disposti in modo regolare in un triangolo equilatero avente  $n$  punti su ciascun lato, come descritto nella figura che segue.



**2.19** Dato il numero  $n$  dei vertici di un poligono regolare, si congiunga ciascun vertice con un altro vertice, saltando  $k$  vertici; ad esempio, se  $k = 0$  si congiunge ciascun vertice con il successivo, ottenendo un poligono di  $n$  lati. Si chiude il ciclo quando si arriva sul vertice di partenza. Nel caso in cui rimangano dei vertici isolati, si ripete il procedimento per ciascuno di essi. Determinare in funzione di  $n$  e  $k$  il numero di lati della figura che così si ottiene ed il numero delle componenti connesse del grafo. La figura che segue illustra la stella che si ottiene per  $n = 9$  e  $k = 2$ .

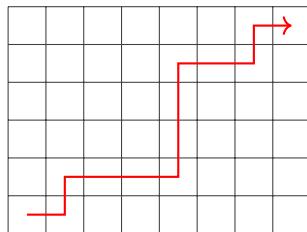


**2.20** È dato un reticolo di percorsi composto da  $n$  livelli. Una pallina posta nel vertice in alto scende verso il basso seguendo uno dei percorsi del reticolo. I nodi dell'ultimo livello in basso sono denotati mediante i numeri naturali da 1 a  $n$ , come indicato nella figura che segue, dove sono evidenziati due possibili percorsi dal vertice al nodo 3.



Ad ogni bivio la pallina ha le stesse probabilità di imboccare il percorso di sinistra o quello di destra. Determinare, in funzione di  $n$ , la probabilità che la pallina arrivi alla base nel nodo  $k$ . Suggerimento: risolvere il problema si risolve determinando preventivamente quanti sono i possibili percorsi che portano dal vertice in alto al nodo  $k$  alla base.

**2.21** Dato un reticolo  $m \times n$  di maglie, determinare il numero di percorsi composti da passi unitari verso l'alto o verso destro che partono dalla casella in basso a sinistra e portano alla casella in alto a destra.



**2.22** Contare tutte le coppie che si possono ottenere con le 26 lettere dell'alfabeto e le 9 cifre 1, 2, ..., 9.

**2.23** Contare il numero di stringhe di 8 caratteri composte dagli usuali 26 caratteri alfabetici, in modo da non avere caratteri duplicati vicini.

**2.24** Determinare il numero di squadre composte da 11 giocatori che si possono formare con 15 giocatori.

**2.25** Determinare in quanti modi si possono disporre 11 giocatori di calcio per una fotografia in cui 5 giocatori stanno in piedi e 6 sono accosciati davanti.

**2.26** Determinare in quanti modi un giocatore di carte può tenere in mano 13 carte da gioco.

**2.27** Nel gioco del poker ad ogni giocatore vengono distribuite 5 carte estratte da un mazzo di 32. Determinare in quanti modi diversi un giocatore può ricevere le carte.

**2.28** In una classe di 24 allievi si devono eleggere 2 capi classe. Determinare in quanti modi si può fare questa scelta.

**2.29** Calcolare il numero di anagrammi che si possono formare con le parole ROMA ed ALESSANDRIA.

**2.30** Determinare in quanti modi si possono disporre 2 torri su una scacchiera  $8 \times 8$  in modo che le due torri non siano reciprocamente sotto attacco, ossia esse devono stare su righe e colonne diverse.

**2.31** Un campionato a  $n$  squadre si svolge con partite di andata e ritorno. Determinare il numero di partite che vengono giocate.

**2.32** Contare le  $n$ -stringhe palindrome che si possono costruire con  $k$  caratteri.

**2.33** Determinare in quanti modi si possono assegnare  $n$  numeri di maglia a  $n$  giocatori.

**2.34** C'è una squadra di 11 giocatori schierata in fila per una foto. Vengono invitati anche i 5 giocatori di riserva che si inseriscono fra i giocatori già schierati. Determinare il numero delle diverse disposizioni della squadra.

**2.35** Ci sono  $n$  giocatori ed  $n$  maglie numerate da 1 a  $n$ ,  $n \geq 13$ . Determinare in quanti modi si possono assegnare le maglie sapendo che le maglie 1 e 13 devono essere assegnate ai due portieri.

**2.36** Determinare in quanti diversi modi si può formare una squadra di calcio a 5 avendo a disposizione 10 giocatori di cui 2 sono vincolati al ruolo di portiere.

**2.37** Ad una gara automobilistica partecipano  $n$  piloti. Determinare quanti diversi podi si possono avere.

**2.38** Ad una gara automobilistica partecipano  $n$  piloti. I primi 3, nell'ordine salgono sui gradini del podio. Quanti podi diversi si possono avere?

**2.39** Dimostrare matematicamente e motivare dal punto di vista combinatorio la seguente identità:

$$C(n, k) = \frac{D(n, k)}{P(k)}$$

**2.40** Determinare il numero di sottosequenze che si possono estrarre, mantenendo l'ordine di posizione degli elementi, da una sequenza di lunghezza  $n$  composta da elementi distinti.

**2.41** Determinare il numero di sottosequenze di lunghezza  $k$  che si possono estrarre, mantenendo l'ordine di posizione degli elementi, da una sequenza di lunghezza  $n$  composta da elementi distinti, con  $k \leq n$ .

**2.42** Determinare il numero di sottosequenze, composte da elementi adiacenti, che si possono estrarre da una sequenza di lunghezza  $n$  composta da elementi distinti.

**2.43** Determinare il numero di sottosequenze, di lunghezza  $k$ , composte da elementi adiacenti, che si possono estrarre da una sequenza di lunghezza  $n$  composta da elementi distinti, con  $k \leq n$ .

**2.44** Determinare il numero di diagonali di un poligono regolare; con *diagonale* si intende un segmento congiungente due vertici non contigui. Motivare i passaggi e le formule utilizzate.

**2.45** Determinare il numero di configurazioni diverse che si possono costruire con

due posizioni di un display a 7 segmenti. Determinare il numero nel caso in cui si imponga il vincolo che una configurazione sia costituita da almeno 2 segmenti. Determinare il numero nel caso si imponga il vincolo che i segmenti costituiscano una spezzata connessa.

**2.46** Determinare quanti numeri di  $n$  cifre in base 10 si possono costruire, con il vincolo che le cifre siano tutte diverse fra loro, escludendo i numeri con cifre 0 all'inizio.

**2.47** Ad un torneo partecipano 32 squadre. Nella prima fase le squadre vengono suddivise in 8 gironi da 4 squadre; passano alla fase successiva le prime due squadre di ogni girone. Nella seconda fase a squadre le 16 squadre rimaste in gara si incontrano con degli scontri ad eliminazione diretta. Determinare quante sono le possibili composizioni delle semifinali.

**2.48** Determinare il numero di possibili diversi tabelloni del lotto. Un tabellone è composto da 10 righe (ruote) e 5 colonne (estratti); ogni ruota viene composta estraendo da un'urna, contenente i numeri da 1 a 90, 5 numeri senza reimbussolamento; alla ruota successiva i 5 numeri precedentemente estratti vengono reimbussolati.

# 3

---

## SOMMARE

---

$$\pi = \frac{9801}{\sqrt{8}} \left( \sum_{n=0}^{\infty} \frac{(4n)!(26390n + 1103)}{(n!)^4 396^{4n}} \right)^{-1}$$

*Formula per il calcolo di  $\pi$*   
Srinivasa Ramanujan

L'operazione di addizione viene considerata come operazione primitiva in quanto è stata la prima ad essere concepita dall'uomo nella sua evoluzione storica ed anche perché è la prima che ciascuno di noi ha imparato a scuola. Questa operazione è strettamente legata al meccanismo del contare, basandosi sul seguente schema di isomorfismo strutturale: il conteggio degli elementi dell'unione fra due insiemi è pari alla somma delle cardinalità dei due insiemi.

Ricorrendo all'addizione è possibile esprimere algoritmicamente anche altre operazioni come la moltiplicazione (mediante ripetizione di addizioni), l'operazione fattoriale (mediante ripetizione di moltiplicazioni e, quindi, mediante ripetizione di ripetizioni di addizioni), l'operazione di radice quadrata (mediante un algoritmo ideato da Newton), ed altre.

L'operazione di addizione costituisce anche lo strumento matematico per costruire artefatti matematici, ossia dei numeri non esprimibili direttamente. La citazione sopra riporta una generale relazione ideata dal grande matematico indiano Srinivasa Ramanujan (1887-1920) che la scoprì nel 1910 e la pubblicò nel 1914.

### 3.1 Somme di numeri

L'operazione di addizione fra due generici numeri  $a$  e  $b$ , indicata con la tradizionale notazione  $a + b$  è una delle operazioni più elementari e nello stesso tempo la più frequentemente utilizzata. Abbiamo già visto nel precedente capitolo Contare che l'operazione di addizione è strettamente collegata con i procedimenti di conteggio. Questa operazione gode di molte proprietà; fra queste elenchiamo le seguenti:

1. per ogni coppia di numeri  $a, b$ , risulta  $a + b = b + a$  [proprietà *commutativa*]
2. per ogni terna di numeri  $a, b, c$ , risulta:  $(a+b)+c = a+(b+c)$  [proprietà *associativa*]
3. per ogni numero  $a$  si ha  $a + 0 = 0 + a = a$  [esistenza dell'*elemento neutro*]

Queste proprietà consentono di utilizzare, in alcuni casi, degli artifici per eseguire in modo semplificato delle addizioni fra due o più numeri.

**Esempio 3.1.1 -** Nelle righe che seguono sono indicati alcuni processi di calcolo semplificati che si sfruttano le proprietà elencate sopra:

$$\begin{aligned} 998 + 347 + 2 &\rightarrow 998 + 2 + 347 \rightarrow (998 + 2) + 347 \rightarrow 1000 + 347 \rightarrow 1347 \\ 994 + 1347 &\rightarrow 994 + 6 + 1341 \rightarrow 1000 + 1341 \rightarrow 2341 \end{aligned}$$

In tante situazioni si presenta il problema di descrivere l'addizione di molti addendi; in taluni casi queste addizioni di molti termini vengono indicate, in modo approssimativo indicando solo alcuni addendi che lasciano intuire i termini non esplicitamente indicati, come ad esempio si vede nella scrittura che segue:

$$2 + 4 + 6 + 8 + \cdots + 100$$

In queste situazioni viene spesso utilizzata una notazione alternativa: supponiamo di avere un generico insieme  $A = \{a_1, a_2, \dots, a_n\}$  di numeri di cui vogliamo calcolarne la somma; questa operazione è detta anche *sommatoria dell'insieme A* e viene denotata con una delle seguenti notazioni:

$$\sum_{x \in A} x \quad \sum_{k=1}^n a_k$$

Con queste notazioni l'addizione dei numeri pari compresi fra 2 e 100 scritta sopra può essere scritta in forma compatta con la notazione

$$\sum_{k=2}^{100} k$$

e la somma dei primi  $n$  numeri dispari con la notazione

$$\sum_{k=1}^n (2k - 1)$$

Se la successione degli addendi da sommare non presenta delle regolarità sulle quali basare delle efficienti strategie di calcolo, la somma degli addendi può essere calcolata mediante un algoritmo avente la seguente struttura:

---

**Algoritmo 1** - Somma di numeri
 

---

**Input:** numeri  $a_1, a_2, \dots, a_n$   
**Output:** somma dei numeri  $a_1, a_2, \dots, a_n$

```

1: s $\leftarrow 0$
2: for k from 1 to n do
3: $s \leftarrow s + a_k$
4: end for
5: return s
```

---

**Problema 3.1.1** I numeri di Fibonacci sono dei numeri naturali definiti come segue: i primi due numeri della successione sono 0 e 1; i successivi numeri sono definiti come somma dei due precedenti numeri nella successione. In base a questa definizione i primi numeri della successione di Fibonacci sono: 0, 1, 1, 2, 3, 5, 8, 13, 21, .... Determinare la somma dei primi  $k$  numeri di Fibonacci.

**Soluzione.** Il calcolo elementare di un singolo numero di Fibonacci richiederebbe l'esecuzione di un ciclo di addizioni di coppie di numeri fino ad arrivare alla determinazione del numero desiderato. In questo problema il procedimento può essere migliorato interlacciando il processo di calcolo di ciascun numero con il calcolo incrementale della loro somma. L'algoritmo di calcolo della somma dei primi  $n$  numeri della successione di Fibonacci è il seguente:

---

**Algoritmo 2** - Somma di numeri di Fibonacci
 

---

**Input:** numero  $n$  dei numeri da sommare  
**Output:** somma dei primi  $n$  numeri di Fibonacci

```

1: s $\leftarrow 0$
2: a $\leftarrow 1$
3: b $\leftarrow 1$
4: for n times
5: $s \leftarrow s + a$
6: $b \leftarrow a + b$
7: $a \leftarrow b - a$
8: end for
9: return s
```

---

**Osservazione.** Si incontrano molti problemi di somma in cui, al posto di un algoritmo di forza bruta come descritto sopra, si possono ideare delle strategie che evitano l'addizione di ciascun addendo e riconducono il procedimento all'esecuzione di poche operazioni, spesso facendo appoggio all'operazione di moltiplicazione.

## 3.2 Algoritmi per sommare

Un famoso aneddoto che riguarda il piccolo Karl Friederich Gauss (1777-1855), uno dei più grandi matematici di tutti i tempi, racconta che un giorno il suo maestro, per tenere gli alunni impegnati per un po' di tempo, avesse assegnato il compito di sommare i primi 100 numeri naturali:  $1+2+3+4+\cdots+98+99+100$ . Il maestro pensava che tutti gli allievi adottassero una tecnica brute-force che comporta l'esecuzione di 99 addizioni, ma Gauss, che all'epoca aveva 8 anni, poco dopo che il maestro ebbe terminato la dettatura della consegna, si avvicinò alla cattedra e vi posò la lavagnetta sulla quale vi era scritto il risultato. Ritornò a sedere e rimase per un lungo tempo con le braccia conserte attendendo che i suoi compagni terminassero il lavoro. Alla fine il maestro controllò le lavagnette e trovò che solo quella di Gauss conteneva il risultato corretto. I dettagli storici di questo aneddoto sono avvolti nella nebbia dell'agiografia, come avviene per i grandi personaggi. Si può quindi solo ipotizzare che il piccolo Karl abbia ragionato come segue: si costituiscono delle coppie di numeri: (*primo,ultimo*), (*secondo,penultimo*) e così via; si costituiscono 50 coppie di numeri ciascuna delle quali ha per somma 101; pertanto la somma dei 100 numeri dà 5050, come correttamente scrisse sulla sua lavagnetta Gauss.

Il problema risolto da Gauss ammette un'immediata generalizzazione come descritto nel prossimo problema.

**Problema 3.2.1** Determinare la somma dei primi  $n$  numeri naturali:

$$1 + 2 + 3 + \cdots + n$$

**Soluzione.** Una soluzione leggermente diversa a quella presentata sopra, che evita di distinguere i casi di  $n$  pari ed  $n$  dispari, consiste nel disporre gli addendi in due righe, in una in senso crescente e nell'altra in senso decrescente, come segue:

$$\begin{array}{cccccc} 1 & 2 & 3 & \cdots & n \\ n & n-1 & n-2 & \cdots & 1 \\ \hline n+1 & n+1 & n+1 & \cdots & n+1 \end{array}$$

Sommando in verticale si ottiene per ciascuna delle  $n$  coppie di numeri il risultato parziale  $n+1$ ; addizionando queste  $n$  somme parziali si ottiene  $n(n+1)$ ; questo valore va dimezzato in quanto gli addendi  $1, 2, \dots, n$  sono stati considerati due volte. In definitiva si ottiene il risultato

$$s(n) = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

In modo alternativo la somma  $1 + 2 + 3 + \cdots + n$  può essere calcolata utilizzando il metodo geometrico già illustrato nella soluzione del problema 2.6.1.  $\square$

Un'ulteriore generalizzazione del problema precedente è descritta nel seguente problema.

**Problema 3.2.2** Determinare la somma  $t(m, n)$  dei numeri naturali compresi fra due dati numeri naturali  $m$  ed  $n$ ,  $m \leq n$ .

*Soluzione.* Indichiamo con  $t(m, n)$  la somma dei numeri naturali compresi fra i numeri naturali  $m$  ed  $n$ . Ovviamente non è accettabile una soluzione nella forma aperta

$$t(m, n) = m + (m + 1) + (m + 2) + \cdots + n$$

in quanto questa scrittura *non* è un'espressione e neanche un algoritmo. Per questo problema, nonostante l'apparenza, non c'è alcuna esigenza di ricorrere a dei controlli ciclici; tenendo presente che la somma dei primi  $k$  numeri naturali vale  $s(k) = \sum_1^k i = k(k + 1)/2$ , come visto nell'esempio precedente, si ottiene

$$\begin{aligned} &m + (m + 1) + (m + 2) + \cdots + n = \\ &s(n) - s(m - 1) = \\ &n(n + 1)/2 - m(m - 1)/2 = \\ &(n^2 - m^2 + n + m)/2 = \\ &(n + m)(n - m + 1)/2 \end{aligned}$$

L'espressione che costituisce la soluzione è dunque

$$t(m, n) = (n + m)(n - m + 1)/2$$

□

**Problema 3.2.3** Determinare la somma dei primi  $n$  numeri naturali dispari, ossia:

$$1 + 3 + \cdots + (2n - 1)$$

*Soluzione.* Utilizzando l'idea già vista nella dimostrazione di Gauss si può costruire la seguente tabella in cui a ciascuno degli  $n$  termini della somma viene addizionata un'unità:

$$\begin{array}{cccccc} 1 & 3 & 5 & \cdots & (2n - 1) \\ 1 & 1 & 1 & \cdots & 1 \\ \hline 2 & 4 & 6 & \cdots & 2n \end{array}$$

Sommando i numeri che compaiono sotto la linea delle somme parziali in verticale si ottiene il seguente risultato:

$$2 + 4 + 6 + \cdots + 2n = 2(1 + 2 + 3 + \cdots + n) = 2n(n + 1)/2$$

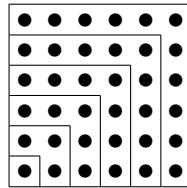
A questo valore bisogna sottrarre la somma  $n$  della riga degli  $n$  addendi 1; si ricava che la somma cercata è pari a

$$2n(n + 1)/2 - n = n^2$$

In modo alternativo, il calcolo può essere sviluppato attraverso i seguenti passi:

$$\begin{aligned}
 1 + 3 + \dots + (2n - 1) &= 1 + 2 + 3 + 4 + \dots + (2n - 1) - (2 + 4 + \dots + 2n - 2) \\
 &= \sum_{k=1}^{2n-1} k - 2 \sum_{k=1}^{n-1} k \\
 &= (2n - 1)2n/2 - 2(n - 1)n/2 \\
 &= 2n^2 - n - n^2 + n \\
 &= n^2
 \end{aligned}$$

La formula può essere dimostrata anche in modo geometrico, basandosi sulla seguente figura: i numeri vengono disposti su due lati consecutivi di un quadrato, generando ad ogni passo un nuovo quadrato. In questo modo si perviene allo stesso risultato.



### 3.3 Progressioni

Una *progressione* è una successione  $a_1, a_2, \dots, a_n, \dots$  di numeri, detti *termini*. La somma dei primi  $n$  termini di una progressione viene indicata con  $S_n$ .

Consideriamo ora delle particolari progressioni che soddisfano a delle proprietà così caratterizzanti che consentono di determinare la somma dei primi termini della successione solo conoscendo il numero di addendi.

#### Progressioni aritmetiche

Una *progressione aritmetica* è una successione in cui ciascun termine è dato dalla somma del termine precedente e di una costante  $d$  detta *ragione*. Ad esempio la successione  $4, 7, 10, 13, 16, \dots$  è una progressione aritmetica di ragione 3.

Dalla definizione stessa, l' $n$ -esimo termine di una successione aritmetica di ragione  $d$  ed il cui primo termine è  $a_1$  è

$$a_n = a_1 + (n - 1)d$$

Calcoliamo ora la somma  $S_n$  dei primi  $n$  termini di una progressione aritmetica; richiamando la strategia di Gauss, scriviamo due volte i termini della serie, disponendoli su due righe come descritto dalla figura che segue:

$$\begin{array}{rcl}
 S_n & = & a_1 \quad a_2 \quad \cdots \quad a_n \\
 S_n & = & a_n \quad a_{n-1} \quad \cdots \quad a_1 \\
 \hline
 2S_n & = & (a_1 + a_n) \quad (a_2 + a_{n-1}) \quad \cdots \quad (a_n + a_1)
 \end{array}$$

Le espressioni fra parentesi sono uguali fra loro in quanto ciascuna ottenuta da due addendi che si bilanciano (uno cresce e l'altro decresce della ragione). Pertanto:

$$S_n = \frac{n(a_1 + a_n)}{2}$$

Sostituendo al termine  $a_n$  il valore calcolato precedentemente si ottiene

$$S_n = \frac{n[2a_1 + (n-1)d]}{2}$$

**Problema 3.3.1** Calcolare la somma dei primi 100 numeri dispari.

**Soluzione.** La somma è calcolabile attraverso i seguenti passaggi:

$$\begin{aligned} a_1 &= 1 \\ d &= 2 \\ n &= 100 \\ S_n &= \frac{n[2a_1 + (n-1)d]}{2} = 10000 \end{aligned}$$

### Progressioni geometriche

Una *progressione geometrica* è una successione di numeri tali che il rapporto tra un elemento ed il suo precedente è sempre costante. Tale costante è detta *ragione* della progressione. Ad esempio la successione 1, 2, 4, 8, 16, 32, 64, ... è una progressione geometrica di ragione 2.

Dalla definizione stessa, l' $n$ -esimo termine di una successione geometrica di ragione  $q$  ed il cui primo termine è  $a_1$  è

$$a_n = a_1 q^{n-1}$$

Calcoliamo ora la somma  $S_n$  dei primi  $n$  termini di una progressione geometrica di ragione  $q$ :

$$S_n = a_1 + a_2 + a_3 + \cdots + a_n = a_1 + a_1 q + a_1 q^2 + \cdots + a_1 q^{n-1} = a_1(1 + q + q^2 + \cdots + q^{n-1})$$

Moltiplicando per  $1 - q$  si ottiene

$$a_1(1-q)(1+q+q^2+\cdots+q^{n-1}) = a_1(1-q+q-q^2+q^2-q^3+\cdots+q^{n-1}-q^n) = a_1(1-q^n)$$

da cui

$$S_n = a_1 \frac{q^n - 1}{q - 1}$$

**Problema 3.3.2** Determinare la somma dei seguenti  $n$  termini:

$$\underbrace{1 + 11 + 111 + 1111 + 11111 + \dots}_{n \text{ addendi}}$$

*Soluzione.* Una soluzione di tipo algebrico è la seguente:

$$\begin{aligned} \frac{1}{9}(9 + 99 + 999 + \dots) &= \frac{1}{9}((10 - 1) + (10^2 - 1) + (10^3 - 1) + \dots + (10^n - 1)) \\ &= \frac{1}{9}(10(1 + 10 + 10^2 + \dots + 10^{n-1}) - (1 + 1 + 1 + \dots + 1)) = \frac{1}{9}\left(10 \frac{10^n - 1}{10 - 1} - n\right) \\ &= \frac{10^{n+1} - 9n - 10}{81} \end{aligned}$$

□

### 3.4 Somma di tabelle di numeri

In alcuni casi i numeri sono disposti secondo una particolare struttura geometrica la cui regolarità suggerisce dei metodi per calcolare la somma degli elementi, senza ricorrere ad algoritmi con cicli.

**Problema 3.4.1** Calcolare la somma dei numeri della seguente tabella  $10 \times 10$ .

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 |
| 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

*Soluzione.* La prima mossa consiste nel determinare una regola dei numeri presenti nello schema. Siamo indotti a ricercare delle proprietà locali, fra numeri vicini; in particolare, per la nostra abitudine di leggere per righe, da sinistra verso destra, emergono le seguenti regolarità:

- i numeri in ogni riga sono in progressione aritmetica di ragione 1
- i numeri dalla seconda riga in poi hanno un'unità in più dei corrispondenti numeri della riga precedente

Questa regolarità suggerisce la seguente strategia:

1. la somma della prima riga vale  $(10 \times 11)/2 = 55$

2. la seconda riga vale  $55 + 10 = 65$
3. la terza riga vale  $65 + 10$

e così via per le righe a seguire. Si individua così una progressione aritmetica di 10 termini, di ragione 10 il cui primo termine vale 55. La somma vale pertanto:

$$\begin{aligned} a_n &= a_1 + (n-1)d = 55 + 9 \cdot 10 = 145 \\ s_n &= \frac{n(a_1 + a_n)}{2} = \frac{10(55 + 145)}{2} = 1000 \end{aligned}$$

Una regolarità più efficace emerge considerando le coppie di numeri simmetrici rispetto alla diagonale principale della tabella. Si evidenziano così delle coppie che danno per somma 20. Ciò suggerisce la seguente idea: in ciascuna coppia la componente più piccola *presta* 10 unità all'altra componente della coppia; in questo modo tutti i numeri della tabella diventano pari a 10 e pertanto la somma dei 100 numeri vale 1000. In alternativa, si hanno 50 coppie che hanno per somma 20 e si arriva ancora al risultato  $20 \times 50 = 1000$ . La linearità del ragionamento, la brevità del calcolo finale ci appaga pienamente e ci lascia con la consapevolezza che di meglio non si possa fare. Questo ragionamento si presta ad essere generalizzato ad una generica tabella quadrata di dimensione  $n \times n$  la cui somma degli elementi vale  $n^3$ .  $\square$

**Problema 3.4.2** Calcolare la somma dei numeri di una tavola pitagorica  $n \times n$ .

|          |      |      |     |          |
|----------|------|------|-----|----------|
| 1        | 2    | 3    | ... | $n$      |
| 2        | 4    | 6    | ... | $2n$     |
| 3        | 6    | 9    | ... | $3n$     |
| $\vdots$ |      |      |     | $\vdots$ |
| $n$      | $2n$ | $3n$ | ... | $n^2$    |

**Soluzione.** Ciascun numero della tabella è il prodotto del corrispondente numero di riga e di colonna. Pertanto la somma di tutti i numeri è pari a

$$(1 + 2 + \dots + n)^2$$

Sviluppando questa espressione si ottengono gli addendi corrispondenti ai prodotti dei numeri di riga e di colonna della tabella. Utilizzando la formula di Gauss si ottiene come risultato

$$\frac{n^2(n+1)^2}{4}$$

$\square$

## 3.5 Somme infinite

È opinione comune che la somma di infiniti numeri positivi e non nulli non possa produrre un valore finito. Su questo presupposto il filosofo Zenone,

nato ad Elea (attualmente in Campania) e vissuto tra la fine del VI e l'inizio del V secolo a.C. propose un famoso paradosso con l'intenzione di dimostrare che il movimento non esiste. La storia è questa: Achille insegue una tartaruga su una pista rettilinea. Indicando con  $x_0$  la posizione iniziale di Achille, e con  $x_1 > x_0$  la posizione iniziale della tartaruga, Achille impiega un tempo  $t_1$  per raggiungere la posizione  $x_1$ ; nel frattempo la tartaruga avrà raggiunto la posizione  $x_2 > x_1$ . Achille impiega il tempo  $t_2$  per spostarsi da  $x_1$  a  $x_2$  ma nel frattempo la tartaruga avrà raggiunto la posizione  $x_3 > x_2$ . Questo processo va avanti all'infinito; dunque, per raggiungere la tartaruga, Achille impiega il tempo

$$t = t_1 + t_2 + \dots$$

Poichè Zenone riteneva che la somma di infiniti tempi non poteva essere finita, concluse che, contrariamente alla nostra esperienza comune, Achille non raggiungerà mai la tartaruga. Ovviamente questa conclusione è inficiata dalla fallacia dell'ipotesi iniziale.

La somma di infiniti addendi viene detta *serie*. Una serie composta dagli addendi  $a_1, a_2, \dots, a_n, \dots$  viene solitamente scritta nella forma compatta

$$\sum_{n=1}^{\infty} a_n$$

Le somme infinite sollevano il problema relativo alla convergenza: ci sono somme infinite il cui risultato è un termine infinitamente grande, ad esempio le seguenti due:

$$1 + 1 + 1 + \dots + 1 + \dots$$

$$1 + 2 + 3 + \dots + n + \dots$$

danno un risultato infinito, denotato con il tradizionale simbolo  $\infty$ . È evidente che una condizione necessaria affinché una somma infinita possa convergere ad un valore finito debba essere composta da infiniti addendi piccoli. Il bilanciamento generato dai due attributi *infiniti* e *piccoli* dovrà produrre un valore finito. Prima di andare oltre è necessario considerare un esempio di serie che non da risultato  $\infty$ .

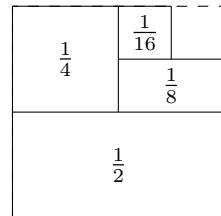
**Esempio 3.5.1 -** Consideriamo la progressione geometrica di ragione  $q = \frac{1}{2}$  ed avente  $a_1 = \frac{1}{2}$  come primo termine. Calcoliamo

$$\sum_{k=1}^{\infty} \frac{1}{2} \left(\frac{1}{2}\right)^{k-1} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots$$

Applicando la formula che fornisce la somma dei primi  $n$  termini di una progressione geometrica, si ha

$$\sum_{k=1}^{\infty} \frac{1}{2} \left(\frac{1}{2}\right)^{k-1} = \frac{1}{2} \cdot \frac{\left(\frac{1}{2}\right)^n - 1}{\frac{1}{2} - 1}$$

Essendo  $\lim_{n \rightarrow \infty} \left(\frac{1}{2}\right)^n = 0$  si ottiene che il risultato della serie è 1. Questo risultato può essere ottenuto anche in modo geometrico basandosi sulla seguente figura.



Da questa figura si deduce che la somma degli infiniti termini della serie tende all'area del quadrato di lato unitario e pertanto la serie tende al valore 1.

**Esempio 3.5.2 -** La condizione che gli addendi della somma infinita debbano diventare sempre più piccoli non è sufficiente a garantire la convergenza del risultato ad un valore finito; ad esempio si può dimostrare che la seguente somma infinita diverge al valore  $\infty$ :

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$$

**Problema 3.5.1** Determinare la frazione alla quale converge la seguente serie:

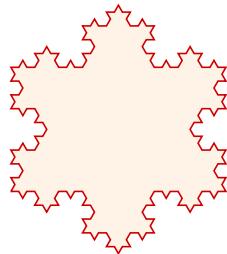
$$1 + \frac{1}{10} + \frac{1}{100} + \frac{1}{1000} + \frac{1}{10000} + \dots$$

**Soluzione.** La richiesta di ottenere come risultato una frazione e non un numero approssimato ci impone di utilizzare una strategia alternativa rispetto all'uso di un algoritmo ciclico. Si può osservare che ogni addendo della somma infinita contribuisce alla somma con un'unità decimale che dipende dalla posizione dell'addendo; il valore a cui tende la somma è pertanto la frazione periodica

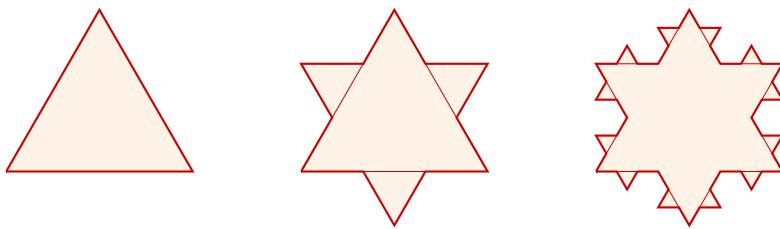
$$1.111111111111\dots = 1.\bar{1}$$

Convertendo il numero periodico  $1.\bar{1}$  si ottiene  $\frac{10}{9}$ .  $\square$

**Problema 3.5.2** Calcolare il perimetro e l'area della figura limite alla quale tende la figura nota come *fioccetto di von Kock*, generata da un triangolo equilatero iniziale di lato  $l$ ; ad ogni passo ciascun lato viene suddiviso in 3 parti e sulla parte centrale di queste viene estroflesso un triangolo equilatero.



*Soluzione.* Per orientarci nei ragionamenti e nei calcoli consideriamo le prime istanze del fiocco di von Kock, descritte nella figura che segue.



Dal processo di generazione delle varie istanze della figura si può notare che ogni segmento di lunghezza da vita a 4 lunghezza complessiva  $\frac{4}{3}l$ . Indicando con  $P_n$  il perimetro della figura ottenuta al passo  $n$  si ha:  $P_0 = 3l$ ,  $P_1 = 3 \cdot \frac{4}{3}l$ , ..., ...,  $P_n = 3 \cdot (\frac{4}{3})^n l$ ; essendo  $\frac{4}{3} > 1$ , all'aumentare di  $n$  il perimetro  $P_n$  tende a  $\infty$ .

Per il calcolo dell'area facciamo alcune osservazioni preliminari. L'area è limitata in quanto il fiocco è incluso nel cerchio circoscritto al triangolo equilatero iniziale. Ad ogni passo, alla figura generata al passo precedente vengono aggiunti dei triangolini aventi ciascuno il lato di lunghezza pari ad  $\frac{1}{3}$  la lunghezza dei lati dei triangoli aggiunti al passo precedente. Il numero dei triangolini che vengono aggiunti ad ogni passo sono pari al numero dei lati della figura ottenuta al passo precedente. Quindi, al passo  $n$ -esimo vengono aggiunti  $3 \cdot 4^n$  nuovi triangolini ciascuno avente una lunghezza pari a  $(\frac{1}{3})^n$ . Indicando con  $S_0$ , al passo  $n$  si formano  $3 \cdot 4^{n-1}$  triangoli ciascuno dei quali ha un'area pari  $S_0/9^k$ . L'area del fiocco ottenuto al passo  $n$  è pari a

$$S_n = S_0 \left( 1 + \sum_{k=1}^n \frac{3 \cdot 4^{k-1}}{9^k} \right) = S_0 \left( 1 + \frac{1}{3} \sum_{k=1}^n \left(\frac{4}{9}\right)^k \right) = S_0 \left( 1 + \frac{1}{3} \cdot \frac{1}{1 - \frac{4}{9}} \right) = \frac{8}{5} S_0$$

dove, nel calcolo della sommatoria, è stata applicata la formula per il calcolo di una serie geometrica. Essendo  $S_0 = \frac{l^2 \sqrt{3}}{4}$  si ottiene che l'area del fiocco di von Kock di lato iniziale  $l$  tende al valore

$$S_\infty = \frac{2\sqrt{3}}{5} l^2$$

□

## ESERCIZI

**3.1** Calcolare, mediante un algoritmo e mediante una formula chiusa, la somma dei primi  $n$  numeri pari.

**3.2** Calcolare, mediante un algoritmo e mediante una formula chiusa, la somma dei primi  $n$  numeri dispari.

**3.3** Calcolare la somma dei quadrati dei primi  $n$  numeri naturali:  $1^2 + 2^2 + 3^2 + \dots + n^2$ .

**3.4** Determinare la somma dei primi  $n$  numeri dispari:  $1 + 3 + 5 + \dots + (2n - 1)$

**3.5** Determinare la somma delle prime  $n$  potenze di 2:  $2^0 + 2^1 + \dots + 2^{n-1}$

**3.6** Determinare la somma dei primi  $n$  termini della sommatoria

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

**3.7** Determinare il numero minimo  $n$  di addendi necessari affinchè la sommatoria

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

superi un dato valore  $x$ .

**3.8** Determinare il più grande numero rappresentabile con  $n$  cifre binarie.

**3.9** Determinare la somma di tutti i multipli di un dato valore naturale  $k$  compresi fra due numeri naturali  $m$  ed  $n$ ,  $m \leq n$ .

**3.10** Calcolare la somma del seguente insieme di numeri, disposti su  $n$  righe in modo triangolare:

$$\begin{array}{ccccccc} & & & & 1 \\ & & & & 2 & 2 \\ & & & & 3 & 3 & 3 \\ & & & & 4 & 4 & 4 & 4 \end{array}$$

**3.11** Calcolare mediante un algoritmo e mediante una formula chiusa la somma degli addendi naturali consecutivi (a partire da 1), disposti su  $n$  righe, secondo lo schema descritto a seguire.

$$\begin{array}{ccccccc} & & & & 1 \\ & & & & 2 & 3 \\ & & & & 4 & 5 & 6 \\ & & & & 7 & 8 & 9 & 10 \end{array}$$

**3.12** Calcolare la somma dei numeri della tabella  $n \times n$  sotto riportata dove ogni elemento è la somma dei corrispondenti numeri di riga e colonna.

|          |         |         |     |          |
|----------|---------|---------|-----|----------|
| 2        | 3       | 4       | ... | $n + 1$  |
| 3        | 4       | 5       | ... | $n + 2$  |
| 4        | 5       | 6       | ... | $n + 3$  |
| $\vdots$ |         |         |     | $\vdots$ |
| $n + 1$  | $n + 2$ | $n + 3$ | ... | $2n$     |

## 4

---

### MISURARE

---

*Qual è il geometra che tutto s'affige  
per misurar lo cerchio, e non ritrova,  
pensando, quel principio ond'elli indige,  
tal era io a quella vista nova;  
veder volea come si convenne  
l'immagine al cerchio e come vi si indova.*

Dante,  
*Paradiso, Canto XXXIII, v. 133-138*

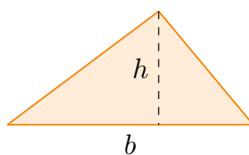
Già al tempo degli Egizi la geometria veniva utilizzata per risolvere il problema di ridefinire i confini dei terreni dopo lo straripamento del Nilo e complessi calcoli numerici si resero necessari (con l'uso anche di frazioni e di radici quadrate) per la costruzione delle piramidi dei faraoni. Tuttavia per gli egizi tutte le discipline scientifiche (matematica, geometria, astronomia) erano di natura essenzialmente pratica, senza risvolti teorici.

La prima sistemazione scientifica inquadrata in un impianto formale rigoroso risale alla civiltà greca: geometri e filosofi quali Euclide, Pitagora e Talete hanno gettato le basi della geometria che si è mantenuta pressochè invariata fino al XIX secolo.

## 4.1 Area del triangolo

La figura geometrica più semplice ed elementare che esiste ed al contempo la più generale è il *triangolo*. Più semplice perché è la figura delimitata dal minor numero possibile di lati; più elementare perché ciascun'altra figura può essere scomposta in triangoli; più generale perché non deve rispettare alcun vincolo sui lati ed angoli eccetto un vincolo di esistenza che afferma che: *In un triangolo ogni lato deve essere minore della somma degli altri due*.

Il metodo tradizionale per determinare l'area di un triangolo consiste nell'applicare la notissima formula che a parole recita: *Base per altezza diviso due*. Questo metodo richiede la conoscenza della lunghezza di un lato e dell'altezza corrispondente.



In questo caso il procedimento per il calcolo dell'area del triangolo si riduce alla valutazione dell'espressione  $(b * h)/2$ . Anche una semplice formula come questa può essere espressa in forma di algoritmo:

---

### Algoritmo 1 - Area di un triangolo: $\text{area}(b, h)$

---

**Input:** misure  $b$  della base ed  $h$  dell'altezza del triangolo

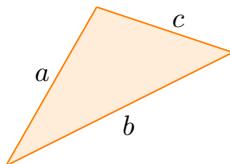
**Output:** area del triangolo

1: **return**  $(b * h)/2$

---

L'algoritmo 1 richiede che l'esecutore sia in grado di eseguire la moltiplicazione e la divisione fra numeri.

Nel precedente procedimento, il dato costituito dalla conoscenza della lunghezza dell'altezza risulta artificioso e spurio e non direttamente collegato alla conoscenza dei dati che individuano direttamente il triangolo, ossia la lunghezza dei tre lati. Essendo che un triangolo è univocamente determinato dalla lunghezza dei suoi lati, deve esistere una legame funzionale fra l'area e la lunghezza dei lati; in altre parole, una formula che fornisce l'area in funzione dei lati.



Le lunghezze dei tre lati devono soddisfare la condizione di costruibilità del triangolo richiamata all'inizio di questo paragrafo. Con questi ipotesi l'area può essere calcolata mediante la formula, attribuita ad Erone vissuto nel I sec. d.C. ad Alessandria d'Egitto, che, inserita in un algoritmo, assume la

seguente forma: denotando con  $a, b, c$  le lunghezze dei lati del triangolo e con  $p$  il semiperimetro, l'area  $s$  è data dalla formula

$$s = \sqrt{p(p-a)(p-b)(p-c)}$$

che può essere frammentata e vestita ad algoritmo (algoritmo 2).

---

**Algoritmo 2** - Formula di Erone:  $erone(a, b, c)$ 


---

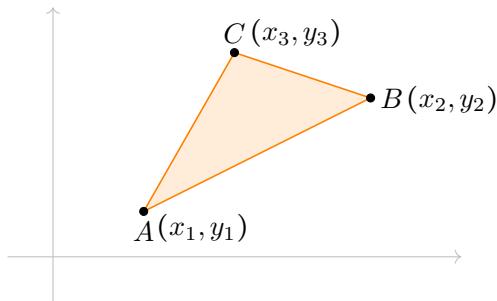
**Input:** misure  $a, b, c$  dei lati del triangolo

**Output:** area del triangolo

- 1:  $p \leftarrow (a + b + c)/2$
  - 2:  $s \leftarrow \sqrt{p * (p - a) * (p - b) * (p - c)}$
  - 3: **return**  $s$
- 

Questo algoritmo richiede che l'esecutore sia in grado di eseguire le quattro operazioni fondamentali sui numeri e l'operazione di calcolo della radice quadrata.

Nel caso in cui il problema sia ambientato in un sistema di assi cartesiani i dati ed i risultati vengono esplicitati come illustrato nella figura che segue.



La soluzione più breve consiste nel calcolo delle lunghezze dei tre lati, in modo da ricondursi al calcolo dell'area mediante la formula di Erone (algoritmo 2). La distanza fra due generici punti è calcolabile mediante l'algoritmo 3.

---

**Algoritmo 3** - Distanza fra due punti:  $dist(x_1, y_1, x_2, y_2)$ 


---

**Input:** punti  $P(x_1, y_1), Q(x_2, y_2)$

**Output:** distanza fra i punti  $P$  e  $Q$

- 1:  $x \leftarrow x_2 - x_1$
  - 2:  $y \leftarrow y_2 - y_1$
  - 3:  $d \leftarrow \sqrt{x^2 + y^2}$
  - 4: **return**  $d$
- 

Componendo i due algoritmi 2 e 3 si ottiene l'algoritmo complessivo per il calcolo dell'area di un triangolo (algoritmo 4).

---

**Algoritmo 4** - Area di un triangolo:  $\text{area}(x_1, y_1, x_2, y_2, x_3, y_3)$ 

---

**Input:** coordinate dei tre vertici:  $[A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)]$ **Output:** area del triangolo

- 1:  $a \leftarrow \text{dist}(x_2, y_2, x_3, y_3)$
  - 2:  $b \leftarrow \text{dist}(x_1, y_1, x_3, y_3)$
  - 3:  $c \leftarrow \text{dist}(x_1, y_1, x_2, y_2)$
  - 4:  $s \leftarrow \text{erone}(a, b, c)$
  - 5: **return**  $s$
- 

Un modo alternativo consiste nel calcolare, usando gli strumenti della Geometria Analitica, la lunghezza di una base del triangolo e della corrispondente altezza, in modo da ricondursi alla situazione dell'algoritmo 1. Il dettaglio di questo metodo è descritto nell'algoritmo 5.

---

**Algoritmo 5** - Area di un triangolo nel piano cartesiano

---

**Input:** coordinate dei tre vertici:  $[A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)]$ **Output:** area  $s$  del triangolo

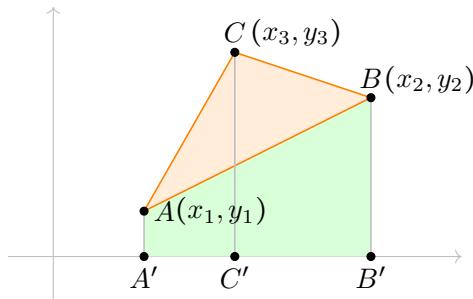
- 1:  $b \leftarrow$  distanza fra i punti  $A$  e  $B$
  - 2:  $r \leftarrow$  retta per  $A, B$
  - 3:  $p \leftarrow$  retta per  $C$  e perpendicolare a  $r$
  - 4:  $H \leftarrow$  punto di intersezione fra  $r$  ed  $p$
  - 5:  $h \leftarrow$  distanza fra i punti  $C$  e  $H$
  - 6:  $s \leftarrow (b * h)/2$
  - 7: **return**  $s$
- 

Le istruzioni corrispondenti alle linee 2-5 costituiscono la soluzione del sotto-problema espresso nella seguente singola assegnazione:

$$h \leftarrow \text{distanza del punto } C \text{ dalla retta } r$$

L'algoritmo 5 richiede che l'esecutore sia in grado di svolgere le operazioni della geometria cartesiana indicate alle linee 1-5. Nel caso l'esecutore fosse solo in grado di operare sui numeri, queste istruzioni dovranno essere esplicitate con delle espressioni ed assegnazioni coinvolgenti solo numeri, avvalendosi anche di variabili per memorizzare i risultati intermedi dei calcoli. Un ulteriore livello di astrazione consiste nella creazione di altre entità geometriche (punti, rette) e definizione su queste di operazioni (retta per due punti, ...). È questo un tipico e produttivo approccio realizzabile mediante i moderni linguaggi di programmazione.

Un altro metodo per il calcolo dell'area di un triangolo cartesiano discende dalla seguente osservazione: tracciando le proiezioni verticali dei vertici (punti  $A', B', C'$ ) si evidenziano dei trapezi rettangoli aventi le basi poste verticalmente ( $ABB'A'$ ,  $BBC'B'$ ,  $CAA'C'$ ):



L’area del triangolo  $\triangle ABC$  risulta uguale alla somma algebrica delle aree di questi trapezi, prendendo il segno positivo quando il lato del triangolo (lato obliquo del trapezio) viene percorso da destra verso sinistra e negativo negli altri casi; con riferimento alla figura riportata sopra si ha:

$$\text{area}(ABC) = -\text{area}(ABB'A') + \text{area}(BCC'B') + \text{area}(CAA'C')$$

Svolgendo i calcoli si ricava:

$$s = \frac{1}{2} |x_2 y_3 - x_3 y_2 - x_1 y_3 + x_3 y_1 + x_1 y_2 - x_2 y_1|$$

(il segno di valore assoluto permette di generalizzare la situazione in modo che che i tre vertici si possano trovare in qualsiasi punto del piano cartesiano e che possano avere coordinate negative). La precedente formula può essere espressa in forma equivalente usando la notazione del determinante di una matrice  $3 \times 3$  costruita con le coordinate dei vertici del triangolo:

$$s = \left| \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \right|$$

C’è da osservare che il calcolo con il metodo appena presentato richiede solamente l’uso delle operazioni elementari, senza calcoli di radici quadrate.

## 4.2 Area dei quadrilateri

I poligoni più semplici dopo il triangolo sono i *quadrilateri*. Spesso vengono considerate delle particolari sottocategorie di quadrilateri, che assumono delle denominazioni specifiche e che hanno delle specifiche formule per determinare la loro area. La lista che segue illustra queste particolari famiglie di quadrilateri ed il corrispondente procedimento per determinarne l’area.

- *quadrato*: quadrilatero avente i lati e gli angoli uguali ( $\text{area} = \text{lato} \times \text{lato}$ )
- *rettangolo*: quadrilatero composto da lati a coppie uguali e con gli angoli retti ( $\text{area} = \text{lato}_1 \times \text{lato}_2$ )
- *parallelogramma*: quadrilatero avente le coppie di lati opposti paralleli ( $\text{area} = \text{base} \times \text{altezza}$ )

- *trapezio*: quadrilatero avente una coppia di lati opposti paralleli (*area* =  $(\text{BASE} + \text{base}) \times \text{altezza}/2$ )

Nel caso di un quadrilatero generico l'area può essere calcolata suddividendo, mediante una diagonale, il quadrilatero in due triangoli.

Nel caso in cui il quadrilatero sia inscrivibile in una circonferenza, l'area  $s$  può essere calcolata mediante la formula di Brahmagupta che si esprime nella forma

$$s = \sqrt{(p-a)(p-b)(p-c)(p-d)}$$

dove con  $a, b, c, d$  sono indicate le misure dei quattro lati del quadrilatero e con  $p$  viene indicato il perimetro. Questa formula generalizza la formula di Erone.

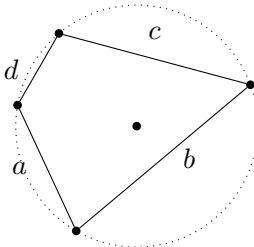


Figura 4.1: Area di un quadrilatero inscritto in una circonferenza.

### 4.3 Area dei poligoni

Per determinare l'area di poligoni con un numero di lati maggiore a quattro, si ricorre alla tecnica di triangolarizzazione del poligono, che consiste nella suddivisione del poligono in triangoli dei quali si calcola l'area ricorrendo generalmente alla formula di Erone.

Nel caso di poligoni definiti in un sistema di assi cartesiani, la formula vista precedentemente per un triangolo può essere estesa ad un generico poligono di  $n$  vertici di coordinate  $(x_i, y_i)$ , con  $i = 1, \dots, n$ ; l'area è espressa dalla formula

$$s = \frac{1}{2} \left| \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) \right|$$

dove  $x_{n+1} = x_1$  e  $y_{n+1} = y_1$ .

### 4.4 Area del cerchio

Archimede (Siracusa 287-212 a.C.), nel suo breve trattato *Misura del cerchio*, ha mostrato come approssimare l'area di un cerchio con quella di un poligono regolare, dimostrando che l'area del cerchio è uguale a quella di un triangolo rettangolo avente come cateti la circonferenza e il raggio del cerchio stesso. Indicato, come tradizione, con  $\pi$  il rapporto delle lunghezze della circonferenza e del corrispondente diametro, l'area  $s$  del cerchio si esprime con

$$s = \pi r^2$$

In questo modo l'area del cerchio viene ricondotta al calcolo del valore di  $\pi$ . Lo stesso Archimede ha fornito una sorprendente stima di  $\pi$ :

$$3 + \frac{10}{71} < \pi < 3 + \frac{10}{70}$$

I tentativi di comprendere la natura del  $\pi$  ha impegnato moltissimi matematici. Un passaggio importante è stato la dimostrazione che  $\pi$  è un numero irrazionale, ossia non è esprimibile come frazione di due numeri naturali, dimostrazione fornita nel 1767 da J.H. Lambert (1728-1777). Dimostrando l'irrazionalità di  $\pi$ , Lambert garantiva che il computo dei suoi decimali non avrebbe mai avuto fine. Nel 1882 F. Lindemann dimostrò che  $\pi$  non solo era irrazionale, ma anche *trascendente* (detto anche *non-algebrico*), ossia non è soluzione di alcuna equazione polinomiale a coefficienti interi. Poichè nessun numero trascendente può essere costruito con riga e compasso, neanche  $\pi$  è costruibile, ponendo fine alla questione della *quadratura del cerchio*, ossia ai tentativi di trasformazione di un cerchio in un quadrato equivalente usando solo riga e compasso; era questo un problema che aveva occupato i matematici dall'epoca di Ippocrate fino ai tempi moderni.  $\pi$  ha resistito anche ai più sofisticati metodi statistici che hanno cercato di individuare qualche regolarità nella successione delle sue cifre; ma invano: le cifre si susseguono all'infinito, in un modo che appare del tutto casuale.

Anche Ramanujan (1887-1920), uno dei più importanti matematici del secolo scorso, ha tentato di esplorare il recondito mondo di  $\pi$ : le sue grandi intuizioni hanno permesso notevoli progressi nello studio del numero, permettendo di mettere a punto efficienti algoritmi di calcolo in grado di quadruplicare il numero di cifre decimali corrette che si ottengono ad ogni ciclo dell'algoritmo; questi metodi, unitamente all'uso di moderni supercalcolatori, hanno permesso di calcolare  $\pi$  fino a milioni di miliardi di cifre decimali.

## 4.5 Un problema di aree

Consideriamo un quarto di cerchio di raggio  $r$ ; dividiamo l'arco in tre parti uguali e dagli estremi di questi archi tracciamo due segmenti paralleli ad uno dei due raggi che costituiscono il bordo. Si delineano così tre zone, come descritto nella figura 4.2; il problema consiste nel determinare l'area della porzione intermedia (ombreggiata nella figura).

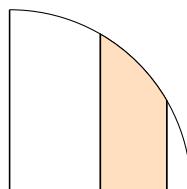


Figura 4.2: Il problema. *Determinare l'area della zona ombreggiata.*

In una situazione così sguarnita, non risulta immediato individuare una linea di

attacco. Per scuotere un po' il problema e partire con qualche considerazione si può tracciare qualche segmento che unisce i punti a disposizione. Unire fra loro i punti estremi degli archi sulla circonferenza non porta lontano. Risulta più interessante unire i punti sulla circonferenza con il centro del cerchio. In questo modo, assegnando anche i nomi ai punti, si arriva alla figura 4.3.

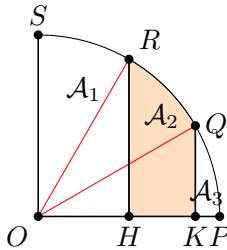


Figura 4.3: La figura completa.

Pur non avendo ancora ben chiaro dove si va a parare, emergono ora alcune considerazioni: i due triangoli  $\triangle OKQ$  e  $\triangle OHR$  sono, per costruzione, rettangoli, hanno entrambi l'ipotenusa lunga  $r$  ed inoltre hanno angoli di ampiezze  $30^\circ, 60^\circ, 90^\circ$ . Di conseguenza sono isometrici. Questa osservazione ci fa prendere coraggio. Volendo si potrebbe trovare facilmente l'area di questi due triangoli visto che sono entrambi la metà di un triangolo equilatero di raggio  $r$ . Prendendo questa strada di calcoli algebrici si arriverebbe alla metà, ma c'è una strada alternativa migliore che si mantiene ad un livello logico, più elevato. Se il triangolo  $\triangle OHR$  viene rimosso dalla zona  $A_1$  ed unito alla zona  $A_3$ , queste due zone si modificano, divenendo entrambe equivalenti ad uno spicchio di cerchio. Di conseguenza anche la zona  $A_2$  avrà un'area pari ad uno spicchio di cerchio. Il ragionamento è tutto qui.

Volendo tradurre queste considerazioni in termini algebrici, indicando con  $t$  l'area dei due triangoli  $\triangle OHR$  ed  $\triangle OKQ$  e con  $v$  l'area di uno spicchio di cerchio e con  $a_1, a_2, a_3$  l'area delle tre zone  $A_1, A_2, A_3$ , si ha

$$a_1 + a_3 = (v + t) + a_3 = v + (t + a_3) = v + v = 2v$$

Pertanto, essendo che  $a_1 + a_2 + a_3 = 3v$ , si ottiene ancora  $a_2 = v$ . Un'equivalente procedimento algebrico è il seguente:

$$a_2 = 2v - \text{area}(\triangle OHR) - a_3 = 2v - \text{area}(\triangle OHR) - (v - \text{area}(\triangle OKQ)) = v$$

## 4.6 Scomposizione di figure

L'area di una generica figura piana soddisfa alla seguente proprietà:

**PROPRIETÀ 1.** L'area di una figura piana  $\mathcal{F}$  scomposta in una sequenza di figure  $(\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n)$  è data dalla formula

$$\text{area}(\mathcal{F}) = \sum_i \text{area}(\mathcal{F}_i)$$

Questa proprietà suggerisce spontaneamente un metodo di calcolo di aree di figure mediante la metodologia top-down.

**Problema 4.6.1** Calcolare l'area del pentagono riportato in figura 4.4, note le lunghezze dei cinque lati, assumendo l'ipotesi che la figura sia convessa e che i due angoli alla base siano retti.

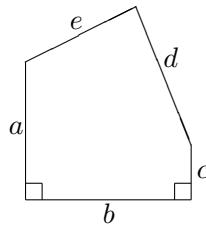


Figura 4.4: Il problema: *Determinare l'area del pentagono.*

**Soluzione.** Una scomposizione della figura in due trapezi come descritto nella figura 4.5 risulterebbe poco utile in quanto comporterebbe il calcolo dell'area di due trapezi rettangoli dei quali sono noti pochi ed insufficienti elementi.

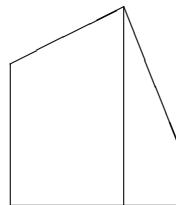


Figura 4.5: Una scomposizione non efficace.

Una scomposizione più efficace è descritta nella seguente figura 4.6.

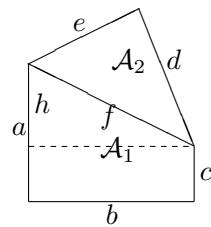


Figura 4.6: Una scomposizione efficace.

Questa scomposizione suggerisce il seguente procedimento risolutivo per il calcolo dell'area  $s$  del pentagono:

$$s \leftarrow \text{area}(\mathcal{A}_1) + \text{area}(\mathcal{A}_2)$$

Adottando la metodologia top-down, i due sottoproblemi corrispondenti al calcolo delle due aree delle due sottofigure  $\mathcal{A}_1$  e  $\mathcal{A}_2$  si esplicitano come segue:

calcolo di  $s_1 = \text{area}(\mathcal{A}_1)$  (area di un trapezio):

$$s_1 \leftarrow (a + c) * b / 2$$

calcolo di  $s_2 = \text{area}(\mathcal{A}_2)$  (area di un triangolo):

$$\begin{aligned} h &\leftarrow |a - c| \\ f &\leftarrow \sqrt{b^2 + h^2} \\ p &\leftarrow (d + e + f) / 2 \\ s_2 &\leftarrow \sqrt{p * (p - d) * (p - e) * (p - f)} \end{aligned}$$

La ricomposizione delle soluzioni parziali sopra riportate conduce all'algoritmo 6.

---

#### Algoritmo 6 - Calcolo dell'area del pentagono riportato nella figura 4.4

---

**Input:** misure  $a, b, c, d, e$  del pentagono

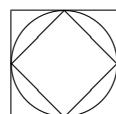
**Output:** area del pentagono

- 1:  $s_1 \leftarrow (a + c) * b / 2$
  - 2:  $h \leftarrow |a - c|$
  - 3:  $f \leftarrow \sqrt{b^2 + h^2}$
  - 4:  $p \leftarrow (d + e + f) / 2$
  - 5:  $s_2 \leftarrow \sqrt{p * (p - d) * (p - e) * (p - f)}$
  - 6:  $s \leftarrow s_1 + s_2$
  - 7: **return**  $s$
- 

*Osservazione.* L'esempio precedente evidenzia che una cattiva scomposizione di un problema rende difficile la soluzione dei sottoproblemi.

## 4.7 Efficacia delle scomposizioni

Determiniamo l'area della figura delimitata dalla circonferenza e dal quadrato interno nota la lunghezza del lato del quadrato esterno:



Una prima scomposizione potrebbe essere:

$$\text{area} = 4 \cdot s$$

dove con  $s$  si denota l'area di uno dei 4 segmenti circolari individuati dal cerchio e dal quadrato più interno che compaiono nella figura precedente. Evidentemente questa scomposizione è poco utile poiché il sottoproblema di trovare

l'area di un segmento circolare è della stessa difficoltà del problema originale. Una scomposizione più efficace è la seguente:

$$\text{area} = \text{circle} - \text{square}$$

in cui si evidenziano i seguenti due sottoproblemi:

$P_1$ : Trovare l'area del cerchio di dato diametro

$P_2$ : Trovare l'area del quadrato di data diagonale

Entrambi questi problemi sono di immediata soluzione. Il sottoproblema  $P_2$  merita un'ulteriore osservazione. Senza scomodare il teorema di Pitagora, l'area di un quadrato di data diagonale può essere determinata seguendo il percorso risolutivo suggerito dalle seguenti uguaglianze geometrico-algebriche fondate su delle proprietà di composizione e scomposizione che valgono per l'operazione di valutazione dell'area di figure piane.

$$\text{square} = \text{square with dashed lines} = 4 \text{ triangle} = 2 \text{ rectangle} = \frac{1}{2} \text{ large square}$$

In definitiva si trova che l'area cercata è esprimibile come segue:

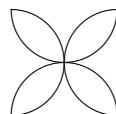
$$\text{area} = \text{circle} - \frac{1}{2} \text{ large square}$$

A questo punto i dettagli dei calcoli risultano banali.

## 4.8 Calcoli geometrici

Il calcolo dell'area di figure può essere svolto in modo geometrico, usando una sorta di *algebra delle figure* basata sulla composizione e scomposizione delle figure.

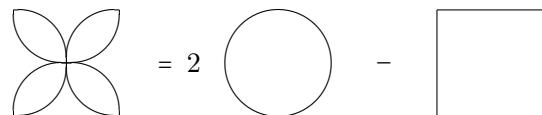
**Problema 4.8.1** Determinare l'area della seguente figura composta da quattro 'petali' ottenuti dall'intersezione di quattro semicerchi, nota la lunghezza del diametro della circonferenza:



**Soluzione.** Senza ricorrere ad alcuna formula di aree, similmente all'esempio precedente, notando che quattro semicerchi che si sovrappongono danno luogo, come intersezione, alla figura in questione, dall'*equazione*



si ricava



e quindi, numericamente, l'area cercata è uguale a

$$2\pi r^2 - 4r^2 = 2r^2(\pi - 2)$$

□

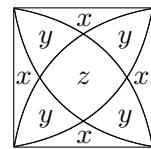
*Osservazione.* In alternativa a quanto è stato fatto nei due esempi sopra proposti, si sarebbe potuto giungere al risultato numerico finale impostando già all'inizio i calcoli numerici; è evidente che un tale approccio sarebbe stato più difficoltoso, richiedendo già all'inizio dei concetti (*area*) e conoscenze (*pi greco*) non strettamente necessarie per l'impostazione del procedimento risolutivo; anche in questo caso, comunque, i vari calcoli numerici sarebbero stati guidati da un'idea risolutiva di fondo, analoga a quella che è stata presentata sopra. Adottando un approccio bottom-up si avrebbe potuto ottenere il risultato del secondo esempio osservando che l'area di questa figura è il doppio di quella dell'esempio precedente, e da ciò si avrebbe conseguito immediatamente il risultato.

*Osservazione.* Dagli esempi sopra proposti e dalle considerazioni appena riportate scaturisce la conclusione che, nello sviluppo della soluzione di un dato problema, è preferibile lavorare il più a lungo possibile con un linguaggio e con un formalismo adeguato al livello del problema trattato; solo in un secondo momento il procedimento risolutivo va espresso dettagliatamente (ad esempio in un linguaggio di programmazione), in una forma pronta da essere sottoposta ad un esecutore.

## 4.9 Calcoli algebrici

In certi casi il calcolo di aree viene svolto in modalità tipica dell'algebra.

**Problema 4.9.1** Calcolare l'area della figura ottenuta dall'intersezione di 4 quarti di cerchio di raggio unitario, come evidenziato nella figura che segue (area  $z$ ).



*Soluzione.* L'area della parte centrale può essere calcolata impostando il seguente sistema lineare:

$$\begin{cases} 4x + 4y + z = 1 \\ 2y + z = \frac{1}{2}\pi - 1 \\ 2x + y = 1 - \frac{1}{4}\pi \end{cases}$$

In questo modo un problema geometrico (calcolo di un'area) viene ricondotto ad un problema algebrico (risoluzione di un sistema lineare) che viene risolto con i metodi tipici dell'Algebra.

## ESERCIZI

**4.1** Analizzare il comportamento della formula di Erone nei seguenti casi:

1. Le misure dei lati non rispettano la condizione di costruibilità di un triangolo.
2. Uno dei lati del triangolo ha lunghezza uguale a 0.
3. tutti e tre i lati hanno la stessa lunghezza (triangolo equilatero).

**4.2** Date le lunghezze  $a, b, c$  dei lati di un triangolo, determinare l'altezza relativa al lato  $a$ .

**4.3** Date le coordinate cartesiane dei vertici di un quadrilatero convesso, determinarne l'area.

**4.4** Determinare l'area di un rombo di cui si conosce la lunghezza  $a$  del lato e la lunghezza  $d$  della diagonale.

**4.5** Determinare l'area di un parallelogramma di cui sono note le lunghezze dei lati e di una diagonale.

**4.6** Determinare il perimetro del rettangolo di perimetro minimo che ricopre un triangolo di lati di lunghezza  $a, b, c$ .

**4.7** Determinare il raggio della circonferenza inscritta in un triangolo di lati di lunghezza  $a, b, c$ .

**4.8** Determinare il raggio della circonferenza circoscritta in un triangolo di lati di lunghezza  $a, b, c$ .

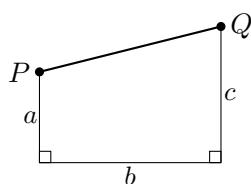
**4.9** Determinare la lunghezza della diagonale di un parallelepipedo rettangolo aventi gli spigoli di lunghezza  $a, b, c$ .

**4.10** Una piramide a base esagonale, avente il lato di base di lunghezza  $k$  e l'altezza  $h$  è appoggiata interamente sopra un cubo di lato  $l$ . Determinare la superficie totale del solido composto così ottenuto.

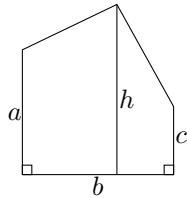
**4.11** Determinare l'area di un quadrilatero di cui sono note le lunghezze  $a, b, c, d$  dei lati ed è noto che l'angolo fra i due lati  $a$  e  $b$  è retto.

**4.12** Dimostrare che l'area di un triangolo avente i vertici di coordinate cartesiane intere è un numero intero o la metà di un numero intero (nonostante le misure dei lati possano essere numeri irrazionali).

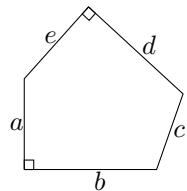
**4.13** Determinare la distanza fra i due punti  $P$  e  $Q$ , note le lunghezze  $a, b, c$  indicate in figura ed essendo retti gli angoli fra i segmenti verticali e la base del quadrilatero.



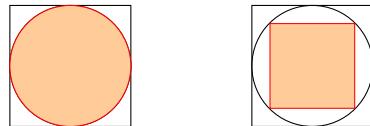
**4.14** Calcolare l'area del seguente pentagono di cui sono note le misure  $a, b, c, h$  riportate in figura.



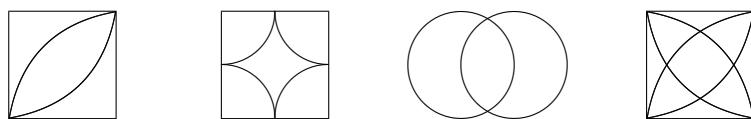
**4.15** Determinare l'area del pentagono descritto nella figura che segue, conoscendo le lunghezze  $a, b, c, d, e$  dei lati e la conoscenza che i due angoli evidenziati sono retti.



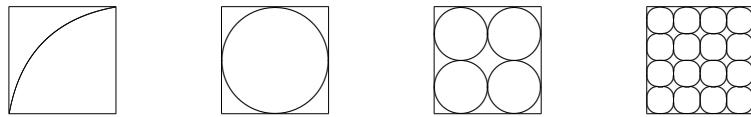
**4.16** Data l'area di un quadrato, determinare le aree del cerchio inscritto e del quadrato inscritto nel cerchio inscritto, come evidenziato nella figura che segue.



**4.17** Determinare l'area della parte più interna delle seguenti figure (sono in ordine di difficoltà crescente). Si assume l'ipotesi che l'esecutore sia in grado di valutare l'area  $c(r)$  di un cerchio di dato raggio  $r$  e l'area  $p(n, r)$  di un  $n$ -poligono regolare di dato raggio  $r$  del cerchio circoscritto. Si richiede di esprimere l'area delle figure come combinazione lineare a coefficienti razionali di aree di cerchi e di poligoni regolari, ossia mediante un'espressione della forma  $\alpha c(r) + \beta p(n, r)$  con  $\alpha, \beta \in \mathbb{Q}$ .

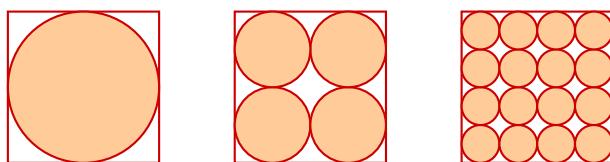


**4.18** Dimostrare l'equivalenza delle seguenti figure costituite da cerchi. Determinarne l'area in funzione della lunghezza  $l$  del lato del quadrato che le racchiude.



**4.19** Si consideri la successione di figure generate mediante la seguente regola ricorsiva:

- la figura di ordine 0 e lato  $h$  è costituita da un cerchio inscritto in un quadrato di lato  $h$
- la figura di ordine  $n$  viene generata suddividendo ciascun quadrato della figura di ordine  $n - 1$  in 4 quadrati in ciascuno dei quali viene generato un cerchio inscritto



Dimostrare che le figure sono equivalenti (ma non isoperimetriche). Definire una funzione ricorsiva per generare una figura della successione.

# 5

---

## DISEGNARE

---

*La geometria della Tartaruga è uno stile di geometria diverso dagli altri, come lo stile assiomatico d'Euclide e lo stile analitico di Cartesio erano anch'essi differenti l'uno dall'altro. Lo stile d'Euclide è logico, quello di Cartesio è algebrico. Lo stile della geometria della tartaruga è informatico.*

S. Papert, *Mindstorm*

In questo capitolo viene descritta la *geometria della tartaruga*, caratterizzata da un automa che si muove su un foglio, lasciando una traccia che genera un disegno. All'interno di molti linguaggi di programmazione è predisposta una componente di linguaggio imperativo costituita dai comandi della grafica della tartaruga, similmente a quanto si ritrova anche nel linguaggio Logo. L'ambiente della tartaruga, nonostante l'apparenza, è molto ricco e versatile: la tartaruga diventa un pretesto su cui realizzare contesti per il disegno, per la geometria e per il calcolo.

## 5.1 Una tartaruga che disegna

Consideriamo un automa, noto con il nome di *tartaruga*, che, muovendosi, lascia una traccia disegnando delle figure. Il piano di movimento della tartaruga è costituito da un foglio virtuale rappresentato dal video del calcolatore; la tartaruga è posizionata sul foglio ed ha una propria direzione di avanzamento (modificabile mediante appositi comandi). Alla tartaruga è attaccata una penna che, muovendosi assieme alla tartaruga, lascia un segno mediante il quale viene generato il disegno. La tartaruga esegue gli ordini che le vengono impartiti: può essere fatta ruotare sul posto, a sinistra ed a destra, e fatta avanzare. A seguire sono schematicamente riportati alcuni comandi di base della grafica della tartaruga che saranno utilizzati più avanti:

|                          |                                       |
|--------------------------|---------------------------------------|
| <i>forward(<i>n</i>)</i> | avanza di <i>n</i> passi              |
| <i>left(<i>a</i>)</i>    | gira a sinistra di un angolo <i>a</i> |
| <i>right(<i>a</i>)</i>   | gira a destra di un angolo <i>a</i>   |

Gli angoli di rotazione sono espressi in gradi sessagesimali. Questi semplici comandi consentono il disegno di figure geometriche, come descritto nell'esempio 5.1.1.

*Esempio 5.1.1 -* Il disegno di un quadrato di lato 50 unità può essere generato mediante il seguente schema di ripetizione di una sequenza di comandi:

---

|                              |
|------------------------------|
| 1: <b>for</b> 4 <b>times</b> |
| 2: <i>forward(50)</i>        |
| 3: <i>left(90)</i>           |
| 4: <b>end for</b>            |

---

Sono predisposti anche i seguenti comandi per gestire l'angolo di avanzamento della tartaruga (riferito ad un prefissato sistema di orientamento assoluto):

|                                    |                                             |
|------------------------------------|---------------------------------------------|
| <i>getang()</i>                    | angolo di avanzamento attuale               |
| <i>setang(<math>\alpha</math>)</i> | imposta ad $\alpha$ l'angolo di avanzamento |

Se si vuole tracciare una linea che inizia in un punto diverso dalla posizione corrente della tartaruga si possono adottare due strategie alternative:

1. sollevare ed abbassare la penna sul foglio; naturalmente, un movimento con la penna abbassata ha l'effetto di lasciare una linea mentre un movimento con la penna alzata non produce alcun effetto grafico. Questa possibilità si fonda sull'uso dei seguenti due comandi:

|                  |                  |
|------------------|------------------|
| <i>penup()</i>   | alza la penna    |
| <i>pendown()</i> | abbassa la penna |

2. far saltare la tartaruga in avanti di un dato numero di passi; il comando corrispondente è il seguente:

|                       |                                   |
|-----------------------|-----------------------------------|
| <i>jump(<i>n</i>)</i> | salta in avanti di <i>n</i> passi |
|-----------------------|-----------------------------------|

*Osservazione.* Per questioni di efficienza e di pulizia grafica, negli esempi, nei problemi e negli esercizi che seguiranno, ammetteremo tacitamente il vincolo che la tartaruga non disegni sopra un tracciato disegnato in precedenza.

*Osservazione.* L'ambiente della tartaruga è primitivo rispetto a qualsiasi forma di concettualizzazione geometrica in quanto non richiede alcuna specifica conoscenza delle proprietà dello spazio geometrico ma solamente una consapevolezza del rapporto del proprio corpo rispetto allo spazio esterno. La caratteristica interessante e peculiare della geometria della tartaruga consiste nella possibilità di poter evitare il ricorso ad un sistema di riferimento di coordinate, in quanto ogni azione grafica fa riferimento alla posizione e direzione attuale della tartaruga. A differenza della tradizionale impostazione assiomatica della geometria euclidea, viene adottata un'impostazione costruttiva; ad esempio, una circonferenza non viene definita dichiarativamente mediante una proprietà caratterizzante (come avviene nella geometria euclidea o cartesiana) ma viene *costruita*: la circonferenza viene specificata mediante un algoritmo che la disegna.

## 5.2 Istruire la tartaruga

L'ambiente grafico predisposto mediante i comandi grafici descritti nel paragrafo precedente può essere arricchito mediante la definizione di nuovi comandi da parte dell'utente. Questa possibilità può essere vista come un meccanismo per istruire la tartaruga, insegnandole nuovi comandi. Questa idea viene concretizzata, a livello teorico, mediante un algoritmo che viene poi codificato in linee di programma mediante un sottoprogramma.

*Esempio 5.2.1 -* Per disegnare un generico esagono poligono regolare di  $n$  lati di lunghezza  $l$  si può definire il seguente algoritmo.

---

### Algoritmo 1 - *poligono*( $n, l$ )

---

**Input:** numero  $n$  di lati, lunghezza  $l$  del lato

```

1: for n times
2: forward(l)
3: left($360/n$)
4: end for
```

---

Per disegnare un esagono di lato 50 basta richiamare il precedente algoritmo mediante l'istruzione

*poligono*(6,50)

□

## 5.3 Disegnare figure ripetitive

Per sviluppare un algoritmo per disegnare una figura composta da un elemento grafico che si ripete è conveniente svolgere delle fasi di analisi che portano ad individuare i seguenti elementi:

1. parametri che definiscono la classe delle figure
2. elementi che si ripetono
3. struttura dello schema ripetitivo

*Esempio 5.3.1* - Le stelle costituiscono delle tipiche strutture ripetitive che si prestano ad essere generate mediante la grafica della tartaruga. A seguire è riportato l'algoritmo per disegnare una stella a 5 punte con lati di lunghezza  $l$ , posti lungo i prolungamenti di un pentagono regolare.



In questo caso:

1. la classe delle figure è individuata dalla lunghezza  $l$  del lato della stella
2. gli elementi grafici che si ripetono sono le "punte" della stella
3. la stella è composta da una sequenza di punte, ciascuna ruotata rispetto alla precedente

In base a queste considerazioni si arriva al seguente algoritmo risolutivo. I valori 144 e 72 degli angoli di rotazione sono ricavabili in base a semplici considerazioni sugli angoli.

---

#### Algoritmo 2 - *stella(l)*

---

**Input:** lunghezza  $l$  del lato

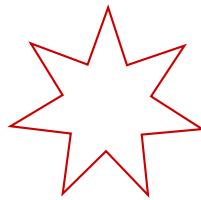
- ```

1: for 5 times
2:   forward( $l$ )
3:   right(144)
4:   forward( $l$ )
5:   left(72)
6: end for

```
-

Problema 5.3.1 Generalizzare il precedente esempio, disegnando una generica stella.

Soluzione. Una generica stella è individuata dal numero n di punte, dalla lunghezza l del lato e dall'angolo a in punta. L'elemento che si ripete è costituito dalla *punta*. La figura che segue illustra una stella a 7 punte.



Il procedimento per disegnare una stella ad n punte si fonda sulle seguenti considerazioni: indicando con $\alpha = 180 - a$ l'angolo di rotazione a sinistra che la tartaruga esegue trovandosi in una punta della stella e con β l'angolo di rotazione a destra quando la tartaruga si trova su uno spigolo interno, deve valere la relazione $n\alpha - n\beta = 360$ da cui si ricava: $\beta = \alpha - 360/n$. Il procedimento è descritto nell'algoritmo 3.

Algoritmo 3 - *stella(n, l, a)*

Input: numero n di punte, lunghezza l del lato, angolo a in punta

- 1: $\alpha \leftarrow 180 - a$
 - 2: $\beta \leftarrow \alpha - 360/n$
 - 3: **for** n **times**
 - 4: *forward(l)*
 - 5: *left(α)*
 - 6: *forward(l)*
 - 7: *right(β)*
 - 8: **end for**
-

□

5.4 Figure ricorsive

La ricorsione trova un fertile terreno di applicazione nell'ambito della grafica. Gli esempi che seguono illustrano alcune famose figure ricorsive.

Esempio 5.4.1 - Un altro famoso esempio di figura ricorsiva è costituito dalla *curva di Von Kock*, definita come segue: la curva di base l ed ordine 0 è costituita da un segmento di lunghezza l ; una curva di ordine 1 viene costruita dividendo il segmento di base in 3 parti, eliminando la parte centrale, sulla quale viene innalzato un triangolo equilatero; in generale una curva di ordine superiore viene costruita replicando questo procedimento, innalzando un triangolo equilatero sulla parte centrale di ogni tratto rettilineo della curva di ordine precedente. La curva di Von Kock ha la caratteristica, di particolare interesse nell'ambito dell'Analisi Matematica, di generare, al tendere all'infinito dell'ordine n di ricorsione, una funzione continua che non è derivabile in alcun punto.

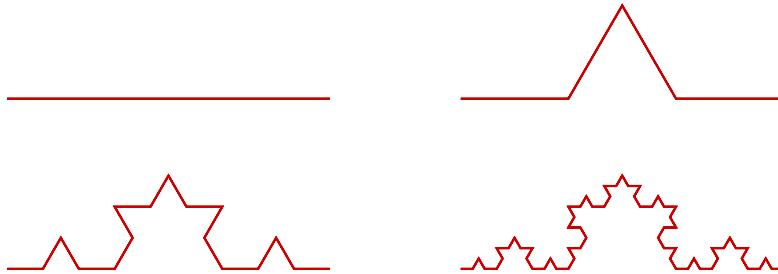


Figura 5.1: Istanze della curva di von Kock di ordine 1, 2, 3 e 4.

L'algoritmo 4 disegna una curva di Von Kock, mediante i comandi della grafica della tartaruga.

Algoritmo 4 - *vonKock*(l, n) : curva di lunghezza l ed ordine n

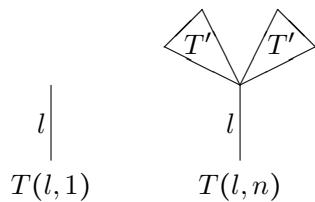
Input: l, n

```

1: if  $n = 0$  then
2:   forward( $l$ )
3: else
4:   vonKock( $l/3, n - 1$ )
5:   left(60)
6:   vonKock( $l/3, n - 1$ )
7:   right(120)
8:   vonKock( $l/3, n - 1$ )
9:   left(60)
10:  vonKock( $l/3, n - 1$ )
11: end if
```

□

Esempio 5.4.2 - Gli *alberi* sono delle particolari figure che si prestano naturalmente ad essere definite in modo ricorsivo. La forma più semplice di albero è costituita dagli *alberi binari* in cui ogni ramo genera due altri rami, di lunghezza pari alla metà del ramo padre che li ha generati e divaricati ad angolo retto. Indicando con $T(l, n)$ un generico albero binario di lunghezza del fusto l ed ordine di ramificazione n , un albero binario può essere descritto ricorsivamente dalla figura 5.2.

Figura 5.2: Schema della definizione ricorsiva di un *albero binario*; T' è un sottoalbero definito come segue: $T' = T(l/2, n - 1)$.

Nella figura 5.3 è riportato un albero binario di lunghezza 2 centimetri (altezza del fusto dell'albero) e di ordine 4.

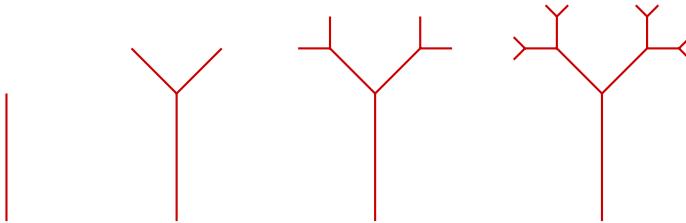


Figura 5.3: Le prime istanze della successione di alberi binari di ordine 1, 2, 3, 4.

L'algoritmo 5 disegna, usando la grafica della tartaruga, un albero mediante una procedura ricorsiva derivata direttamente dalla precedente definizione. Le istruzioni 8 – 12 servono per riportare la tartaruga alla radice dell'albero in modo da raccordare correttamente i due sotto alberi che, ad ogni livello, si biforcano a destra ed a sinistra del fusto.

Algoritmo 5 - *albero*(*l*, *n*) : albero di lunghezza *l* ed ordine *n*

Input: lunghezza *l*, ordine *n*

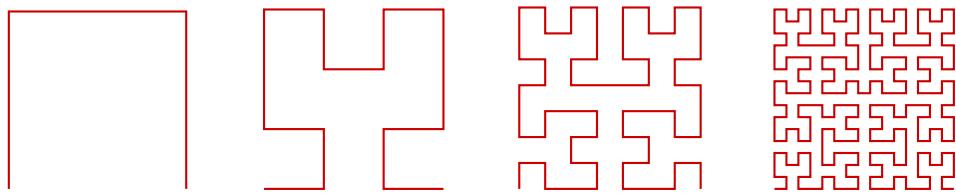
```

1: forward(l)
2: if n > 0 then
3:   left(45)
4:   albero(l/2, n - 1)
5:   right(90)
6:   albero(l/2, n - 1)
7: end if
8: penup()
9: left(180)
10: forward(l)
11: right(180)
12: pendown()

```

□

Esempio 5.4.3 - Nell'ambito dell'Analisi Matematica è famosa la seguente *curva di Hilbert-Peano*, nota con il nome dei matematici che l'hanno scoperta ed analizzata. L'importante caratteristica che la contraddistingue consiste nel generare, al tendere all'infinito dell'ordine di ricorsione, una curva che *riempie* una porzione di piano. Alcune istanze di questa curva sono riportate nella figura 5.4. La curva di Hilbert-Peano può essere disegnata mediante un procedimento ricorsivo descritto nell'algoritmo 6; il parametro *k* = 1 genera una curva a destra rispetto alla direzione attuale, il valore *k* = -1 una curva a sinistra.

Figura 5.4: Istanze della curva di Hilbert-Peano di ordine 1, 2, 3 e 4 (per $k = 1$).

Algoritmo 6 - *hilbert(l, n, k)* : curva di lunghezza l , ordine n e verso k

Input: lunghezza l , ordine n , verso k

```

1: if  $n > 0$  then
2:   left( $k * 90$ )
3:   hilbert( $l, n - 1, -k$ )
4:   forward( $l$ )
5:   right( $k * 90$ )
6:   hilbert( $l, n - 1, k$ )
7:   forward( $l$ )
8:   hilbert( $l, n - 1, k$ )
9:   right( $k * 90$ )
10:  forward( $l$ )
11:  hilbert( $l, n - 1, -k$ )
12:  left( $k * 90$ )
13: end if

```

Esempio 5.4.4 - La curva del drago fu scoperta dal fisico della NASA John Heighway. Fu successivamente analizzata dai matematici e teorici dei computer Donald Knuth e Chandler Davis, con particolare riferimento alle rappresentazioni dei numeri. Il modo più semplice per descriverla si basa su un procedimento grafico ricorsivo, simile al metodo usato per la generazione della curva di Von Kock: si inizia con un segmento che costituisce la curva di ordine 0; ad ogni passo successivo della ricorsione si sostituisce ciascun segmento della curva con due segmenti adiacenti che formano un angolo retto, alternativamente dalle due bande della curva (vedi figura 5.5).

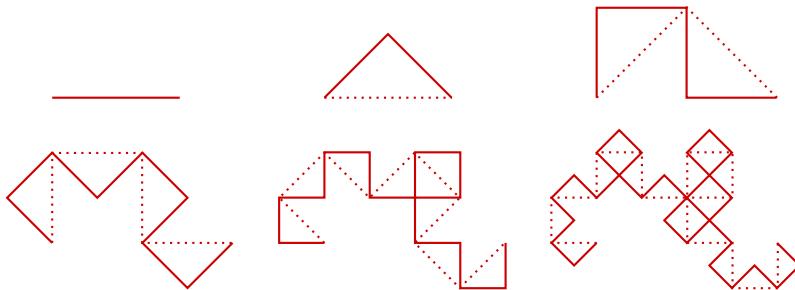


Figura 5.5: Passi della generazione di alcune istanze della curva del drago, dall’istanza di ordine 0 all’istanza di ordine 5; la linea tratteggiata denota la curva di supporto ottenuta al passo precedente.

La figura 5.6 descrive la curva del drago di ordine 6 senza le linee di supporto dell’istanza di ordine 5 e con gli angoli arrotondati per evitare l’apparente incrocio delle linee tangenti fra loro in alcuni spigoli.

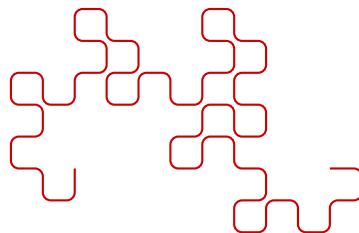


Figura 5.6: Curva del drago di ordine 6.

Il procedimento sopra illustrato suggerisce direttamente l’algoritmo 7.

Algoritmo 7 - *drago*(l, n, k) : curva di lunghezza l , ordine n e verso k

Input: lunghezza l , ordine n , verso k : 0: a sinistra, 1: a destra

```

1: if  $n = 0$  then
2:   forward( $l$ )
3: else
4:   left( $45 + k * 270$ )
5:   drago( $l, n - 1, 0$ )
6:   right( $90 + k * 180$ )
7:   drago( $l, n - 1, 1$ )
8:   left( $45 + k * 270$ )
9: end if

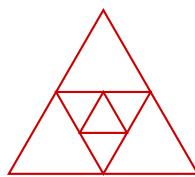
```

Un modo pratico per generare la curva del drago consiste nel prendere un foglio di carta ed eseguire dei successivi piegamenti del foglio a metà: il profilo del foglio, opportunamente dispiegato in modo da formare degli angoli retti in corrispondenza di ciascun piegamento, assumerà, di profilo, la forma di una curva del drago.

5.5 Ripetere e ricorrere

Ripetere e *ricorrere* sono due meccanismi trasversali che risultano potenzialmente equivalenti, nel senso che quello che può essere descritto con l'uno è possibile farlo con l'altro. Nonostante l'equivalenza fra i due meccanismi, dal punto di vista pratico sono sostanzialmente differenti, sia dal punto di vista all'approccio alla soluzione dei problemi sia dal punto di vista della complessità ed efficienza del processo generato.

Esempio 5.5.1 - Vogliamo disegnare con la grafica della tartaruga una sequenza di triangoli equilateri annidati, ciascuno avente i vertici sul punto medio del triangolo in cui è inscritto.



La figura può essere scomposta e considerata formata da una *sequenza di triangoli* ciascuno dei quali avente per vertici i punti medi del triangolo che lo contiene. Questa scomposizione porta facilmente al seguente algoritmo iterativo 8.

Algoritmo 8 - *triangoli(l, n)* : sequenza di n triangoli di lato l

Input: lunghezza l del lato del triangolo esterno, numero n di triangoli

- 1: **for** n times **do**
 - 2: disegna un triangolo di lato l
 - 3: spostati sul punto medio di un lato del triangolo appena disegnato
 - 4: orientati verso un altro punto medio del triangolo appena disegnato
 - 5: dimezza la lunghezza l del lato
 - 6: **end for**
-

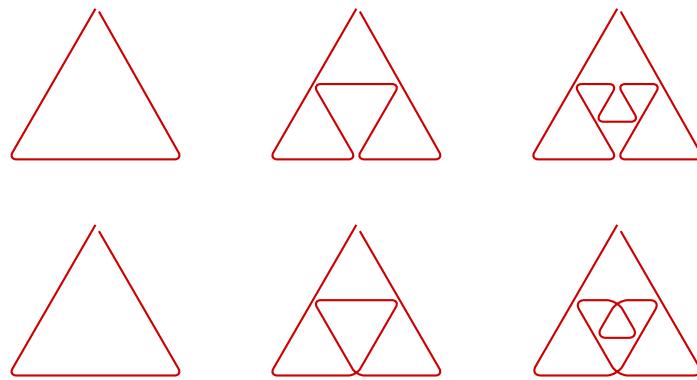
La figura sopra descritta può essere vista, adottando un'ottica ricorsiva, formata da un triangolo esterno in cui è iscritta, appoggiata sui punti medi del triangolo, una figura simile, composta da un triangolo in meno. Questa impostazione porta direttamente al seguente algoritmo ricorsivo 9.

Algoritmo 9 - *triangolazione(l, n)* : triangolazione di lato l ed ordine n

Input: lunghezza l del lato del triangolo esterno, ordine n della figura

- 1: disegna un triangolo di lato l
 - 2: **if** $n > 1$ **then**
 - 3: spostati sul punto medio di un lato del triangolo appena disegnato
 - 4: orientati verso un altro punto medio del triangolo appena disegnato
 - 5: *triangolazione*($l/2, n - 1$)
 - 6: **end if**
-

L'algoritmo 9 risulta poco efficiente perché (alla linea 3.) comporta uno spostamento improduttivo da un vertice del triangolo appena disegnato al punto medio di uno dei suoi tre lati appena disegnati. Per migliorare questo aspetto disegniamo la figura senza passare su un tratto già disegnato, percorrendo un circuito hamiltoniano, ossia senza alzare la penna dal foglio e senza passare su un tratto già disegnato. Questa possibilità è garantita dal fatto che si tratta di un grafo avente tutti i suoi nodi di ordine 4 che è pari; e questo garantisce, per una proprietà dei grafi, che sia possibile disegnare la figura senza alzare la matita dal foglio e senza passare su un tratto già disegnato. Le figure che seguono descrivono due possibili soluzioni di circuito di disegno.



Le due soluzioni suggerite dalle due sequenze di figure sopra descritte risultano equivalenti come risultato finale, ma, al fine di descriverle algoritmamente, la prima presenta la difficoltà che le sottofigure autosimili devono essere alternando il verso di percorrenza. La seconda risulta più facilmente descrivibile, come riportato nell'algoritmo ricorsivo 10.

Algoritmo 10 - *circuitotriangoli(l, n)* : circuito hamiltoniano di triangoli

Input: lunghezza l del lato del triangolo esterno, ordine n della figura

- 1: *forward(l)*
 - 2: *left(120)*
 - 3: *forward(l/2)*
 - 4: *left(60)*
 - 5: **if** $n > 1$ **then**
 - 6: *circuitotriangoli(l/2, n - 1)*
 - 7: **end if**
 - 8: *right(60)*
 - 9: *forward(l/2)*
 - 10: *left(120)*
 - 11: *forward(l)*
 - 12: *left(120)*
-

5.6 La tartaruga si muove nel piano

Una significativa evoluzione della grafica della tartaruga consiste nel denotare con dei nomi identificativi i punti di stazionamento della tartaruga, in modo da poterli utilizzare nel seguito del programma, usando i nomi precedentemente definiti. Questa possibilità si fonda sulla funzione *pos* che fornisce il punto in cui staziona la tartaruga. Per denotare con un nome, ad esempio *P*, il punto di stazionamento attuale della tartaruga viene utilizzata l'istruzione di assegnazione

$$P \leftarrow pos()$$

Similmente a *pos* si può usare la funzione *dpos* che ritorna il punto di stazionamento della tartaruga, con la differenza che un'assegnazione della forma $P \leftarrow dpos()$ comporta che il punto *P* viene automaticamente modificato qualora la tartaruga si muova. Avendo, in questo modo, la possibilità di denotare con dei nomi i punti del piano, risultano coerenti e ben fondate le seguenti istruzioni:

<i>move(P)</i>	muove la tartaruga sul punto <i>P</i> , senza disegnare
<i>draw(P)</i>	sposta la tartaruga sul punto <i>P</i> , disegnando una linea
<i>look(P)</i>	orienta la tartaruga verso il punto <i>P</i> , senza spostarla
<i>dist(P)</i>	distanza della tartaruga dal punto <i>P</i>

Le primitive *move* e *draw* non modificano l'angolo di avanzamento della tartaruga e lo stato su/giù della penna.

Esempio 5.6.1 - Il disegno di un triangolo di dati vertici può essere realizzato mediante il seguente algoritmo.

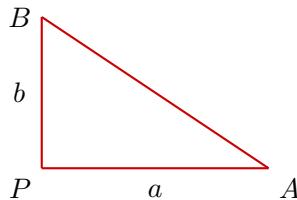
Algoritmo 11 - *triangolo(A, B, C)* - triangolo di vertici *A, B, C*

Input: vertici *A, B, C* del triangolo

- 1: *move(A)*
 - 2: *draw(B)*
 - 3: *draw(C)*
 - 4: *draw(A)*
-

Problema 5.6.1 Disegnare un triangolo rettangolo di dati cateti.

Soluzione. Il disegno di un triangolo rettangolo di cateti *a* e *b* ed avente il vertice dell'angolo retto sul punto attuale di stazionamento della tartaruga può essere realizzato mediante il seguente algoritmo 12.



Algoritmo 12 - *triangoloRettangolo(a, b)* - triangolo rettangolo di cateti a, b

Input: lunghezze a e b dei cateti

- 1: $P \leftarrow pos()$
 - 2: $forward(a)$
 - 3: $A \leftarrow pos()$
 - 4: $move(P)$
 - 5: $right(90)$
 - 6: $forward(b)$
 - 7: $draw(A)$
-

□

Insegnando alla tartaruga delle nuove istruzioni è possibile realizzare un livello di funzionalità in cui la tartaruga non viene più percepita come tale, ma diventa uno strumento invisibile che lascia trasparire delle funzionalità puramente geometriche. Ciò è confermato dai seguenti algoritmi che evidenziano che la tartaruga può essere usata come *strumento di disegno* (algoritmo 13), *strumento di misura* (algoritmo 14) e *strumento di calcolo di punti* (algoritmo 15).

Algoritmo 13 - *segmento(A, B)* - disegno del segmento fra due punti A e B

Input: punti A e B

- 1: $move(A)$
 - 2: $draw(B)$
-

Algoritmo 14 - *distanza(A, B)* - distanza fra due punti A e B

Input: punti A e B

- 1: $move(A)$
 - 2: **return** $dist(B)$
-

Algoritmo 15 - *medio(A, B)* - punto medio fra i due punti A e B

Input: punti A e B

- 1: $move(A)$
 - 2: $look(B)$
 - 3: $jump(dist(B)/2)$
 - 4: **return** $pos()$
-

Osservazione. Una tipica estensione dell'ambiente del piano geometrico consiste nel considerare un sistema di riferimento cartesiano mediante il quale un punto risulta individuato dalla coppia numerica delle sue coordinate; in questo modo si permette l'uso del calcolo algebrico a supporto della soluzione di problemi geometrici nel piano.

5.7 La tartaruga come automa con stato

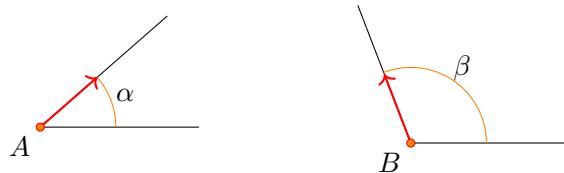
La tartaruga è un *automa con stato*; lo stato è costituito dai seguenti attributi:

- posizione P attuale
- angolo α di avanzamento
- stato s (su/giù) della penna

Oltre a questi attributi, di tipo geometrico, se ne possono considerare altri, di tipo grafico, quali: colore della penna, tipo di linea, ed altri.

Lo stato della tartaruga può essere modificato agendo direttamente su di essa, mediante dei comandi. Gli attributi dello stato della tartaruga sono accessibili tramite le funzioni *pos*, *getang* ed *ispPENDOWN*.

Lo stato della tartaruga che si trova sul punto A , con angolo di avanzamento pari ad α e con la penna nello stato s è definito mediante una terna del tipo (A, α, s) . Per passare dallo stato (A, α, s) allo stato (B, β, t) si può eseguire l'algoritmo 16.



Algoritmo 16 - Passaggio dallo stato (A, α, s) allo stato (B, β, t)

```

1: move(B)
2: setang(β)
3: if  $s \neq t$  then
4:   if  $s$  then
5:     pendown()
6:   else
7:     penup()
8:   end if
9: end if

```

Lo *stato* (posizione, angolo, stato (su/giù) della penna) attuale della tartaruga costituisce una sorta di memoria attuale, istantanea, che serve per gestire il comportamento della tartaruga nell'esecuzione di ciascun comando. Un'evoluzione di questa situazione consiste nel memorizzare in una struttura di memoria la storia di alcuni istanti passati, ciascuno dei quali descritto mediante un'immagine dello stato. Una semplice struttura di memoria adatta per salvare e ripristinare degli stati è costituita da una pila in cui vengono salvati, uno sopra l'altro, i vari stati. Gli stati memorizzati possono essere successivamente ripresi e ristabiliti, a partire dai più recenti ai più vecchi, mediante i

seguenti comandi che memorizzano e recuperano le informazioni in una pila di stati:

<i>init()</i>	riporta la tartaruga allo stato iniziale
<i>push()</i>	salva nella pila lo stato attuale della tartaruga
<i>pop()</i>	ripristina come attuale l'ultimo stato salvato

Ogni comando ha l'effetto di modificare lo stato interno della tartaruga ed eventualmente di produrre un effetto grafico. Come in ogni sistema con memoria, cambiamento di stato ed effetto prodotto dipendono dal comando e dallo stato precedente. La situazione è descritta nella figura 5.7.

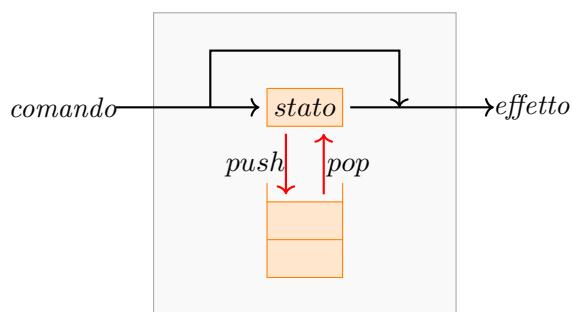
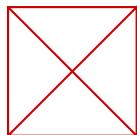
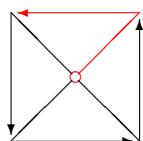


Figura 5.7: Architettura interna dell'automa della grafica della tartaruga.

Problema 5.7.1 Disegnare la seguente figura:



Soluzione. Una soluzione consiste nel percorrere il perimetro del quadrato, memorizzando, strada facendo, in 4 variabili distinte i 4 vertici del quadrato e disegnare le due diagonali congiungendo a coppie i vertici. Questa strategia ha il difetto complicare, se non precludere, un efficace uso di un ciclo e una semplice generalizzazione ad un generico poligono regolare. Una soluzione alternativa consiste nel considerare la figura scomposta come descritta a seguire, dove è evidenziato in rosso l'elemento che si ripete e con un circoletto il punto di partenza della tartaruga:



L'algoritmo che ne deriva è il seguente:

Algoritmo 17 - Disegno di un quadrato con diagonali

Input: lunghezza l della semidiagonale

- 1: **for** 4 times
 - 2: salva lo stato corrente
 - 3: disegna un elemento
 - 4: ripristina l'ultimo stato salvato
 - 5: ruota di 90 gradi
 - 6: **end for**
-

L'algoritmo precedente può essere dettagliato come descritto nell'algoritmo 18.

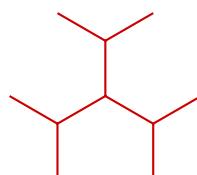
Algoritmo 18 - Disegno di un quadrato con diagonali

Input: lunghezza l del lato

- 1: **for** 4 times
 - 2: $push()$
 - 3: $push()$
 - 4: $jump(l)$
 - 5: $P \leftarrow pos()$
 - 6: $pop()$
 - 7: $right(90)$
 - 8: $forward(l)$
 - 9: $draw(P)$
 - 10: $pop()$
 - 11: $right(90)$
 - 12: **end for**
-

□

Problema 5.7.2 Disegnare un *trifoglio* come riportato nella figura che segue.



Soluzione. Guidati da un principio di simmetria, conviene assumere il punto centrale del trifoglio come punto di partenza della tartaruga; in questo modo risulta più facile descrivere un algoritmo iterativo (algoritmo 17) che fa ricorso alle operazioni *push* e *pop* per il salvataggio dello stato.

Algoritmo 19 - *trifolio(l)* : trifolio di lato di lunghezza l

Input: lunghezza l della semidiagonale

```

1: for 3 times
2:   push()
3:   forward(l)
4:   push()
5:   left(60)
6:   forward(l)
7:   pop()
8:   right(60)
9:   forward(l)
10:  pop()
11:  right(120)
12: end for
```

Nell'algoritmo 19 si può notare che le operazioni *push* e *pop* compaiono come delle parentesi virtuali che racchiudono i comandi di movimento. Indicando con { e } queste parentesi, l'algoritmo può essere scritto in forma compatta, come segue:

$$3 * [\{ \textit{forward}(l), \{ \textit{left}(60), \textit{forward}(l) \}, \textit{right}(60), \textit{forward}(l) \}, \textit{right}(120)]$$

□

Problema 5.7.3 Determinare il k -esimo vertice P_k di un poligono regolare di n lati, percorrendo il poligono in senso antiorario. Sono dati due punti P_1 e P_2 consecutivi del poligono, il numero n dei suoi vertici e l'indice k del punto da determinarsi.

Soluzione. Il procedimento è descritto nell'algoritmo 20.

Algoritmo 20 - vertice P_k

Input: punti P_1 e P_2 iniziali, numero n dei vertici, ordine k del vertice

```

1: move( $P_1$ )
2: look( $P_2$ )
3:  $l \leftarrow \text{dist}( $P_2\math>)$ 
4: repeat  $k - 1$  times
5:   jump( $l$ )
6:   left( $360/n$ )
7: end repeat
8: return pos()$ 
```

□

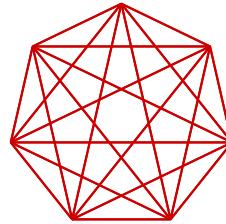
5.8 Disegno di un grafo completo

Arrivati a questo punto abbiamo a disposizione tre modalità operative diverse della grafica della tartaruga:

1. tartaruga elementare, con le sole operazioni *forward*, *left*, *right*, con possibilità di gestire la penna (funzioni *penup*, *pendown*)
2. tartaruga con possibilità di denotazione dei punti (funzioni *pos*, *move*, *draw*)
3. tartaruga come automa con stato, con possibilità di salvare lo stato e di ripristinarlo (funzioni *push*, *pop*)

Il problema che segue ingloba tutte le modalità operative della tartaruga precedentemente esaminate.

Problema 5.8.1 Un *grafo completo* è la figura che si ottiene unendo con dei segmenti i vertici di un poligono regolare in tutti i modi possibili. La figura che segue descrive un grafo completo di 7 vertici. Disegnare un grafo completo.



Soluzione. Indicando con P_1, P_2, \dots, P_n i vertici del poligono, il grafo può essere disegnato basandosi sull'algoritmo 21.

Algoritmo 21 - *grafo(n, l)* : grafo completo - ver.1

Input: numero n dei vertici, lunghezza l del lato

- ```

1: for i from 1 to $n - 1$ do
2: for j from $i + 1$ to n do
3: congiungi il vertice P_i con il vertice P_j
4: end for
5: end for
```
- 

A questo punto rimane da risolvere il problema di determinare un generico punto del poligono; questo problema può essere risolto con approcci diversi. Basandosi sulle operazioni fondamentali della tartaruga i punti del poligono possono essere determinati come segue: il punto  $P_1$  viene inizializzato con la posizione iniziale della tartaruga (linea 2 dell'algoritmo 22); i successivi punti  $P_i$  vengono determinati con un'azione di avanzamento e rotazione (linee 12–13 dell'algoritmo 22); i punti  $P_j$  vengono calcolati con un ciclo di  $j - i - 1$  azioni

di avanzamento e rotazione, a partire da  $P_i$  (linee 5 – 8 dell'algoritmo 22). In questo modo si arriva al definitivo algoritmo 22.

---

**Algoritmo 22 -  $grafo(n, l)$  : grafo completo - ver.2**


---

**Input:** numero  $n$  dei vertici, lunghezza  $l$  del lato

```

1: for i from 1 to $n - 1$ do
2: $P_i \leftarrow pos()$
3: $a \leftarrow getang()$
4: for j from $i + 1$ to n do
5: for $j - i$ times
6: $jump(l)$
7: $left(360/n)$
8: endfor
9: $draw(P_i)$
10: $setang(a)$
11: end for
12: $jump(l)$
13: $left(360/n)$
14: end for
```

---

Usando la tartaruga come automa con stato ripristinabile, il grafo può essere disegnato mediante l'algoritmo 23.

---

**Algoritmo 23 -  $grafo(n, l)$  : grafo completo di  $n$  vertici e lato di lunghezza  $l$** 


---

**Input:** numero  $n$  dei vertici, lunghezza  $l$  del lato

```

1: $\alpha \leftarrow 360/n$
2: for i from 1 to $n - 1$ do
3: $P \leftarrow pos()$
4: $jump(l)$
5: $left(\alpha)$
6: $push()$
7: for j from $i + 1$ to n do
8: $push()$
9: $draw(P)$
10: $pop()$
11: $jump(l)$
12: $left(\alpha)$
13: end for
14: $pop()$
15: end for
```

---

Una soluzione alternativa consiste nel far fare alla tartaruga un giro periferico sul bordo del poligono, memorizzando in un array i vertici del poligono; dopodiché con due cicli *for* annidati si disegna facilmente il grafo completo.  $\square$

## 5.9 La tartaruga come calcolatore

La funzione  $dist(P)$ , che ritorna la distanza della tartaruga da un generico punto  $P$  del piano, permette di utilizzare la tartaruga come un rudimentale strumento di calcolo, combinando le sue capacità di spostamento, rotazione e misurazione di distanza da un punto. La tecnica consiste nell'allontanare la tartaruga dal punto iniziale, seguendo un percorso descritto dall'algoritmo; alla fine si ottiene il risultato misurando la distanza della tartaruga dal punto iniziale. Per muovere la tartaruga senza vedere effetti collaterali grafici è sufficiente muovere la tartaruga con la penna alzata oppure eseguire gli spostamenti mediante il comando *jump*.

**Esempio 5.9.1** - Il calcolo di  $\sqrt{2}$  può essere svolto mediante il seguente algoritmo 24 che si basa sul teorema di Pitagora applicato ad un triangolo rettangolo di cateti di lunghezza unitaria.

---

### Algoritmo 24 - Calcolo di $\sqrt{2}$

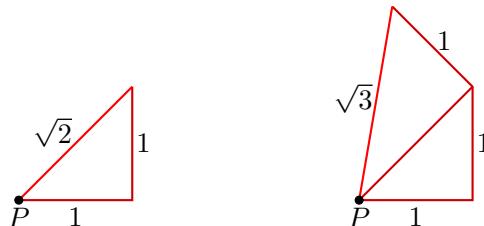
---

**Input:** nessun dato

**Output:** valore di  $\sqrt{2}$

- 1:  $P \leftarrow pos()$
  - 2:  $jump(1)$
  - 3:  $left(90)$
  - 4:  $jump(1)$
  - 5: **return**  $dist(P)$
- 

In modo analogo si può calcolare la  $\sqrt{3}$ , come descritto nella figura che segue.



Il procedimento descritto nell'algoritmo 24 può essere generalizzato per calcolare la radice quadrata di un generico numero naturale, basandosi sulla costruzione della spirale di Teodoro descritta nell'esercizio 5.47. Si lascia per esercizio la stesura dell'algoritmo.

## 5.10 La grafica della tartaruga in modalità ad oggetti

Seguendo la tradizione, come fatto nei precedenti paragrafi, la geometria della tartaruga viene presentata secondo un'impostazione *imperativa*: esiste una tartaruga alla quale si inoltrano i comandi che, eseguiti, producono un effetto su video. Una variante di questa impostazione è fornita dalla modalità *orientata agli oggetti*: anziché rivolgere i comandi *alla* tartaruga, vengono create delle tartarughe, ciascuna avendo un proprio nome, a ciascuna delle quali si può inoltrare un comando, l'esecuzione del quale avrà un effetto che dipende dal particolare stato della tartaruga alla quale è stato rivolto.

La grafica della tartaruga impostata secondo l'impostazione orientata agli oggetti si fonda sui seguenti due aspetti sintattici:

- per creare una tartaruga *t* con specifici argomenti di inizializzazione viene richiamato il costruttore della classe ed utilizzata un'assegnazione:

$$t \leftarrow \text{Turtle}()$$

- per eseguire un'*azione*<sup>1</sup> (comando o funzione) viene utilizzata la notazione puntata

$$t.\text{azione}(\dots)$$

*Esempio 5.10.1* - La seguente porzione di algoritmo disegna un quadrato di lato di 10 unità:

---

```

1: t ← Turtle()
2: for 4 times
3: t.forward(10)
4: t.left(90)
5: end for

```

---

Adottando l'impostazione orientata agli oggetti risulta semplice disegnare delle figure composte da parti, ciascuna delle quali viene disegnata da una diversa tartaruga. L'effetto grafico risulta dinamicamente interessante se le diverse tartarughe si muovono contemporaneamente; questo effetto può essere simulato associando un *thread* di esecuzione a ciascuna tartaruga oppure con un ciclo in cui si fanno eseguire brevi spostamenti a tutte le tartarughe. L'impostazione orientata agli oggetti permette anche di simulare l'interazione fra tartarughe diverse.

*Esempio 5.10.2* - Nella porzione di algoritmo che segue viene creata una sequenza di tartarughe che vengono inizialmente orientate verso diverse direzioni, a raggiera; successivamente vengono fatte avanzare ottenendo l'effetto di una spirale a più rami di diversi colori che si sviluppa dal centro verso l'esterno.

<sup>1</sup>Nella terminologia orientata agli oggetti un'azione viene detta *metodo*.

---

```

1: n ← 17 ▷ numero di tartarughe
2: t ← n * [None]
3: colors ← [RED, GREEN, BLUE]
4: for k from 0 to n - 1 do
5: t[k] ← Turtle()
6: t[k].color(colors[k mod len(colors)])
7: t[k].left(360/n * k)
8: end for
9: for 100 times
10: for k from 0 to n - 1 do
11: t[k].forward(.1) ▷ passi di lunghezza 1
12: t[k].left(2) ▷ rotazione di 2°
13: end for
14: end for

```

---

*Osservazione.* L'impostazione orientata agli oggetti risulta particolarmente produttiva in quanto, fra tante altre interessanti caratteristiche, permette di realizzare in modo semplice e veloce delle nuove classi di oggetti, *estendendo* classi già realizzate, aggiungendo nuovi metodi oppure ridefinendo il comportamento di metodi già realizzati, *ereditando* tutti i metodi già presenti nella classe che si estende. Ad esempio è possibile realizzare una nuova classe *Tarta*, estendendo la classe *Turtle*, aggiungendo nuovi metodi per gestire la velocità di avanzamento delle tartarughe ed ereditando tutti i metodi già presenti nella classe *Turtle*.

*Osservazione.* L'impostazione orientata agli oggetti permette di gestire più stati della tartaruga senza ricorrere a salvare e ripristinare lo stato della tartaruga, in quanto ciascuna tartaruga incapsula il proprio stato.

## ESERCIZI

**5.1** Con riferimento alla geometria della tartaruga, discutere quali delle seguenti equivalenze sono vere:

|                            |            |                                                |
|----------------------------|------------|------------------------------------------------|
| <i>forward</i> ( $x + y$ ) | equivale a | <i>forward</i> ( $x$ ), <i>forward</i> ( $y$ ) |
| <i>left</i> ( $a + b$ )    | equivale a | <i>left</i> ( $a$ ), <i>left</i> ( $b$ )       |
| <i>left</i> ( $a - b$ )    | equivale a | <i>left</i> ( $a$ ), <i>right</i> ( $b$ )      |
| <i>left</i> ( $360 + a$ )  | equivale a | <i>left</i> ( $a$ )                            |
| <i>left</i> ( $360 - a$ )  | equivale a | <i>right</i> ( $a$ )                           |
| <i>left</i> ( $180 + a$ )  | equivale a | <i>right</i> ( $180 - a$ )                     |
| <i>left</i> ( $180 - a$ )  | equivale a | <i>right</i> ( $180 + a$ )                     |

**5.2** Dimostrare che per disegnare un poligono regolare la tartaruga compie delle rotazioni che sommate danno un angolo giro. Dimostrare che questo risultato vale per disegnare un generico poligono (anche non regolare) convesso. Disegnare un poligono regolare di  $n$  lati di lunghezza  $l$ .

**5.3** Disegnare un rettangolo di lati di lunghezze  $a, b$ .

**5.4** Disegnare un triangolo isoscele di base  $b$  ed altezza  $h$ .

**5.5** Disegnare un triangolo di lati di lunghezze  $a, b, c$ .

**5.6** Determinare, senza eseguire al computer, le figure disegnate mediante le seguenti porzioni di algoritmi:

1.

---

```

1: for 8 times
2: forward(l)
3: left(45)
4: forward(l)
5: end for

```

---

2.

---

```

1: for 8 times
2: forward(l)
3: for 4 times
4: left(45)
5: forward(l)
6: end for
7: end for

```

---

**5.7** Determinare, senza eseguire al computer, le figure disegnate mediante le seguenti porzioni di algoritmi:

1.

---

```

1: for 8 times
2: forward(l)
3: push()
4: forward(l)
5: pop()
6: left(45)
7: end for
```

---

2.

---

```

1: for 8 times
2: forward(l)
3: push()
4: left(45)
5: forward(l)
6: pop
7: end for
```

---

**5.8** Mediante la grafica della tartaruga, un poligono regolare può essere disegnato equivalentemente mediante rotazioni a sinistra o a destra. Stabilire la relazione (uguaglianza, equivalenza, isometria, similitudine, simmetria, ...) che esiste fra due poligoni disegnati l'uno con rotazioni a sinistra, l'altro con rotazioni a destra, partendo, nei due casi, dalla stessa posizione ed orientamento iniziale della tartaruga.

**5.9** Sia  $F$  una sequenza di espressioni per disegnare una figura. Stabilire quale relazione (uguaglianza, equivalenza, isometria, similitudine, simmetria, ...) intercorre fra la figura  $F$  e le seguenti:

$F_1: \text{jump}(x), F$   
 $F_2: \text{left}(a), F$   
 $F_3: \text{jump}(x), \text{left}(a), F$   
 $F_4: \text{left}(a), \text{jump}(x), F$

**5.10** Sia  $F(\alpha)$  una figura definita mediante la geometria della tartaruga, con angoli di rotazione di ampiezza  $f(\alpha)$ , essendo  $f$  una funzione lineare di  $\alpha$ . Stabilire le relazioni esistenti fra le seguenti figure:  $F(\alpha)$ ,  $F(\alpha+90)$ ,  $F(\alpha+180)$ ,  $F(\alpha+360)$ ,  $F(\alpha/2)$ .

**5.11** Sia  $F(x)$  una figura composta da segmenti aventi lunghezze proporzionali a  $x$ . Stabilire la relazione che intercorre tra la figura  $F(x)$  e la figura  $F(kx)$ , essendo  $k$  un parametro reale positivo.

**5.12** Disegnare una figura formata da una serie di quadrati concentrici, coassiali ed equispaziati. Calcolare la lunghezza complessiva dei tratti costituenti la figura.

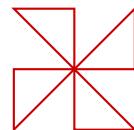
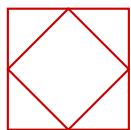
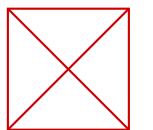
**5.13** Disegnare una figura costituita da una sequenza di triangoli equilateri ciascuno avente per vertici i punti medi del triangolo nel quale è inscritto. Calcolare la lunghezza complessiva dei tratti costituenti la figura.

**5.14** Disegnare una spirale avente lo stesso angolo di rotazione fra i lati ed avente i lati le cui misure costituiscono una progressione aritmetica. Calcolare la lunghezza complessiva dei tratti costituenti la figura.

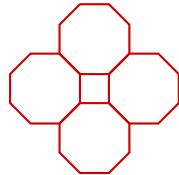
**5.15** Disegnare una  $(n - k)$ -stella avente i vertici su  $n$ -poligono regolare di lato di lunghezza  $l$ , avente i lati costruiti saltando di  $k$  vertici sul poligono; per  $k = 1$  si ottiene un poligono regolare. Analizzare per quali valori di  $n$  e  $k$  la stella passa per tutti i vertici del poligono con un solo percorso ciclico.

**5.16** Determinare un generico punto  $P_k$  di un poligono regolare di  $n$  lati, percorrendo il poligono in senso antiorario. Sono dati due punti  $P_1$  e  $P_2$  consecutivi del poligono, il numero  $n$  dei suoi vertici e l'indice  $k$  del punto da determinarsi.

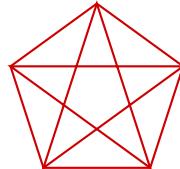
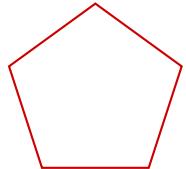
**5.17** Disegnare le seguenti figure. Si richiede di usare negli spostamenti della tartaruga solo numeri naturali o razionali.



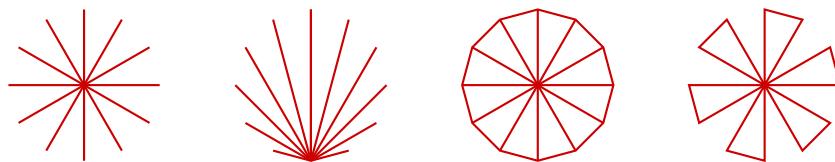
**5.18** Disegnare la seguente figura.



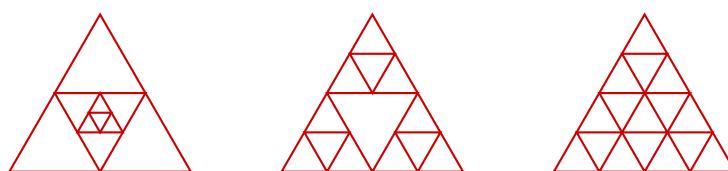
**5.19** Disegnare le seguenti figure composte da segmenti che uniscono i vertici di un pentagono regolare. Generalizzare al caso di un poligono di  $n$  vertici.



**5.20** Disegnare le seguenti figure composte da raggi che uniscono i vertici di un poligono regolare.



**5.21** Disegnare, in modalità iterativa ed in modalità ricorsiva, le seguenti figure composte da una sequenza di triangoli equilateri inscritti ciascuno in un altro ed aventi per vertici i punti medi dei triangoli immediatamente precedente. La prima figura replica il procedimento solo sul triangolo centrale, la seconda su ciascun triangolo agli angoli e la terza su tutti i triangoli.

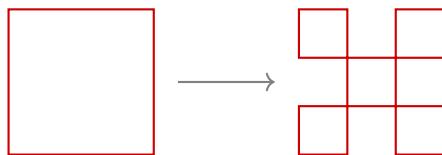


**5.22** Disegnare, in modalità iterativa ed in modalità ricorsiva, le seguenti figure. Determinare la lunghezza di ciascuna figura (mediante un algoritmo o mediante una formula chiusa).



**5.23** Generare un'istanza della successione di figure generate dalla seguente regola ricorsiva:

- una figura di ordine 0 e lato  $h$  è costituita da un quadrato di lato  $h$
- una figura di ordine  $n$  viene generata suddividendo ciascun quadrato della figura di ordine  $n-1$  in 9 quadrati uguali, eliminando i tratti centrali di ciascun lato



**5.24** Una *tassellazione* del piano consiste nella partizione del piano mediante un disegno di figure appartenenti ad una classe predefinita, similmente a come si ricopre un pavimento mediante delle piastrelle di forme prefissate. Definire delle funzioni per generare una tassellazione del piano (una porzione del video) definita mediante le seguenti regole:

- la tassellazione è costituita da quadrati ed ottagoni regolari di uguali lati
- ogni quadrato è circondato da 4 ottagoni (ciascuno su ogni lato del quadrato)
- ogni ottagono è circondato da 4 quadrati (ciascuno alternato sui lati dell'ottagono)

**5.25** Una figura è composta da una sequenza di triangoli uno dentro l'altro, ciascuno avente per vertici i punti medi dei triangoli esterno contiguo, secondo il seguente schema ricorsivo:

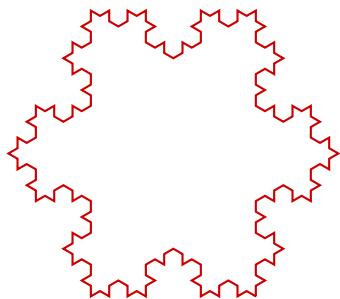
$T_0$  : triangolo assegnato

$T_{n+1}$  : è il triangolo ottenuto da  $T_n$  congiungendone i punti medi dei lati

Definire in forma iterativa ed in forma ricorsiva una funzione che generi la figura sopra descritta.

Determinare il punto a cui tende la successione dei triangoli descritta sopra. Discutere la convergenza del procedimento iterativo.

**5.26** Un *fiocco di von Kock* è ottenuto sostituendo i lati di un triangolo equilatero con tre curve di von Kock. Determinare, in funzione della lunghezza del lato di base e dell'ordine della curva, il perimetro e l'area di un fiocco di von Kock. Valutare questi valori al tendere all'infinito dell'ordine della curva. Disegnare un fiocco di von Kock.



**5.27** Determinare la lunghezza di un albero binario  $T(l, n)$ .

**5.28** La ricorsività unita alla generazione di numeri casuali può costituire la base per ottenere dei disegni e degli effetti grafici gradevoli, molto utilizzati in alcuni campi della *computer graphics*. In particolare gli alberi binari possono essere forme più simili agli alberi botanici agendo sui seguenti aspetti:

- definire un parametro  $k$  che esprime il rapporto fra la lunghezza di un ramo e la lunghezza del suo ramo padre
- definire lo spessore della linea che man mano si assottiglia andando verso le foglie
- aggiungere delle foglie alle estremità dell'albero
- aggiungere delle ramificazioni lungo i fusti
- aggiungere più diramazioni di ciascun ramo
- ricorrere a dei numeri casuali che intervengono in tutti i precedenti aspetti

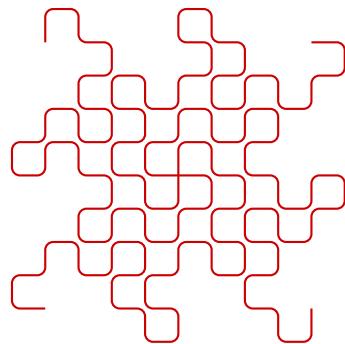
Definire una funzione ricorsiva per generare un *albero casuale*. Tali figure vengono dette *pianze graftali*.

**5.29** Determinare il numero di segmenti che compongono una curva del drago di ordine  $n$ .

**5.30** Determinare la lunghezza di una curva del drago di ordine  $n$  costruita a partire da un segmento iniziale di lunghezza  $l$  (curva di ordine 0).

**5.31** Nonostante l'apparenza, la curva del drago è costituita da una linea continua che non interseca mai se stessa. Definire una funzione che generi una curva del drago in modo da smussare gli spigoli per evidenziare meglio la struttura lineare della curva.

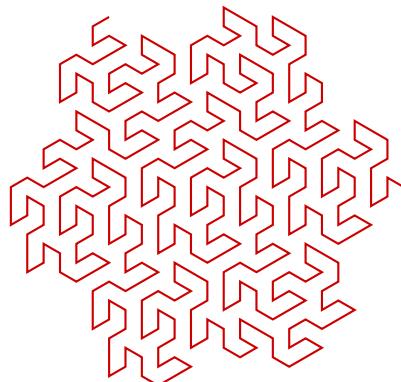
**5.32** Combinando 4 istanze della curva del drago si può ottenere una figura più articolata, come descritto nella seguente figura. Generare queste figure.



**5.33** Individuare gli schemi ricorsivi che generano le seguenti figure. Generare le figure.



**5.34** Individuare lo schema ricorsivo che genera la seguente figura, detta curva di Peano-Gosper. Generare la figura.



**5.35** Adottando l'impostazione orientata agli oggetti, simulare il movimento di due tartarughe: una che si muove lungo una circonferenza ed un'altra che la insegue fino a raggiungerla. Simulare il movimento contemporaneo delle due tartarughe mediante un ciclo all'interno del quale le due tartarughe vengono fatte avanzare di piccoli passi e fare delle piccole rotazioni, oppure associando due diversi *thread* di esecuzione a ciascuna delle due tartarughe.

**5.36** Posizionare 4 tartarughe ai vertici di un quadrato; ciascuna tartaruga è orientata verso quella posizionata al vertice successivo. Muovere contemporaneamente le 4 tartarughe fino a che si incontrano nel centro del quadrato.

**5.37** Alternativamente, in modo più aderente al paradigma *funzionale*<sup>2</sup>, si può vedere un comando rivolto alla tartaruga come un'espressione che, elaborata, produce un risultato che, riprodotto su video, produce un effetto grafico. Una figura diventa una struttura dati (composta da alcuni componenti atomici) sulla quale è possibile fare delle operazioni (traslazione, dilatazione, rotazione, ...). Studiare questo approccio alternativo e realizzarlo in un linguaggio di programmazione.

**5.38** Usando la tartaruga, determinare la lunghezza di un poligono intrecciato costituito da una stella a 5 punte congiungente i vertici di un pentagono regolare di lato 100.

**5.39** Usando la tartaruga calcolare il valore assoluto della differenza (distanza) fra due dati numeri.

**5.40** Usando la tartaruga, dati tre punti  $A$ ,  $B$ ,  $C$ , determinare l'ampiezza dell'angolo  $A\widehat{B}C$ .

**5.41** Usando la tartaruga determinare il perimetro di un triangolo avente due lati di lunghezza  $a$  e  $b$  e l'angolo fra essi compreso di ampiezza  $\alpha$ .

**5.42** Determinare la lunghezza della base di un triangolo isoscele aventi i lati di lunghezza  $l$  e l'angolo al vertice di ampiezza  $a$ .

**5.43** Usando la tartaruga come *calcolatore*, calcolare  $\sqrt{5}$ .

<sup>2</sup>Non si tratta di un'impostazione funzionale pura in quanto il risultato dell'elaborazione dipende dallo *stato* della tartaruga e, d'altra parte, l'elaborazione di un'espressione può modificare lo stato della tartaruga.

**5.44** Usando la tartaruga come *calcolatore*, calcolare la radice quadrata di un generico numero naturale.

**5.45** Usando la tartaruga come *calcolatore*, calcolare il valore delle seguenti espressioni, essendo  $a$  e  $b$  numeri naturali:

1.  $\sqrt{a^2 + b^2}$
2.  $\sqrt{a}$
3.  $a + b\sqrt{2}$
4.  $a + \sqrt{b}$
5.  $\sqrt{a} + \sqrt{b}$

**5.46** Usando la tartaruga come *calcolatore*, calcolare il *seno* ed il *coseno* di un generico angolo di ampiezza  $\alpha$ .

**5.47** Disegnare la *spirale di Teodoro*, così chiamata perché tracciata per la prima volta da Teodoro di Cirene (V secolo a.C.). La spirale viene costruita disegnando un triangolo equilatero isoscele con cateti unitari e utilizzando poi l'ipotenusa dell'ultimo triangolo disegnato come cateto del successivo, l'altro cateto restando unitario, come mostrato nella figura che segue.

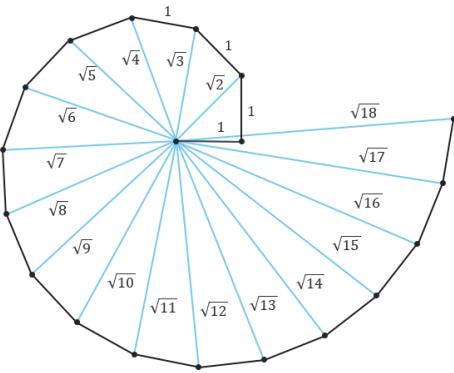


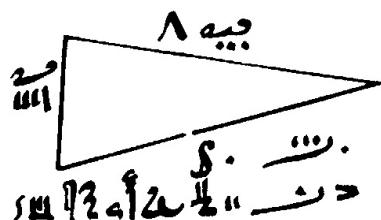
Figura 5.8: I primi 18 lati della spirale di Teodoro.

# 6

---

## TRACCIARE

---



Parte del papiro di Rhind.

Nell'evoluzione del pensiero geometrico un significativo progresso è consistito nel passare dal concetto di *piano su cui disegnare figure* (argomento trattato nel capitolo *Disegnare* dedicato alla geometria della tartaruga) al concetto di piano come ambiente in cui creare figure individuali, denominarle e sulle quali fare operazioni. È ciò che verrà analizzato in questo capitolo, dove i disegni nero su bianco descritti nel capitolo *Disegnare* iniziano a differenziarsi in entità distinte che assumono dei nomi identificativi.

I protagonisti di questo capitolo sono i *punti*, le *rette* e le *circonferenze*. Seguendo le tracce della millenaria tradizione greca, queste figure vengono generate mediante l'uso di due elementari strumenti: la riga (non graduata) ed il compasso; questi strumenti permettono di eseguire operazioni di *tracciamento* e di *intersezione* di linee, in un circolo virtuoso di generazione di nuovi punti e nuove figure. L'approccio di Euclide alla geometria, come descritto in questo capitolo, coinvolge un forte aspetto operativo in quanto le operazioni sulle figure trovano una diretta corrispondenza con le operazioni di disegno delle figure mediante la riga ed il compasso.

## 6.1 Le entità geometriche

Nel seguito non ci preoccuperemo di definire, ma assumeremo note, da come sono state incontrate nella geometria intuitiva, le seguenti entità fondamentali: *punto, retta, piano*.

Prendendo spunto dagli enti geometrici fondamentali individuati da Euclide e nei suoi primi tre postulati, nell'insieme delle figure del piano vengono privilegiate delle classi di figure, particolarmente importanti perché oggetto di studio della geometria elementare e presenti anche in diverse assiomatizzazioni della geometria; queste classi di entità sono descritte nella lista che segue:

$$\text{Point} = \{ \text{punti del piano} \}$$

$$\text{Line} = \{ \text{rette del piano} \}$$

$$\text{Segment} = \{ \text{segmenti del piano} \}$$

$$\text{Circle} = \{ \text{circonferenze del piano} \}$$

Indicheremo con  $\mathcal{G}$  l'insieme di queste entità geometriche del piano, ossia l'unione degli insiemi *Point*, *Line*, *Segment*, *Circle* sopra descritti. Gli elementi di  $\mathcal{G}$  devono essere considerati ancorati al piano nel senso che sono qualificati, altre che dalla loro *forma*, anche dalla loro *posizione* sul piano. In base a questa ipotesi, due elementi di  $\mathcal{G}$  sono *uguali* se e solo se sono sovrapposti (e non solo se sono sovrapponibili). Ad esempio, un quadrato del piano è composto da quattro segmenti *diversi* (*non-uguali*). Sugli elementi di  $\mathcal{G}$  sono definiti gli operatori di confronto di uguaglianza, di diversità e gli operatori di controllo di inclusione (tutti questi operatori sono indotti dalla struttura insiemistica dell'insieme  $\mathcal{G}$ ). Inoltre, sugli elementi di  $\mathcal{G}$  sono definite delle operazioni di natura metrica e geometrica che saranno descritte nel seguito.

## 6.2 Costruzioni con riga e compasso

La riga e il compasso sono strumenti pratici per disegnare; definendo in modo preciso le modalità ed i limiti del loro utilizzo diventano strumenti ideali con cui costruire figure geometriche in modo rigoroso all'interno di un sistema formale di regole, integrandosi perfettamente con la struttura formale e deduttiva della geometria euclidea.

Con *riga* si intende un'asta rigida che permette di tracciare la retta passante per due dati punti e non uno strumento per misurare o segnare distanze. Con *compasso* si intende lo strumento che permette di tracciare una circonferenza individuata dal suo centro e da un punto su di essa: si punta il compasso nel centro, si apre fino a raggiungere con la matita del compasso il punto della circonferenza e si traccia la circonferenza. Il compasso, utilizzato con questa modalità è detto *compasso molle*, dove l'attributo *molle* specifica il fatto che il compasso non può essere usato per prendere e riportare delle distanze. Spesso una circonferenza viene definita assegnandone il centro ed il raggio. Il problema richiede allora di rilevare tale lunghezza con il compasso e poi trasportare il compasso fino a poter puntare nel centro e tracciare con l'apertura data. Con questa modalità d'uso si parla di *compasso rigido*, intendendo che lo strumento è in grado di mantenere inalterata l'apertura impostata.

Nelle costruzioni con riga e compasso i punti sono gli elementi fondamentali per costituire delle entità geometriche più complesse quali segmenti, rette, circonferenze; queste entità non sono altro che insiemi di punti soddisfacenti a specifiche proprietà che le caratterizzano univocamente. Usando i punti come elementi di base si possono generare, mediante tracciamento ed intersezione, altre entità geometriche usando le seguenti operazioni di costruzione:

- a) segmento avente due dati punti come estremi
- b) retta passante per due dati punti distinti
- c) circonferenza avente un dato centro e passante per un dato punto

Si ammette anche l'uso del compasso come rudimentale strumento di misura corrispondentemente all'operazione di puntare il compasso in un dato punto, aprirlo fino ad un altro punto e poi utilizzare questa apertura per riportarla su una linea per costruire un nuovo punto oppure utilizzare tale apertura per disegnare una circonferenza; in pratica si ammette l'uso del compasso rigido. Questa modalità di operare non presuppone che sia definita alcuna unità di misura per determinare la distanza fra due punti. Utilizzando il compasso rigido risulta possibile anche la seguente operazione di costruzione:

- d) circonferenza di dato centro e raggio uguale alla distanza fra due dati punti

Le operazioni a), b), c) e d) sopra descritte possono essere declinate in modo procedurale, coerentemente con l'impostazione imperativa delle costruzioni di figure geometriche su un foglio da disegno; in tale caso è come se ciascuna operazione fosse introdotta dal termine “tracciare”; ad esempio, l'operazione a) assume la seguente formulazione: *tracciare il segmento avente due dati punti come estremi.* Adottando invece un'impostazione di tipo funzionale, coerentemente con l'operatività tipica di molti linguaggi di programmazione, le operazioni sopra descritte possono essere viste come funzioni che restituiscono come risultato l'entità grafica costruita; l'elenco dei prototipi di queste funzioni, dette *costruttori*, è riportato a seguire:

*Segment* :  $\text{Point}^2 \rightarrow \text{Segment}$  segmento avente due dati vertici

*Line* :  $\text{Point}^2 \rightarrow \text{Line}$  retta passante per due dati punti

*Circle* :  $\text{Point}^2 \rightarrow \text{Circle}$  circonferenza di dato centro e  
passante per un dato punto

*Circle* :  $\text{Point}^3 \rightarrow \text{Circle}$  circonferenza di dato centro e raggio  
uguale alla distanza fra due dati punti

Simmetricamente ai costruttori, sono predisposte delle apposite funzioni, dette *selettori*, che, data un'entità geometrica, forniscono gli elementi geometrici che la compongono. I prototipi di queste funzioni sono elencati di seguito.

|               |                                             |                              |
|---------------|---------------------------------------------|------------------------------|
| <i>head</i>   | $: \text{Segment} \rightarrow \text{Point}$ | primo punto di un segmento   |
| <i>tail</i>   | $: \text{Segment} \rightarrow \text{Point}$ | secondo punto di un segmento |
| <i>center</i> | $: \text{Circle} \rightarrow \text{Point}$  | centro della circonferenza   |

Nelle costruzioni geometriche con riga e compasso una tipica operazione consiste nel selezionare un punto (a caso) su una data entità geometrica. Tale operazione viene predisposta tramite il selettore *point* che, applicato ad una generica entità geometrica, fornisce un suo punto come risultato; tale selettore è caratterizzato dal seguente prototipo:

*point* :  $\mathcal{G} \rightarrow \text{Point}$  punto casuale della figura geometrica

Nelle precedenti operazioni ci si è limitati ad operare su un foglio bianco, disponendo di una riga non graduata, di un compasso e di una matita per tracciare le linee, senza usare alcun sistema di riferimento e nessuna unità di misura delle lunghezze, anche se le entità geometriche sono considerate ancorate al piano. Nel caso in cui sia definita l'unità di misura della distanza fra due punti, si può estendere il precedente repertorio di operazioni con le seguenti:

*Circle* :  $\text{Point} \times \mathbb{R} \rightarrow \text{Circle}$  circonferenza di dato centro e raggio

*radius* :  $\text{Circle} \rightarrow \mathbb{R}$  raggio della circonferenza

### 6.3 Intersezione di linee

Una delle operazioni più forti e produttive eseguibili sulle entità geometriche del piano consiste nel generare nuovi punti mediante l'intersezione di linee. Questa operazione si collega alla tradizionale attività geometrica del tracciamento di linee mediante riga e compasso e fornisce un supporto operativo al sistema della tradizionale geometria euclidea. L'operazione di intersezione risulta complementare rispetto all'operazione di costruzione di entità geometriche a partire dai punti, in quanto permette di generare nuovi punti mediante i quali si costruiscono altre entità che, intersecandosi, generano altri punti, secondo lo schema descritto nella figura 6.1.

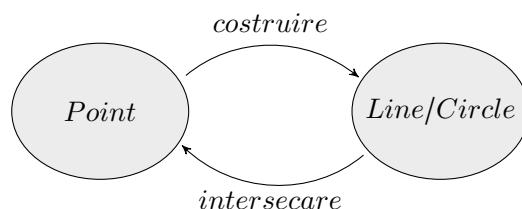


Figura 6.1: Schema operativo per la generazione di entità mediante costruttori ed operazioni di intersezione.

Date due generiche linee  $\alpha, \beta$  del piano (rette, corconferenze, segmenti, segmentirette) indicheremo con  $\alpha \cap \beta$  o con  $\text{inters}(\alpha, \beta)$  l'operazione che determina come risultato i punti di intersezione fra le due linee. Nel caso in cui  $\alpha, \beta$  siano segmenti o semirette i punti di intersezione vengono calcolati con riferimento alla retta di supporto; nel caso in cui entrambe le linee di intersezione siano due rette l'intersezione produce un singolo punto mentre se almeno una delle due linee di intersezione è una circonferenza, l'intersezione produce una coppia di punti. Nel caso le linee non si incrocino (rette parallele o circonferenza che non interseca l'altra linea) adottiamo la convenzione di assumere come risultato il *punto nullo*, denotato con  $\theta$ . Adottiamo, inoltre, la convenzione che ogni operazione coinvolgente il punto nullo come argomento produca il punto nullo.

Il repertorio delle operazioni *a), b), c) e d)* descritto nel paragrafo precedente viene ampliato mediante le seguenti operazioni che generano nuovi punti:

- e) intersezione fra due rette*
- f) intersezione fra due circonferenze o fra una circonferenza ed una retta*

Queste operazioni di intersezione fra entità geometriche danno come risultato, nei casi non particolari, uno o due punti e sono descrivibili mediante le seguenti funzioni:

$$\begin{array}{lll} \cap : \mathcal{L}ine^2 & \rightarrow \mathcal{P}oint & \text{intersezione fra due rette} \\ \cap : \mathcal{C}ircle^2 & \rightarrow \mathcal{P}oint^2 & \text{intersezione fra due circonferenze} \\ \cap : \mathcal{C}ircle \times \mathcal{L}ine & \rightarrow \mathcal{P}oint^2 & \text{intersezione fra circonferenza e retta} \end{array}$$

L'operazione di intersezione fra rette può essere estesa al caso di segmenti, considerando la retta di supporto dei segmenti ed operando poi l'intersezione fra queste rette di supporto; in questo modo si possono considerare anche le seguenti operazioni di intersezione coinvolgenti segmenti:

$$\begin{array}{lll} \cap : \mathcal{S}egment^2 & \rightarrow \mathcal{P}oint & \text{intersezione fra due segmenti} \\ \cap : \mathcal{C}ircle \times \mathcal{S}egment & \rightarrow \mathcal{P}oint^2 & \text{intersezione fra circonferenza e segmento} \end{array}$$

Nei casi in cui l'intersezione non sia possibile (rette parallele, retta esterna alla circonferenza, ...) si assume il *valore nullo* come risultato dell'operazione. L'operazione  $\cap$  di intersezione viene spesso scritta in notazione funzionale con il termine *inters*.

## 6.4 Un linguaggio per le costruzioni geometriche

Due componenti fondamentali per definire un linguaggio sono costituite dagli *enti primitivi* predisposti dal linguaggio e dalle *operazioni* ammesse su questi enti. Gli enti fondamentali della geometria delle costruzioni geometriche sono i *punti* del piano. Sui punti sono previste delle operazioni che vengono suddivise in *costruttori*, *selettori* ed *operatori*: i costruttori (riportati nella tabella 6.1) permettono di costruire nuovi enti componendo degli enti elementari; i selettori (tabella 6.2) hanno una funzione inversa ai costruttori e permettono di accedere alle componenti delle entità; gli operatori (tabella 6.3) costituiscono le operazioni fra entità e generano nuove entità come risultato.

| Costruttore              | Descrizione                                               |
|--------------------------|-----------------------------------------------------------|
| $\text{Segment}(A, B)$   | segmento di estremi i punti $A$ e $B$                     |
| $\text{Line}(A, B)$      | retta passante per i due punti $A$ e $B$                  |
| $\text{Circle}(C, P)$    | circonferenza di centro $C$ e passante per $P$            |
| $\text{Circle}(C, A, B)$ | circonferenza di centro e raggio uguale a $\overline{AB}$ |
| $\text{Circle}(C, r)$    | circonferenza di centro $C$ e raggio $r$                  |

Tabella 6.1: *Costruttori* per un linguaggio delle costruzioni geometriche.

| Selettore               | Descrizione                         |
|-------------------------|-------------------------------------|
| $\text{head}(a)$        | primo estremo del segmento $a$      |
| $\text{tail}(a)$        | secondo estremo del segmento $a$    |
| $\text{center}(\alpha)$ | centro della circonferenza $\alpha$ |
| $\text{point}(\alpha)$  | punto casuale sull'entità $\alpha$  |

Tabella 6.2: *Selettori* per un linguaggio delle costruzioni geometriche.

| Operatore                      | Descrizione                                                        |
|--------------------------------|--------------------------------------------------------------------|
| $\text{inters}(r, s)$          | punto di intersezione fra le rette $r$ ed $s$                      |
| $\text{inters}(\alpha, \beta)$ | punti di intersezione fra le circonferenze $\alpha$ e $\beta$      |
| $\text{inters}(\alpha, r)$     | punti di intersezione fra la circonferenza $\alpha$ e la retta $r$ |

Tabella 6.3: *Operatori* per un linguaggio delle costruzioni geometriche.

Qualora all'entità che viene generata si associa un nome identificativo, corrispondentemente, ad esempio, alla formulazione:

*tracciare la retta  $r$  passante per i due punti  $A$  e  $B$*

viene utilizzata la seguente notazione, corrispondente alla tradizionale operazione di assegnazione:

$$r \leftarrow \text{Line}(A, B)$$

In alcuni algoritmi di costruzioni geometriche risulta necessario ricorrere a comparazioni fra i punti; a questo scopo ammetteremo le seguenti forme sintattiche, di evidente significato:  $A = B$ ,  $A \neq B$ .

**Esempio 6.4.1** - Con  $\text{Segment}(A, B)$  si denota il segmento avente i due punti  $A$  e  $B$  come estremi. I vertici di un segmento vengono considerati in un dato ordine; per evidenziare questa convenzione, al posto del termine *segmento*, è spesso utilizzato il termine *segmento orientato*; il segmento di estremi  $A$  e  $B$ , di cui  $A$  è il primo vertice e  $B$  il secondo viene indicato con la notazione  $AB$  e graficamente viene rappresentato mediante un segmento orientato che parte dal primo vertice e termina con una freccia sul secondo vertice. Con  $\overline{AB}$  o con  $|AB|$  si denota la lunghezza del segmento  $AB$ ; similmente  $\text{Line}(A, B)$  denota la retta passante per i punti  $A$  e  $B$ ,  $\text{Circle}(C, P)$  la circonferenza di centro  $C$  e passante per il punto  $P$ ,  $\text{Circle}(C, A, B)$  la circonferenza di centro  $C$  e raggio uguale alla lunghezza del segmento  $AB$ .  $\square$

## 6.5 Costruzioni geometriche

La seguente definizione precisa in cosa consiste un procedimento di costruzione geometrica mediante riga e compasso.

**DEFINIZIONE 1.** Un punto  $P$  del piano dicesi *costruibile in un solo passo* a partire da un insieme  $\mathcal{A}$  di punti, se è ottenibile mediante intersezione di due rette, di una retta ed una circonferenza o di due circonference costruite con i punti di  $\mathcal{A}$ , mediante le operazioni  $a), b), c), d), e)$  e  $f)$  descritte nei due precedenti paragrafi. Un punto  $P$  dicesi *costruibile* a partire da  $\mathcal{A}$  se esiste una sequenza di  $n$  punti  $(P_1, P_2, \dots, P_n)$  tale che  $P_n = P$  e, per ogni  $i = 1, 2, \dots, n$ , il punto  $P_i$  sia costruibile in un passo a partire dall'insieme  $\mathcal{A} \cup \{P_1, P_2, \dots, P_{i-1}\}$ . *Costruire* un punto significa descrivere una sequenza di operazioni che lo generano.

Nei problemi che seguono sono descritte alcune semplici costruzioni geometriche. Per capire ed interpretare correttamente gli algoritmi è consigliabile riprodurre graficamente su un foglio lo sviluppo del procedimento.

**Problema 6.5.1** Determinare l'asse di un segmento.

**Soluzione.** L'asse di un segmento di estremi  $A$  e  $B$  può essere determinato mediante la costruzione geometrica che ricalca l'usuale metodo di costruzione manuale con riga e compasso ed è descritto nella figura 6.2 e nell'algoritmo 1.

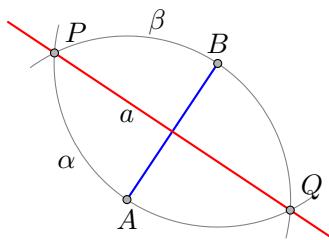


Figura 6.2: Costruzione dell'asse del segmento  $AB$ .

**Algoritmo 1** - Asse del segmento  $AB$  -  $\text{asse}(A, B)$ 

**Input:**  $\text{Point } A, \text{Point } B$   
**Output:** asse del segmento  $AB$

- 1:  $\alpha \leftarrow \text{Circle}(A, B)$
- 2:  $\beta \leftarrow \text{Circle}(B, A)$
- 3:  $P, Q \leftarrow \text{inters}(\alpha, \beta)$
- 4:  $a \leftarrow \text{Line}(P, Q)$
- 5: **return**  $a$

Nell'algoritmo 1 il risultato è stato ottenuto con una sequenza di più passi, costruendo sequenzialmente delle entità geometriche che vengono assegnate temporaneamente a delle variabili. In alternativa, il risultato può essere espresso, in notazione funzionale, mediante un'unica espressione, composta da entità ed operatori geometrici, che, valutata, fornisce il risultato:

$$\text{Line}(\text{inters}(\text{Circle}(A, B), \text{Circle}(B, A)))$$

Il procedimento di una costruzione geometrica può essere concettualmente sintetizzato come un'operazione fra entità geometriche ed espresso mediante un nome, ad esempio, l'algoritmo 1 può essere utilizzato richiamandolo con il nome  $\text{asse}(A, B)$ .  $\square$

In molti casi una costruzione viene usata come sotto procedimento all'interno di altri procedimenti, come evidenziano i seguenti problemi.

**Problema 6.5.2** Determinare il punto medio di un segmento.

**Soluzione.** Il punto medio di un segmento  $AB$  può essere costruito applicando l'algoritmo 1, come descritto nella figura 6.3 e nell'algoritmo 2.

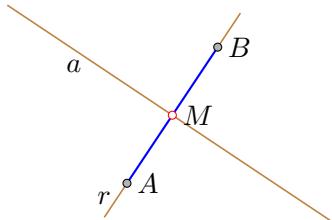


Figura 6.3: Costruzione del punto medio del segmento  $AB$ .

---

**Algoritmo 2** - Punto medio del segmento  $AB$  -  $\text{medio}(A, B)$ 

---

**Input:**  $\text{Point } A, \text{Point } B$ **Output:** punto medio del segmento  $AB$ 

- 1:  $a \leftarrow \text{asse}(A, B)$
  - 2:  $r \leftarrow \text{Line}(A, B)$
  - 3:  $M \leftarrow \text{inters}(r, a)$
  - 4: **return**  $M$
- 

Questo algoritmo può essere condensato nella singola espressione in notazione funzionale

$$\text{inters}(\text{Line}(A, B), \text{asse}(A, B))$$

□

**Problema 6.5.3** Determinare il punto simmetrico di un punto rispetto ad un altro.

**Soluzione.** Il procedimento, illustrato nella figura 6.4 e descritto nell'algoritmo 3, si basa sulla costruzione della circonferenza  $\alpha$  in cui il diametro ha il punto  $A$  come uno dei suoi vertici.

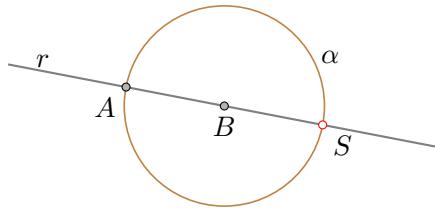


Figura 6.4: Costruzione del punto  $S$  simmetrico di  $A$  rispetto a  $B$ .

---

**Algoritmo 3** - Punto simmetrico rispetto ad un punto -  $\text{simmetrico}(A, B)$ 

---

**Input:**  $\text{Point } A, \text{Point } B$ **Output:**  $\text{Point}$  simmetrico di  $A$  rispetto a  $B$ 

- 1: **if**  $A = B$  **then**
  - 2:   **return**  $A$
  - 3: **end if**
  - 4:  $r \leftarrow \text{Line}(A, B)$
  - 5:  $\alpha \leftarrow \text{Circle}(B, A)$
  - 6:  $P, Q \leftarrow \text{inters}(r, \alpha)$
  - 7:  $S \leftarrow \text{if } A = P \text{ then } Q \text{ else } P$
  - 8: **return**  $S$
- 

□

**Problema 6.5.4** Determinare la retta passante per il punto  $A$  e perpendicolare alla retta  $r$ .

*Soluzione.* Il procedimento è descritto nella figura 6.5 e nell'algoritmo 4.

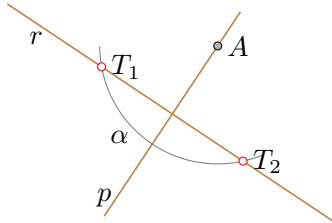


Figura 6.5: Costruzione della retta passante per il punto  $A$  e perpendicolare alla retta  $r$ .

---

**Algoritmo 4** - Retta per  $A$  perpendicolare ad  $r$  -  $\text{perp}(A, r)$

---

**Input:**  $\text{Point } A$ ,  $\text{Line } r$

**Output:** retta per  $A$  perpendicolare ad  $r$

- 1:  $T \leftarrow \text{point}(r)$
  - 2:  $\alpha \leftarrow \text{Circle}(A, T)$
  - 3:  $T_1, T_2 \leftarrow \text{inters}(\alpha, r)$
  - 4:  $p \leftarrow \text{asse}(T_1, T_2)$
  - 5: **return**  $p$
- 

□

**Problema 6.5.5** Determinare la retta passante per il punto  $A$  e parallela alla retta  $r$ .

*Soluzione.* Il procedimento è derivabile direttamente da quello visto nel precedente problema ed è descritto nella figura 6.6.

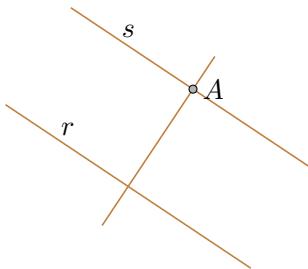


Figura 6.6: Costruzione della retta passante per il punto  $A$  e parallela alla retta  $r$ .

Il problema viene risolto applicando due volte la soluzione vista nel problema 6.5.4, arrivando all'algoritmo 5.

---

**Algoritmo 5** - Retta per  $A$  parallela ad  $r$  -  $\text{paral}(A, r)$ 


---

**Input:**  $\text{Point } A$ ,  $\text{Line } r$

**Output:** retta per  $A$  parallela ad  $r$

1: **return**  $\text{perp}(A, \text{perp}(A, r))$

---

□

**Problema 6.5.6** Determinare il centro della circonferenza passante per tre dati punti non allineati.

**Soluzione.** La semplice costruzione si basa sulla determinazione del punto di intersezione di due assi di due lati del triangolo avente per vertici i tre dati punti, come descritto nella figura 6.7; il procedimento è riportato nell'algoritmo 6.

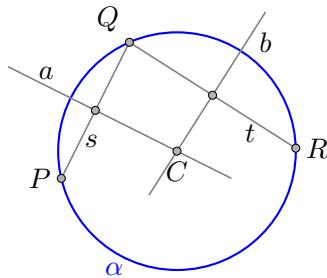


Figura 6.7: Procedimento grafico per la costruzione del centro  $C$  della circonferenza passante per i tre punti  $P, Q, R$ .

---

**Algoritmo 6** - Centro della circonferenza per tre punti -  $\text{centro}(P, Q, R)$ 


---

**Input:**  $\text{Point } P, Q, R$

**Output:**  $\text{Point}$  centro della circonferenza passante per  $P, Q, R$

1:  $a \leftarrow \text{asse}(P, Q)$

2:  $b \leftarrow \text{asse}(Q, R)$

3:  $C \leftarrow \text{inters}(a, b)$

4: **return**  $C$

---

□

**Problema 6.5.7** Determinare il centro di una circonferenza.

*Soluzione.* Per risolvere il problema è sufficiente selezionare tre punti distinti sulla circonferenza e ricondursi al problema precedente. Il procedimento è descritto nell'algoritmo 7.

---

**Algoritmo 7** - Centro di una circonferenza - *centro*( $\alpha$ )

---

**Input:** *Circle*  $\alpha$

**Output:** *Point* centro della circonferenza  $\alpha$

- 1:  $P \leftarrow \text{point}(\alpha)$
  - 2:  $Q \leftarrow \text{point}(\alpha)$
  - 3:  $R \leftarrow \text{point}(\alpha)$
  - 4: **return** *centro*( $P, Q, R$ )
- 

□

**Problema 6.5.8** Dato un punto su una circonferenza, costruire il diametro avente il punto come vertice.

*Soluzione.* Il procedimento illustrato nella figura 6.8 e descritto nell'algoritmo 8 costruisce sulla circonferenza  $\alpha$  il diametro avente il punto  $A$  come uno dei suoi vertici.

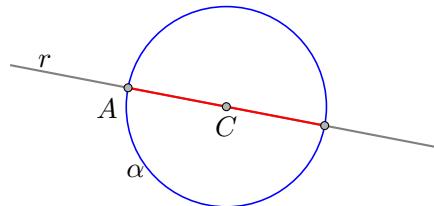


Figura 6.8: Costruzione geometrica per determinare il diametro di una circonferenza avente per estremo un dato punto della circonferenza.

---

**Algoritmo 8** - Diametro di una circonferenza - *diametro*( $\alpha, A$ )

---

**Input:** *Circle*  $\alpha$ , *Point*  $A$

**Output:** diametro della circonferenza  $\alpha$  avente estremo  $A$

- 1:  $C \leftarrow \text{centro}(\alpha)$
  - 2:  $r \leftarrow \text{Line}(A, C)$
  - 3:  $P, Q \leftarrow \text{inters}(r, \alpha)$
  - 4:  $s \leftarrow \text{Segment}(P, Q)$
  - 5: **return**  $s$
- 

□

**Problema 6.5.9** Costruire i punti di tangenza ad una circonferenza condotte da un punto esterno.

**Soluzione.** Indichiamo con  $\alpha$  la circonferenza, con  $P$  il punto esterno e con  $S$  e  $T$  i due punti di tangenza (figura 6.9).

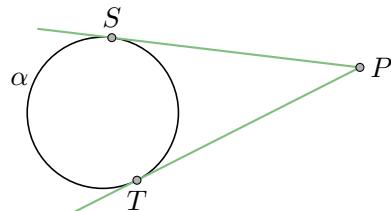


Figura 6.9: Il problema: Data la circonferenza  $\alpha$  ed il punto esterno  $P$ , determinare i punti di tangenza  $S, T$  delle tangenti ad  $\alpha$  condotte da  $P$ .

Il procedimento risolutivo si fonda sull'idea di tracciare la circonferenza di centro il punto medio  $M$  fra  $C$  e  $P$  e notare che i due triangoli  $\triangle CPS$  e  $\triangle CTP$  sono rettangoli in quanto inscritti in una semicirconferenza e richiamare la proprietà che una tangente ad una circonferenza forma un angolo retto con il raggio nel punto di tangenza (figura 6.10).

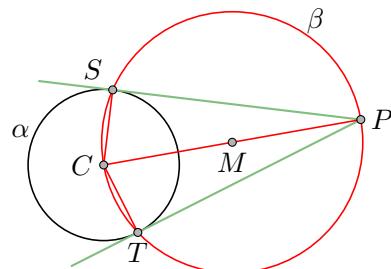


Figura 6.10: La soluzione: Data la circonferenza  $\alpha$  ed il punto esterno  $P$ , costruire i punti di tangenza  $S, T$  delle rette tangenti ad  $\alpha$  condotte da  $P$ .

A seguire è riportato un algoritmo che risolve il problema.

---

**Algoritmo 9** - Rette tangenti ad una circonferenza da un punto esterno

---

**Input:**  $\text{Circle } \alpha, \text{Point } P$

**Output:**  $\text{Line per } P \text{ tangenti a } \alpha$

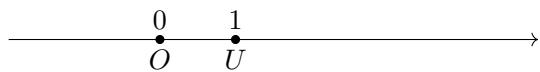
- 1:  $C \leftarrow$  centro della circonferenza  $\alpha$  [algoritmo 7]
  - 2:  $M \leftarrow$  punto medio del segmento  $CP$  [algoritmo 2]
  - 3:  $\beta \leftarrow$  circonferenza di centro  $M$  e passante per  $P$
  - 4:  $S, T \leftarrow$  punti di intersezione fra  $\alpha$  e  $\beta$
  - 5: **return**  $(S, T)$
- 

□

## 6.6 Numeri costruibili

In questo paragrafo viene presentato un interessante caso di simbiosi fra una struttura geometrica (insieme dei punti di una retta) ed una struttura algebrica (insieme dei numeri reali). La situazione è simile a quella che si instaura fra l'algebra e la geometria del piano cartesiano.

Consideriamo una retta  $r$  sulla quale fissiamo due distinti punti base, denotati con  $O$  (origine) ed  $U$  (unità). Su questa retta definiamo un sistema di riferimento cartesiano fondato sui due punti  $O$  ed  $U$ ; in particolare assumiamo che la coordinata di  $O$  sia 0 e la coordinata di  $U$  sia 1; otteniamo una retta orientata  $r$  descritta nella seguente figura:



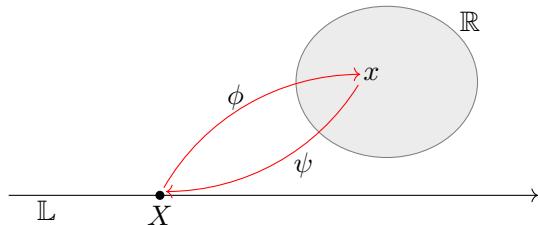
In questo modo risulta univocamente definita una corrispondenza biunivoca  $\phi$  fra l'insieme  $\mathbb{L}$  dei punti della retta e l'insieme  $\mathbb{R}$  dei numeri reali:

$$\phi : \mathbb{L} \rightarrow \mathbb{R}$$

Sulla base di questa corrispondenza adotteremo la convenzione di denotare con  $X$  ed  $x$  due elementi che si corrispondono (il punto e la sua coordinata); in pratica, a livello di denotazione, varrà la seguente corrispondenza biunivoca:

$$\phi(X) = x$$

Indichiamo con  $\psi$  l'inversa della funzione  $\phi$ , ossia  $\psi = \phi^{-1}$ , che fa corrispondere ad ogni numero di  $\mathbb{R}$  un punto della retta  $\mathbb{L}$ .



Il numero  $\phi(X)$  è detto *coordinata* del punto  $X$ , mentre  $\psi(x)$  è il punto della retta avente coordinata  $x$ . Per indicare che il punto  $X$  ha coordinata  $x$  viene spesso usata anche la notazione  $X(x)$ . La biunivocità della funzione  $\phi$  viene evidenziata mediante il seguente schema grafico:



**DEFINIZIONE 2.** Un numero reale  $x$  si dice *costruibile* se il punto  $\psi(x)$  è costruibile con riga e compasso a partire da un dato segmento di lunghezza unitaria.

In virtù della corrispondenza  $\phi$  risultano definite le operazioni sui punti, basandosi sulle corrispondenti operazioni aritmetiche di base  $(+, -, \times, /)$ , in base alla seguente definizione, riferita ad una generica operazione  $\circ$ :

$$P \circ Q \stackrel{\text{def}}{=} \psi(\phi(P) \circ \phi(Q))$$

Per operare geometricamente immersiamo l'insieme  $\mathbb{L}$  nell'insieme dei punti del piano punteggiato, dove abbiamo già visto come eseguire le operazioni geometriche di *tracciamento* con riga e compasso. In pratica è come calcolare numericamente usando riga e compasso, ossia: riga e compasso diventano un calcolatore in grado di eseguire le operazioni aritmetiche di base. Esempi di questa modalità operativa sono presentati nei seguenti problemi.

**Problema 6.6.1** Determinare la somma di due punti  $X_1$  ed  $X_2$ .

**Soluzione.** Il testo del problema va interpretato come segue: Usando le operazioni di tracciamento con riga e compasso, determinare il punto corrispondente alla somma delle coordinate di  $X_1$  ed  $X_2$ , ossia il punto  $\psi(\phi(X_1) + \phi(X_2))$ . L'idea dell'algoritmo che segue si basa sul fatto che il punto  $X_1 + X_2$ , denominato  $Y$ , corrisponde al simmetrico di  $O$  rispetto al punto medio fra  $X_1$  ed  $X_2$ , come indica la figura che segue:



Il calcolo numerico può essere svolto geometricamente, mediante operazioni di tracciamento con riga e compasso, come descritto nell'algoritmo 10.

---

**Algoritmo 10 - Somma di due punti - *somma*( $X_1, X_2$ )**

---

**Input:**  $\text{Point } X_1, X_2$

**Output:**  $\text{Point } X_1 + X_2$

- 1:  $M \leftarrow \text{medio}(X_1, X_2)$
  - 2:  $Y \leftarrow \text{simmetrico}(O, M)$
  - 3: **return**  $Y$
- 

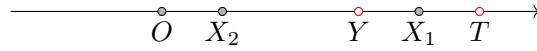
Il procedimento di somma di due punti può essere facilmente adattato per calcolare la somma di due numeri  $x_1, x_2$  come segue:

$$x_1 + x_2 = \psi(\text{somma}(\phi(x_1), \phi(x_2)))$$

□

**Problema 6.6.2** Determinare la differenza di due punti  $X_1$  ed  $X_2$ .

*Soluzione.* Ragionando sulla figura che segue:



si ricava l'algoritmo 11.

---

**Algoritmo 11 - Differenza di due punti - somma( $X_1, X_2$ )**

---

**Input:** Point  $X_1, X_2$

**Output:** Point  $X_1 - X_2$

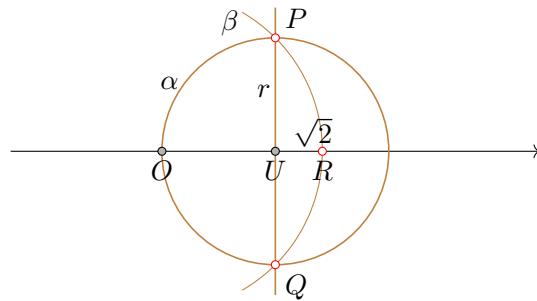
- 1:  $T \leftarrow \text{somma}(X_1, X_2)$
- 2:  $Y \leftarrow \text{simmetrico}(T, X_1)$
- 3: **return**  $Y$

□

Le operazioni geometriche di tracciamento consentono di eseguire calcoli numerici, in particolare calcolare valori specifici.

**Problema 6.6.3** Calcolare  $\sqrt{2}$ .

*Soluzione.* Il procedimento è descritto nella figura che segue, dove  $\overline{OU} = \overline{UP} = 1$ . Si ricava così l'algoritmo 12.




---

**Algoritmo 12 - Calcolo di  $\sqrt{2}$**

---

**Input:** nessuno

**Output:**  $\sqrt{2}$

- 1:  $\text{asse} \leftarrow \text{Line}(O, U)$
- 2:  $r \leftarrow \text{perp}(U, \text{asse})$
- 3:  $\alpha \leftarrow \text{Circle}(U, O)$
- 4:  $P, Q \leftarrow \text{inters}(\alpha, r)$
- 5:  $\beta \leftarrow \text{Circle}(O, P)$
- 6:  $R, \_ \leftarrow \text{inters}(\beta, \text{asse})$
- 7:  $\text{rad2} \leftarrow \phi(R)$
- 8: **return**  $\text{rad2}$

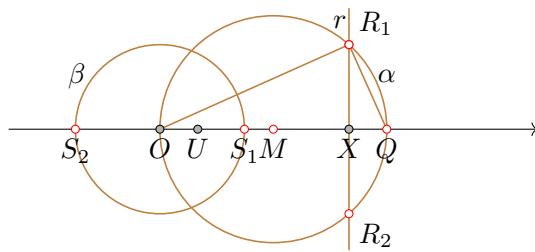
□

Il problema precedente può essere generalizzato come segue.

**Problema 6.6.4** Calcolare  $\sqrt{x}$ ,  $x \in \mathbb{R}$ .

**Soluzione.** Il procedimento esordisce determinando il punto  $X$  corrispondente al valore  $x$ , ossia  $X = \psi(x)$ , ed il punto  $Q$  tale che  $\overline{XQ} = 1$ . Il fulcro del procedimento si basa sull'applicazione del secondo teorema di Euclide che, con riferimento al triangolo rettangolo  $ORQ$  inscritto nella semicirconferenza  $\alpha$  come descritto nella figura che segue, si esprime mediante l'uguaglianza

$$\overline{OX} \cdot \overline{XQ} = \overline{XR_1}^2$$



Essendo  $\overline{XQ} = 1$ , conseguente che  $\overline{XR_1} = \sqrt{x}$ . Il procedimento dettagliato e completo è riportato nell'algoritmo 13. Le linee 7–8 dell'algoritmo servono per traslare il segmento  $XR_1$  sull'asse dei numeri, a sovrapporsi al segmento  $OS_1$ . Si noti come la funzione  $\psi$  serva per inserire il numero  $x$  nel "calcolatore riga-compasso" e come, simmetricamente, la funzione  $\phi$  permetta di convertire in numero il punto ottenuto come risultato dell'elaborazione.

---

#### Algoritmo 13 - Calcolo di $\sqrt{x}$

---

**Input:** numero  $x$

**Output:** numero  $\sqrt{x}$

- 1:  $X \leftarrow \psi(x)$
- 2:  $Q \leftarrow \text{somma}(X, U)$
- 3:  $M \leftarrow \text{medio}(O, Q)$
- 4:  $\text{asse} \leftarrow \text{Line}(O, U)$
- 5:  $r \leftarrow \text{perp}(X, \text{asse})$
- 6:  $\alpha \leftarrow \text{Circle}(M, O)$
- 7:  $R_1, R_2 \leftarrow \text{inters}(\alpha, r)$
- 8:  $\beta \leftarrow \text{Circle}(O, X, R_1)$
- 9:  $S_1, S_2 \leftarrow \text{inters}(\beta, \text{asse})$
- 10:  $\text{radx} \leftarrow \phi(S_1)$
- 11: **return**  $\text{radx}$

□

## 6.7 Costruzioni con il solo compasso

Nel contesto delle costruzioni geometriche è naturale attendersi che l'insieme delle figure costruibili aumenti all'aumentare del numero e della potenza degli strumenti ammessi. I greci, più di due millenni fa, utilizzando solamente riga e compasso erano riusciti a costruire molteplici figure geometriche e presero i risultati di questi procedimenti a fondamento della geometria. Usando altri strumenti, oltre alla classica coppia riga-compasso, erano riusciti a risolvere il problema della duplicazione del cubo. Nel XVII secolo si iniziò a chiedersi quali punti fossero costruibili mediante il solo uso del compasso, ossia ammettendo di poter tracciare circonferenze e determinarne i punti di intersezione e nel considerare una retta definita conoscendone due suoi punti distinti.

Nel 1797 Lorenzo Mascheroni pubblica la *Geometria del Compasso*, opera in cui decide di ripetere le costruzioni della geometria Euclidea con il solo compasso. Con questa limitazione che evita la possibilità di tracciare linee (e di determinare i punti di intersezioni con delle rette) si può ancora considerare il concetto di retta che viene considerata definita mediante due suoi punti. In questo testo viene data data dimostrazione del seguente risultato:

**TEOREMA 1.** (di Mascheroni–Mohr) Ogni costruzione eseguibile con riga e compasso è eseguibile con il solo compasso.

In realtà questo risultato è stato dimostrato per la prima volta dal matematico danese Georg Mohr che lo pubblicò nel 1672 in un testo che venne pubblicato solo in danese ed olandese e rimase, per questo, sconosciuto alla comunità scientifica fino al 1928, quando ne venne trovata una copia in un negozio di libri usati e venne divulgato.

Non esiste un “analogo” del risultato di Mascheroni–Mohr relativamente all’uso della sola riga, in quanto ci si può convincere facilmente che usando solo la riga si possono costruire solo linee rette e le intersezioni restano all’interno del campo di definizione di queste rette. Un risultato intermedio è stato dimostrato Jacob Steiner nel 1833, ed è riassunto nel seguente teorema.

**TEOREMA 2.** (di Steiner) Tutte le costruzioni con riga e compasso sono effettuabili con la sola riga nell’ipotesi che sia data anche una circonferenza (fissa) con il suo centro.

Le dimostrazioni dei teoremi di Mascheroni–Mohr e di Steiner sopra riportati possono essere trovate nei testi di geometria. Nei testi si trova anche la dimostrazione che nel teorema di Steiner non è possibile prescindere dalla conoscenza del centro: se è data solo la circonferenza senza il centro, non si possono più effettuare tutte le costruzioni.

A seguire è riportato l’esempio di un problema risolto mediante l’uso del solo compasso.

**Problema 6.7.1** Costruire il doppio di un segmento usando il solo compasso.

**Soluzione.** Consideriamo il segmento  $AB$  e determiniamo il punto  $F$  in modo che risulti  $\overline{AF} = 2 \cdot \overline{AB}$ . La soluzione è descritta nella figura 6.11 e nell'algoritmo 14.

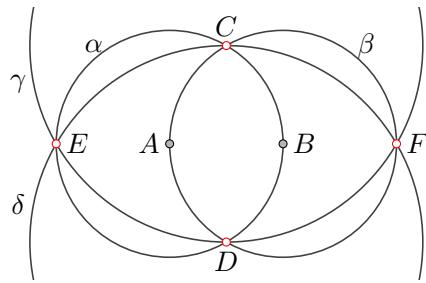


Figura 6.11: Duplicazione di un segmento:  $\overline{AF} = 2 \cdot \overline{AB}$ .

---

#### Algoritmo 14 - Duplicazione di un segmento

---

**Input:**  $\text{Point } A, B$

**Output:**  $\text{Point } F$  tale che  $\overline{AF} = 2 \cdot \overline{AB}$

- 1:  $\alpha \leftarrow \text{Circle}(A, B)$
  - 2:  $\beta \leftarrow \text{Circle}(B, A)$
  - 3:  $C, D \leftarrow \text{inters}(\alpha, \beta)$
  - 4:  $\gamma \leftarrow \text{Circle}(C, D)$
  - 5:  $\delta \leftarrow \text{Circle}(D, C)$
  - 6:  $F, \_ \leftarrow \text{inters}(\gamma, \delta)$
  - 7: **return**  $F$
- 

Una soluzione alternativa è descritta nella figura 6.12 e nell'algoritmo 15.

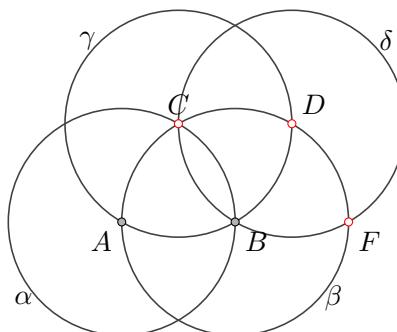


Figura 6.12: Duplicazione di un segmento:  $\overline{AF} = 2 \cdot \overline{AB}$ .

---

**Algoritmo 15** - Duplicazione di un segmento:  $\overline{AF} = 2 \cdot \overline{AB}$ 

---

**Input:**  $\text{Point } A, B$ **Output:**  $\text{Point } F$  tale che  $\overline{AF} = 2 \cdot \overline{AB}$ 

```

1: $\alpha \leftarrow \text{Circle}(A, B)$
2: $\beta \leftarrow \text{Circle}(B, A)$
3: $C, _ \leftarrow \text{inters}(\alpha, \beta)$
4: $\gamma \leftarrow \text{Circle}(C, B)$
5: $D, _ \leftarrow \text{inters}(\gamma, \beta)$
6: $\delta \leftarrow \text{Circle}(D, B)$
7: $F, _ \leftarrow \text{inters}(\delta, \beta)$
8: return F

```

---

□

**6.7.1 Il problema del compasso molle**

Il "problema del compasso molle" consiste nel descrivere come, utilizzando riga e compasso "molle", è possibile costruire una circonferenza una volta che siano assegnati il centro ed un segmento qualsiasi del piano di lunghezza pari al raggio, autorizzando in questo modo un utilizzo accettabile del compasso "rigido", all'interno della teoria. Il problema può essere formulato come segue:

*Problema del compasso molle.* Dati tre punti  $A, B, C$  tracciare con il compasso molle la circonferenza di centro  $C$  e raggio  $\overline{AB}$ .

A seguire è riportato un algoritmo che risolve il problema.

---

**Algoritmo 16** - Problema del *compasso molle*

---

**Input:**  $\text{Point } A, B, C$ **Output:**  $\text{Circle}$  di centro  $C$  e raggio  $\overline{AB}$ 

```

1: $\alpha \leftarrow \text{Circle}(A, C)$
2: $\beta \leftarrow \text{Circle}(C, A)$
3: $D, E \leftarrow \text{inters}(\alpha, \beta)$
4: $\delta \leftarrow \text{Circle}(D, B)$
5: $\eta \leftarrow \text{Circle}(E, B)$
6: $H, K \leftarrow \text{inters}(\delta, \eta)$
7: $T \leftarrow \text{if } \overline{AB} = \overline{CH} \text{ then } H \text{ else } K$
8: $\gamma \leftarrow \text{Circle}(C, T)$
9: return γ

```

---

L'algoritmo 16 è descritto nella figura 6.13.

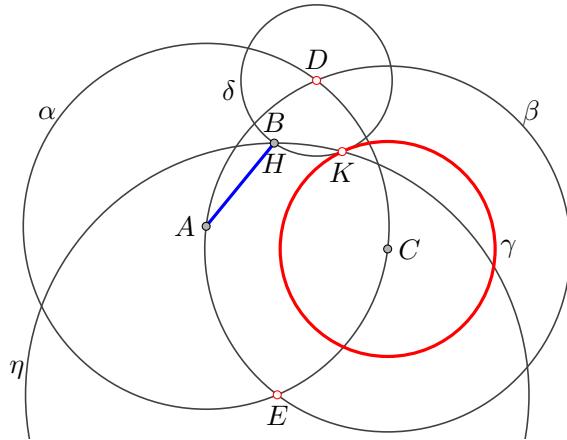


Figura 6.13: Soluzione del problema del *compasso molle*:  $\gamma$  è la circonferenza di centro  $C$  e raggio  $AB$ .

Utilizzando il risultato fornito dalla soluzione del problema del compasso molle risulta lecita l'operazione di tracciare una circonferenza di dato centro e raggio pari alla lunghezza di un dato segmento; equivalentemente, dati tre generici punti  $A, B, C$  risulta lecita l'operazione di puntare in  $C$  con apertura  $AB$ . Indicheremo questa operazione con  $\text{Circle}(C, A, B)$ .

Vediamo ora un'applicazione del teorema del compasso molle.

**Problema 6.7.2** Determinare i punti di intersezione fra una circonferenza ed una retta non passante per il centro della circonferenza.

**Soluzione.** Consideriamo una retta  $r$  individuata dai due punti  $A, B$  ed una circonferenza  $\alpha$  di centro  $C$  e passante per  $P$ . La soluzione del problema è descritta nella figura 6.14 e nell'algoritmo 17.

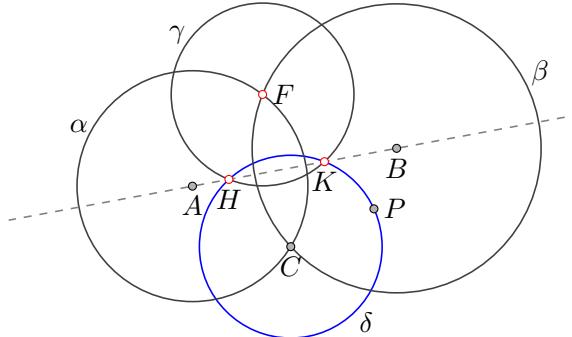


Figura 6.14: Intersezione retta-circonferenza.

---

**Algoritmo 17** - Intersezione retta-circonferenza -  $\text{rettaxcirc}(A, B, C, P)$ 

---

**Input:**  $\text{Point } A, B, C, P$ **Output:**  $\text{Point } H, K$ 

```

1: $\alpha \leftarrow \text{Circle}(A, C)$
2: $\beta \leftarrow \text{Circle}(B, C)$
3: $F, E \leftarrow \text{inters}(\alpha, \beta)$
4: $T \leftarrow \text{if } F = C \text{ then } E \text{ else } F$
5: $\gamma \leftarrow \text{Circle}(T, C, P)$
6: $\delta \leftarrow \text{Circle}(C, P)$
7: $H, K \leftarrow \text{inters}(\gamma, \delta)$
8: return H, K

```

---

La soluzione presentata può essere facilmente adattata anche al caso in cui la retta passi per il centro della circonferenza.  $\square$

### 6.7.2 Il problema di Napoleone

Un famoso problema, proposto dal generale Napoleone Bonaparte, consiste nel determinare, mediante riga e compasso, il centro di una circonferenza. Il problema venne segnalato al matematico italiano Lorenzo Mascheroni (1750-1800) che lo risolse e ne comunicò la soluzione allo stesso Napoleone il quale la espose con orgoglio al matematico francese Pierre-Simone Laplace (1749-1827) che era stato suo professore di matematica. Mascheroni generalizzò il problema di Napoleone vincolando le costruzioni all'uso del solo compasso; il problema è noto proprio in questa riformulazione e si enuncia come segue:

*Problema di Napoleone.* Determinare il centro di una data circonferenza usando il solo compasso.

In notazione algoritmica la soluzione di Mascheroni è sintetizzata nell'algoritmo 18.

**Algoritmo 18** - Problema di Napoleone - *centro*( $\alpha$ )**Input:** Circle  $\alpha$ **Output:** Point  $C$ 

- 1:  $A \leftarrow point(\alpha)$
- 2:  $T \leftarrow point(\alpha)$
- 3:  $\beta \leftarrow Circle(A, T)$
- 4:  $B, C \leftarrow inters(\alpha, \beta)$
- 5:  $\gamma \leftarrow Circle(B, A)$
- 6:  $\delta \leftarrow Circle(C, A)$
- 7:  $D_1, D_2 \leftarrow inters(\gamma, \delta)$
- 8:  $D \leftarrow \text{if } A = D_2 \text{ then } D_1 \text{ else } D_2$
- 9:  $\zeta \leftarrow Circle(D, A)$
- 10:  $E, F \leftarrow inters(\beta, \zeta)$
- 11:  $\eta \leftarrow Circle(E, A)$
- 12:  $\theta \leftarrow Circle(F, A)$
- 13:  $G_1, G_2 \leftarrow inters(\eta, \theta)$
- 14:  $G \leftarrow \text{if } A = G_2 \text{ then } G_1 \text{ else } G_2$
- 15: **return**  $G$

L'algoritmo 18 corrisponde ad una costruzione abbastanza articolata che può essere seguita passo dopo passo facendo riferimento alla figura 6.15.

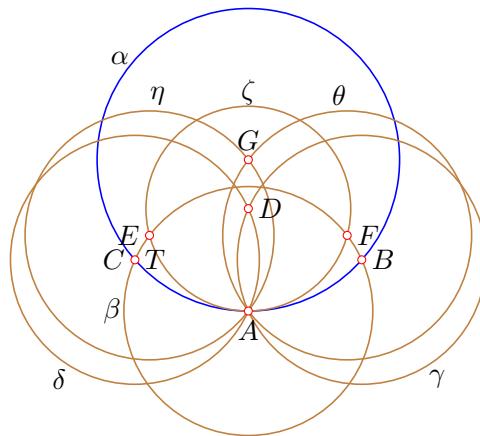


Figura 6.15: Soluzione del problema di Napoleone.

## ESERCIZI

### 6.1 Costruire

1. il punto simmetrico di un punto rispetto ad una retta
2. la proiezione perpendicolare (piede) di un punto su una retta
3. il simmetrico di un segmento rispetto ad una retta

### 6.2 Dati i vertici di un triangolo, costruire

1. il baricentro
2. il circocentro
3. l'incentro

**6.3** Dati due punti  $A$  e  $B$ , costruire il punto  $C$  vertice del triangolo equilatero ministrorso  $\triangle ABC$ .

**6.4** Dati due punti  $A$  e  $B$ , costruire due punti  $C$  e  $D$  in modo che i vertici  $A, B, C, D$  costituiscano, in senso antiorario, i vertici di un quadrato.

**6.5** Dati due punti distinti  $A$  e  $B$ , costruire il quadrato avente il segmento  $AB$  come diagonale.

**6.6** Dati due punti distinti  $A$  e  $B$  del piano, costruire due circonferenze di uguale raggio, fra loro tangenti esternamente nel punto medio  $M$  del segmento  $AB$  ed aventi rispettivamente i segmenti  $AM$  e  $MB$  come diametri.

### 6.7 Costruire l'esagono di cui sono dati

1. il centro ed un vertice
2. due vertici consecutivi
3. due vertici opposti

### 6.8 Costruire il poligono regolare di cui sono dati

1. il centro, un vertice ed il numero di vertici
2. due vertici consecutivi ed il numero dei vertici

### 6.9 Costruire la circonferenza

1. avente un dato segmento come diametro
2. passante per due dati punti ed avente centro su una data retta
3. avente per centro un dato punto e tangente ad una data retta
4. avente per centro un dato punto e tangente ad un'altra data circonferenza
5. inscritta in un triangolo di cui sono dati i vertici
6. circoscritta ad un triangolo di cui sono dati i vertici

### 6.10 Costruire il punto di una circonferenza più vicino ad un dato punto.

- 6.11** Costruire la retta tangente ad una circonferenza in un suo punto.
- 6.12** Costruire la più piccola circonferenza contenente tre dati punti.
- 6.13** Costruire la più piccola circonferenza che contiene due date circonferenze.
- 6.14** Costruire il centro di omotetia di due circonferenze.
- 6.15** Costruire le circonferenze inscritte ad un dato triangolo.
- 6.16** Costruire le circonferenze exinscritte ad un dato triangolo (una circonferenza *exinscritta* ad un triangolo è una circonferenza tangente ad un lato ed ai prolungamenti degli altri due).
- 6.17** Stabilire se due dati segmenti si intersecano.

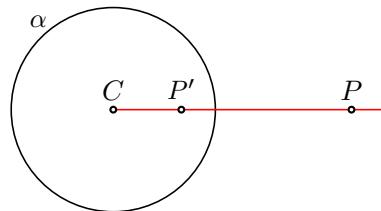
**6.18** Data una circonferenza  $\alpha$  di centro  $C$  e raggio  $r$  ed un punto  $P$  si consideri la semiretta  $\lambda$  uscente da  $C$  e passante per  $P$ . Rimane univocamente identificato il punto  $P' \in \lambda$  tale che

$$\overline{CP} \cdot \overline{CP'} = r^2$$

Il punto  $P'$  così determinato è detto *inverso del punto  $P$  rispetto alla circonferenza  $\alpha$*  e la trasformazione

$$\begin{aligned} inv_{\alpha} : Point &\rightarrow Point \\ P &\mapsto P' \end{aligned}$$

è detta *inversione rispetto alla circonferenza  $\alpha$* .



Dato un punto del piano, costruire il suo inverso. *Suggerimento:* considerare la soluzione del problema 6.5.9.

**6.19** Analizzare le proprietà della trasformazione di inversione di un punto rispetto ad una circonferenza; in particolare:

1. determinare la figura costituita dai punti uniti
2. determinare l'immagine di una retta
3. determinare l'immagine di una circonferenza

**6.20** Usando le funzioni  $\phi$  e  $\psi$  delle costruzioni sulla retta, determinare

1. il doppio di un numero  $x$
2. la distanza fra due generici punti dell'asse reale

3. il simmetrico di un punto rispetto ad un altro dell'asse reale

**6.21** Usando il solo compasso, determinare

1. il punto medio di un segmento dell'asse reale
2. il punto simmetrico di un punto rispetto ad un altro, sull'asse reale
3. la distanza fra due generici punti dell'asse reale
4. la distanza fra due generici punti del piano
5. il punto simmetrico di un punto rispetto ad una retta

# 7

---

## PUNTEGGIARE

---

*Euclide elaborò la sua geometria partendo da un insieme di concetti fondamentali, uno dei quali è il punto. Un punto può definirsi come un'entità che ha una posizione, ma non ha nessun'altra proprietà. Non ha colore, né dimensione, né forma.*

S. Papert, *Mindstorm*

Come avviene spesso in matematica, gli enti trattati si affrancano da un loro iniziale uso esperenziale ed acquistano successivamente un'esistenza concettuale. Nella geometria questo percorso prese avvio considerando l'ente geometrico più elementare immaginabile, il *punto*, e considerando il piano non come *ambiente su cui disegnare* ma come *insieme composto da punti*; per sottolineare questa impostazione si parla solitamente di *piano punteggiato*. Con un ulteriore passo verso un più alto livello di astrazione, si arriva poi a concepire i punti come entità astratte sulle quali eseguire operazioni, in un contesto algebrico, si usano i punti come elementi di base sui quali definire ulteriori concetti e relazioni come allineamento, distanza ed altri. Questo approccio alternativo, incentrato sul concetto di punto, porta a considerare le figure come *insiemi di punti* soddisfacenti a specifiche proprietà; e, naturalmente, le operazioni definite sui punti inducono a loro volta relazioni, proprietà ed operazioni sulle figure.

Il piano punteggiato costituisce un buon punto di partenza per lo studio della geometria. Nonostante l'apparente povertà, in questo ambiente si possono definire alcune operazioni facendo leva sulle quali si possono descrivere interessanti algoritmi; la situazione è simile a quella che si riscontra nell'aritmetica elementare quando si definiscono le operazioni elementari sui numeri naturali e poi, basandosi su queste, si definiscono, direttamente o algoritmicamente, altre operazioni. In sintesi, partendo dal concetto di *punto*, si riesce a tracciare un percorso articolato attraverso varie parti della geometria piana.

## 7.1 I punti e loro denominazione

Per indagare ed analizzare i concetti geometrici primordiali è necessario liberare la nostra mente da tutte le esperienze ed i concetti maturati individualmente nel proprio percorso scolastico ed elaborati globalmente nel corso dell'evoluzione umana; in altri termini, bisogna porsi nella situazione mentale nella quale potrebbe essersi trovato l'uomo vissuto 10 mila anni fa.

Abbandonando momentaneamente la pesante sovrastruttura maturata nel corso del nostro percorso scolastico, affrontiamo la geometria in un ambiente sufficientemente vergine da lasciare spazio alla fantasia ed alla creatività. Cerchiamo di lasciarci guidare dal parallelismo di quanto avviene per i numeri naturali. I numeri naturali hanno una caratterizzazione distintiva che consente di denotarli individualmente mediante dei simboli quali 0, 1, 2, 3, .... Per i punti la situazione è diversa: anche se ontologicamente sono privi di qualsiasi attributo fisico, vengono rappresentati nel piano mediante un puntino e per denotarli individualmente è necessario etichettare i vari punti di interesse del discorso mediante dei nomi  $A$ ,  $B$ ,  $C$ , ....

Pensiamo ad un foglio bianco, illimitatamente grande. Disegniamo su di esso un *puntino* (potremmo chiamarlo *punto* ma, adottando la definizione di Euclide che "un punto è senza dimensioni", il diminutivo è significativo in quanto vogliamo pensarla così piccolo che non ci possa suggerire dei pensieri del tipo: *Che forma ha?*, *Quanto grande è?*, *Di che colore è?*. Per preparare il terreno per l'ingresso in scena di altri punti, denominiamo i punti mediante dei nomi identificativi, al fine di poter riferirsi a più punti distinti. Denominiamo con  $A$  il punto di cui si è parlato appena sopra. Disegniamo sul foglio un altro punto  $B$ . A questo *punto* (come se il nostro discorso si svolgesse lungo una *linea*) ci poniamo una prima domanda: *I due punti  $A$  e  $B$  coincidono?* La risposta a questa domanda richiede che sia specificata la caratterizzazione ontologica dei punti: essi risultano definiti dalla loro *posizione* nel piano e, quindi, due punti sono *uguali* se hanno la stessa posizione ossia, detto in modo più intuitivo, se sono *uno sopra l'altro*.

Consideriamo nel piano tre punti distinti  $A, B, C$  e ci poniamo la seguente ulteriore domanda: *Il punto  $A$  è più vicino (o lontano) ad  $B$  o a  $C$ ?* La risposta non richiede il concetto di unità di misura, di lunghezza o di distanza ma solamente il concetto comparativo di *meno o più lontano*, senza obbligatoriamente passare attraverso il confronto fra le lunghezze dei due segmenti. I concetti di vicino/lontano e l'operazione di confronto di vicinanza/lontananza, benché all'apparenza intuitivi ed elementari, sono profondi e costituiscono il seme per il concetto matematico di *spazio topologico*.

Con riferimento a due dati punti ci poniamo ancora la seguente domanda: *Quanto distano i due punti?* Si tratta di una domanda più avanzata della precedente, che non riusciva a porsi l'uomo di 10 mila anni fa in quanto richiede il concetto di *distanza* o, in alternativa, i concetti di *segmento* e di *misura* di un segmento. Solo in epoche più tarde gli antichi egizi sentirono l'esigenza di misurare numericamente le distanze fra i vari punti che delimitavano le zone delle colture per ricostruire i confini dei terreni coltivati che le esondazioni del Nilo avevano cancellato.

## 7.2 Punti e figure

Nel seguito indicheremo con  $\text{Point}$  l'insieme dei punti del piano e con  $\mathbb{B}$  l'insieme dei valori logici  $\{\text{FALSE}, \text{TRUE}\}$ . Una *relazione n-aria*  $\mathcal{R}$  è un sottoinsieme di  $\text{Point}^n$  costituito dalle  $n$ -ple  $(P_1, P_2, \dots, P_n)$  di punti che soddisfano alla relazione  $\mathcal{R}$ . Una relazione può essere interpretata anche come una funzione  $\mathcal{R} : \text{Point}^n \rightarrow \mathbb{B}$  definita come segue:

$$\mathcal{R} : (P_1, \dots, P_n) \mapsto \begin{cases} \text{TRUE} & \text{se } P_1, \dots, P_n \in \mathcal{R} \\ \text{FALSE} & \text{altrimenti} \end{cases}$$

Una relazione  $\mathcal{R}$  può essere vista anche come una proprietà  $\mathcal{P}$ , considerando equivalenti le due affermazioni "i punti  $P_1, \dots, P_n$  appartengono alla relazione  $\mathcal{R}$ " e "i punti  $P_1, \dots, P_n$  soddisfano alla proprietà  $\mathcal{P}$ ". Questa equivalenza è sottolineata anche dalla scrittura sintetica che assume rispettivamente le forme  $\mathcal{R}(P_1, \dots, P_n)$  e  $\mathcal{P}(P_1, \dots, P_n)$ , anche se, in sostanza,  $\mathcal{R}$  è un sottoinsieme di  $\text{Point}^n$  mentre  $\mathcal{P}$  è una funzione  $\text{Point}^n \rightarrow \mathbb{B}$ .

L'insieme dei punti di una relazione  $\mathcal{R}$  o, equivalentemente, l'insieme dei punti che soddisfano ad una data proprietà  $\mathcal{P}$ , viene detto *figura*.

Il considerare una relazione come insieme o come proprietà orienta verso due diverse linee di sviluppo:

- *logica*: si stabiliscono degli assiomi, coerenti e caratterizzanti una specifica geometria (euclidea, iperbolica, ellittica, ...); usando gli strumenti ed i metodi della logica si derivano teoremi; nell'ambito della programmazione questa impostazione rientra nel *paradigma logico-dichiarativo*
- *algebrica*: si definiscono le operazioni di base sui punti; combinando queste operazioni si costruiscono le espressioni mediante le quali si generano nuovi punti; nell'ambito della programmazione questa impostazione rientra nei *paradigma funzionale-imperativo*

## 7.3 Il sistema geometrico di Tarski

In questo paragrafo viene descritta una porzione interessante di teoria geometrica, seguendo l'impostazione del matematico, logico e filosofo Alfred Tarski (1902-1983), esprimendo definizioni, concetti e proprietà della geometria del piano facendo riferimento al solo concetto di *punto* ed alle relazioni di *interposizione* ed *equidistanza* fra punti. In un articolo del 1959<sup>1</sup> Tarski propose una teoria geometrica basandosi esclusivamente sui *punti*, senza considerare i concetti relativi ad entità geometriche più complesse, quali segmenti, rette, triangoli, circonferenze ed altre. Tarski sviluppa tutta la sua teoria basandosi solo su due concetti: l'*allineamento* di punti e l'*equidistanza* fra punti. La teoria si fonda sulla definizione delle seguenti relazioni fra punti:

1. la relazione ternaria  $\mathcal{F}(A, B, C)$  indica che i tre punti  $A$ ,  $B$ , e  $C$  sono allineati e che  $B$  si trova fra il punto  $A$  ed il punto  $C$  (relazione di *interposizione* o *stare fra*)

<sup>1</sup>Tarski Alfred, *What is elementary Geometry?*

2. la relazione quaternaria  $\mathcal{I}(A, B, C, D)$  indica che i due segmenti  $AB$  e  $CD$  sono isometrici (relazione di *isometria*); dato il significato, questa relazione quaternaria viene spesso indicata, in modo molto più naturale e significativo, con la notazione  $AB \equiv CD$

Si tratta di un sistema assiomatico dove la natura dei punti può essere arbitraria: anche se la teoria trova la sua naturale collocazione considerando i punti dell'usuale piano geometrico, dove trovano un'evidente interpretazione anche le due relazioni  $\mathcal{F}$  e  $\mathcal{I}$ . La potenza di questo sistema geometrico è confermata dal fatto che le due relazioni  $\mathcal{F}$  ed  $\mathcal{I}$  consentono di esprimere tutti i teoremi della geometria elementare.

Il linguaggio usato per descrivere la teoria si basa sulle seguenti convenzioni:

1. tutte le variabili  $A, B, C, \dots$  che vengono utilizzate denotano dei punti del piano
2. vengono utilizzati gli usuali segni di uguaglianza  $=$  e diversità  $\neq$ , il segno di implicazione  $\Rightarrow$ , i quantificatori  $\exists$  (esiste) e  $\forall$  (per ogni) e gli operatori logici  $\vee, \wedge, \neg$  con gli usuali significati

Questo sistema di regole sintattiche genera un concreto ed interessante linguaggio basato sulla logica del primo ordine, ricorrendo al quale si possono esprimere concetti e proprietà che si incontrano nella geometria euclidea. Un aspetto importante dell'impostazione di Tarski consiste nell'essere una teoria completa e decidibile. Di conseguenza per ogni proposizione espressa esiste un algoritmo in grado di dimostrare o confutare la proposizione.

**Problema 7.3.1** Esprimere nel sistema assiomatico di Tarski la condizione che  $M$  è punto medio del segmento  $AB$ .

*Soluzione.* La condizione si esprime come segue:

$$\mathcal{F}(A, M, B) \wedge \mathcal{I}(A, M, M, B)$$

□

## 7.4 Relazioni fra punti

Consideriamo l'insieme dei punti del piano ed analizziamo le principali relazioni che si possono stabilire fra di essi. Le relazioni che vengono descritte a seguire traggono spunto dall'impostazione assiomatica di Tarski e fanno appello a concetti geometrici che vengono assunti intuitivamente, evitando di inoltrarci in un pesante approccio formale su base assiomatica.

### 7.4.1 Uguaglianza fra punti

La caratteristica distintiva dei punti del piano risiede nella loro *posizione*. Basandosi su questa caratteristica viene definita la relazione di *uguaglianza* che indica che due punti sono uguali e ciò, interpretato geometricamente, significa

che i due punti coincidono, ossia che hanno la stessa posizione. Per esprimere il fatto che due generici punti  $A$  e  $B$  sono uguali scriveremo, secondo tradizione,  $A = B$  ( $A \neq B$  se i due punti sono diversi) oppure, in notazione funzionale, con  $equ(A, B)$  ( $\neg equ(A, B)$  se i due punti sono diversi).

La relazione di uguaglianza è una relazione di equivalenza, ossia soddisfa alle seguenti proprietà:

1. *riflessiva*:  $P = P$  per ogni punto  $P$
2. *simmetrica*:  $P = Q \implies Q = P$  per ogni coppia di punti  $P, Q$
3. *transitiva*:  $(P = Q) \wedge (Q = R) \implies (P = R)$  per ogni terna di punti  $P, Q, R$

**Esempio 7.4.1** - La seguente condizione esprime il fatto che i tre punti  $A, B, C$  sono fra loro distinti:

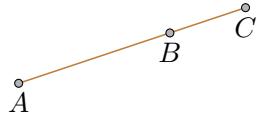
$$(A \neq B) \wedge (B \neq C) \wedge (A \neq C)$$

□

### 7.4.2 Interposizione ed allineamento

Una fondamentale relazione fra punti del piano corrisponde al concetto di *interposizione* tra punti; viene espressa mediante un predicato ternario come segue:

$$fra(A, B, C) \stackrel{\text{def}}{=} \text{il punto } B \text{ è compreso fra il punto } A \text{ ed il punto } C$$



Dato il ruolo simmetrico assunto dai due argomenti  $A$  e  $C$  nella precedente formulazione, si accetta il seguente assunto:

$$fra(A, B, C) \Leftrightarrow fra(C, B, A)$$

Dal predicato *fra* può essere derivata la relazione più debole di *allineamento* fra tre punti come segue:

$$ali(A, B, C) \stackrel{\text{def}}{=} fra(A, B, C) \vee fra(B, A, C) \vee fra(A, C, B)$$

Il predicato *ali* può essere interpretato anche come che uno dei tre punti sta sulla retta individuata dagli altri due, oppure che i tre punti si trovano sulla stessa retta.

**Osservazione.** Le relazioni *fra* ed *ali* presuppongono che il piano, inteso come insieme di punti, abbia una struttura rigida ed indeformabile.

*Esempio 7.4.2 -* La seguente condizione esprime il fatto che i tre punti  $A, B, C$  sono vertici di un triangolo non degenere:

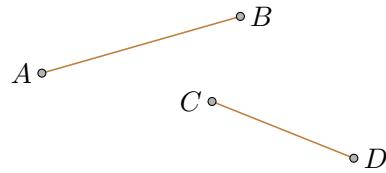
$$\neg \text{ali}(A, B, C)$$

□

### 7.4.3 Isometria

La relazione di *isometria* corrisponde al concetto di uguale distanza fra coppie di punti, ossia uguale lunghezza di segmenti aventi dati punti come estremi; viene descritta mediante il seguente predicato:

$$\text{iso}(A, B, C, D) \stackrel{\text{def}}{=} \text{il segmento } AB \text{ è isometrico al segmento } CD$$



*Esempio 7.4.3 -* Essendo che, per definizione, un parallelogramma è un quadrilatero avente uguali le coppie di lati opposti, i quattro punti  $A, B, C, D$  costituiscono, nell'ordine, i vertici di un parallelogramma se

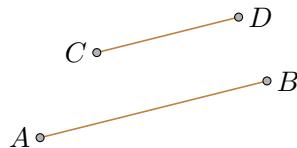
$$\text{iso}(A, B, C, D) \wedge \text{iso}(A, D, B, C)$$

□

### 7.4.4 Parallelismo

Nella geometria tradizionale la relazione di *parallelismo* è una proprietà che viene riferita ai segmenti o alle rette; può comunque essere definita con riferimento ai punti, come segue:

$$\text{par}(A, B, C, D) \stackrel{\text{def}}{=} \text{la retta per } A, B \text{ è parallela alla retta per } C, D$$



*Esempio 7.4.4 -* Un modo alternativo a quanto descritto nell'esempio 7.4.3 per esprimere il fatto che i punti  $A, B, C, D$  costituiscono i vertici di un parallelogramma si basa sul parallelismo dei lati opposti e viene espresso mediante la condizione

$$\text{par}(A, B, C, D) \wedge \text{par}(A, D, B, C)$$

□

## 7.5 Operazioni sui punti

Le relazioni presentate nel precedente paragrafo consentono di definire ed analizzare delle proprietà dei punti e delle figure del piano ma non permettono di generare nuovi punti. Adotteremo ora un'impostazione di tipo operativo-costruttivo, definendo alcune operazioni di base che si possono combinare in espressioni articolate. Vedremo che queste operazioni consentono di derivare le relazioni precedentemente descritte. Questa derivazione è fornita dal fatto che se  $f(x_1, \dots, x_n)$  è un'operazione, denotando con  $y$  il risultato, si può definire il predicato

$$p(x_1, \dots, x_n) \stackrel{\text{def}}{=} (y = f(x_1, \dots, x_n))$$

ricorrendo solamente alla relazione  $=$  di uguaglianza fra punti.

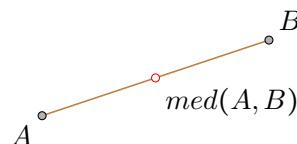
Iniziamo con alcune operazioni *interne* sui punti, ossia operazioni che, operando su punti, generano altri punti come risultato dell'operazione; si tratta dunque di funzioni della forma

$$\mathcal{P}oint^n \rightarrow \mathcal{P}oint$$

La situazione è simile a quella in cui si inizia a costruire l'aritmetica dei numeri naturali a partire dalle operazioni aritmetiche di base (addizione e sottrazione).

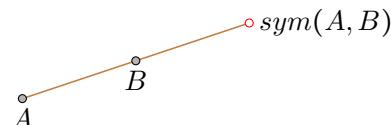
Le prime operazioni che consideriamo sono descritte a seguire; in queste definizioni faremo riferimento ai concetti di segmento, semiretta e retta che assumeremo noti, almeno in modo intuitivo.

1.  $med(A, B)$  : operazione  $\mathcal{P}oint^2 \rightarrow \mathcal{P}oint$  che calcola il punto medio del segmento  $AB$ .

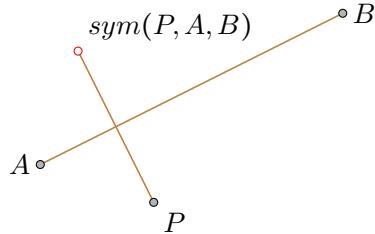


Questa operazione fa leva, in modo latente, sul concetto di equidistanza fra punti, senza richiedere di definire esplicitamente il criterio in base al quale viene calcolata la distanza; in particolare non è richiesta la definizione di un'unità di misura delle lunghezze.

2.  $sym(A, B)$  : operazione  $\mathcal{P}oint^2 \rightarrow \mathcal{P}oint$  che calcola il punto simmetrico di  $A$  rispetto al punto  $B$  (simmetria rispetto ad un punto).



3.  $\text{sym}(P, A, B)$  : operazione  $\text{Point}^3 \rightarrow \text{Point}$  che calcola il punto simmetrico del punto  $P$  rispetto alla retta  $AB$  (simmetria rispetto ad un asse).



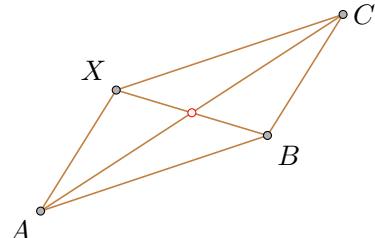
*Osservazione.* In base ad elementari considerazioni geometriche si può dedurre che le funzioni  $\text{med}$  e  $\text{sym}$  sono fra loro legate da una sorta di reciprocità descrittiva evidenziata dalle seguenti proprietà invarianti, valide per ogni coppia di punti  $A, B$ :

$$B \equiv \text{med}(A, \text{sym}(A, B))$$

$$B \equiv \text{sym}(A, \text{med}(A, B))$$

**Problema 7.5.1** Determinare il punto che completa il parallelogramma di vertici  $A, B, C$ .

*Soluzione.* Dall'analisi della seguente figura:



si ricava facilmente che il punto  $X$  che completa il parallelogramma  $A, B, C$  è il simmetrico di  $B$  rispetto al punto medio del segmento  $AC$ , ossia

$$X = f(A, B, C) \stackrel{\text{def}}{=} \text{sym}(B, \text{med}(A, C))$$

□

### 7.5.1 Allineamento, parallelismo e perpendicolarità

Le operazioni  $\text{med}$  e  $\text{sym}$  sopra definite costituiscono un sistema interessante e produttivo all'interno del quale si possono definire operazioni e proprietà geometriche (allineamento di punti, parallelismo e perpendicolarità di rette) ed eseguire calcolazioni, come confermano le seguenti considerazioni.

In base ad elementari considerazioni geometriche, la *proiezione* del punto  $P$  sulla retta passante per i punti  $A, B$  è data dalla seguente espressione:

$$\text{pro}(P, A, B) \stackrel{\text{def}}{=} \text{med}(P, \text{sym}(P, A, B))$$

La funzione *pro* permette di scrivere la condizione che esprime la relazione di *perpendicolarità* fra rette; precisamente, la retta per  $AB$  e la retta  $CD$  sono perpendicolari se la proiezione di  $C$  su  $AB$  coincide con quella di  $D$  su  $AB$ , ossia

$$per(A, B, C, D) \equiv (pro(C, A, B) = pro(D, A, B))$$

La condizione di *allineamento* di tre punti  $A, B, C$  equivale ad affermare che il punto  $C$  coincide con il proprio simmetrico rispetto alla retta  $AB$ ; pertanto

$$ali(A, B, C) \equiv (C = sym(C, A, B))$$

La condizione di *parallelismo* fra rette può essere dedotta dal predicato *ali* di allineamento fra punti; infatti, la condizione di parallelismo fra la retta  $AB$  e la retta  $CD$  equivale ad affermare che il punto  $X = f(A, B, C)$  risulta allineato con i punti  $C, D$ , ossia

$$par(A, B, C, D) \equiv ali(C, D, sym(B, med(A, C)))$$

### 7.5.2 Distanza fra punti

Il problema di determinare quanto distano due punti trova adeguata soluzione nel piano punteggiato dove è definita la nozione di *distanza euclidea*<sup>2</sup> fra due generici punti, definita come la lunghezza del segmento avente per estremi i due dati punti; per fare ciò è sufficiente definire un'unità di misura. È naturale ammettere che il concetto di distanza sia simmetrico nel senso che  $A$  è distante da  $B$  quanto  $B$  è distante da  $A$ ; sarà questo uno dei prerequisiti degli spazi metrici nei quali il concetto di distanza viene definito assiomaticamente.

Un'importantissima operazione che apre la strada a molte altre operazioni, definizioni ed argomenti è fornita dall'operazione di *distanza fra due punti*. Si tratta di un'operazione esterna in quanto il risultato non è un punto, bensì un numero. Proprio questa caratteristica fa sì che questa operazione getti un ponte fra il mondo dei numeri e quello dei punti e, più in generale, delle figure, legando l'aritmetica alla geometria.

Indicando con  $\mathcal{P}oint$  l'insieme dei punti del piano, la funzione di distanza fra due punti è caratterizzata dal seguente prototipo<sup>3</sup>:

$$dist : \mathcal{P}oint^2 \rightarrow \mathbb{R}$$

Per riferirsi al piano così metrizzato si usa il termine *piano euclideo*.

In un contesto metrico, ossia basandosi sulla sola funzione *dist*, si inquadrano molte problematiche geometriche che si possono risolvere mediante il ricorso al concetto di distanza fra punti.

<sup>2</sup>Oltre alla distanza euclidea si possono definire altri criteri di distanza, come sarà visto nel capitolo relativo agli spazi metrici.

<sup>3</sup>Precisamente, la funzione *dist* viene solitamente detta *metrica* mentre il termine *distanza* viene riservato al valore numerico che si ottiene applicando la funzione ad una coppia di punti.

**Problema 7.5.2** Usando la funzione *dist*, stabilire se tre dati punti sono allineati.

**Soluzione.** Il problema è equivalente a stabilire se il triangolo individuato dai tre punti è degenero. Ciò può essere stabilito basandosi sulle lunghezze dei tre lati (la lunghezza del lato più lungo uguale alla somma delle lunghezze degli altri due).

---

**Algoritmo 1** - Punti allineati - *ali(A, B, C)*

---

**Input:** punti  $A, B, C$

**Output:** TRUE se e solo se i tre punti  $A, B, C$  sono allineati

```

1: $a \leftarrow dist(B, C)$
2: $b \leftarrow dist(A, C)$
3: $c \leftarrow dist(A, B)$
4: $m \leftarrow max(a, b, c)$
5: $n \leftarrow a + b + c - m$
6: $r \leftarrow m = n$
7: return r
```

---

□

### 7.5.3 Angoli

Oltre alla distanza, un altro concetto nodale riguardante la geometria del piano è costituito dalla nozione di *angolo*. Un angolo è definito come ciascuna delle due parti di piano comprese da due semirette aventi la stessa origine.

Nell'ambiente del piano punteggiato un angolo risulta individuato da una terna di punti  $A, V, B$  in cui  $V$  è il vertice ed  $A$  e  $B$  sono due punti che individuano le due semirette di vertice  $V$  che delimitano l'angolo, con la convenzione che tale terna rappresenti l'angolo ottenuto dalla rotazione attorno al vertice  $V$ , in senso antiorario, della semiretta  $VA$  sulla semiretta  $VB$  (figura 7.1). Denoteremo tale angolo con  $\widehat{AVB}$ .

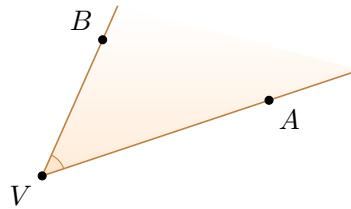


Figura 7.1: Angolo  $\widehat{AVB}$ .

Dalla precedente definizione di angolo risulta naturale definire le relazioni di uguaglianza e di minore e maggiore fra coppie di angoli mediante sovrapposizione di due semirette dei due angoli; ciò risulta però poco utile dal punto di vista operativo ed algoritmico. Risulta più efficace definire il concetto di *misura di un angolo* mediante la quale derivare sia le relazioni di comparazione

che molti altri procedimenti che si fondano su calcoli numerici. Nell'evoluzione del pensiero matematico arrivare a definire il concetto di misura di un angolo ha richiesto uno sforzo intellettuale maggiore di quello richiesto per la misura di lunghezze o, equivalentemente, di distanza fra due punti. Una tradizionale modalità, utilizzata dai babilonesi ancora tre millenni fa, consiste nel suddividere l'angolo giro in 360 parti uguali e considerare ciascuna di queste parti come unità di misura (grado sessagesimale). Denoteremo con

$$\text{ang}(A, V, B)$$

la misura dell'angolo  $A\widehat{V}B$  individuato dai tre punti  $A, V, B$ . Questa funzione è caratterizzata dal prototipo

$$\text{dist} : \text{Point}^3 \rightarrow \mathbb{R}$$

**Problema 7.5.3** Dati tre punti  $A, V, B$ , stabilire se l'angolo  $A\widehat{V}B$  è acuto.

*Soluzione.* Consideriamo la semicirconferenza di diametro  $AB$ .

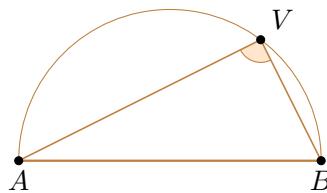


Figura 7.2: Decisione se l'angolo in  $V$  è acuto.

Se il punto  $V$  sta sulla semicirconferenza, allora il triangolo  $\triangle AVB$  è rettangolo e, per il teorema di Pitagora, vale la condizione

$$\overline{AB}^2 = \overline{AV}^2 + \overline{BV}^2$$

La situazione di angolo acuto corrisponde al caso cui il punto  $V$  si trova fuori dalla semicirconferenza; e questo caso corrisponde alla condizione

$$\overline{AB}^2 < \overline{AV}^2 + \overline{BV}^2$$

Tutte le precedenti considerazioni sono riassunte nell'algoritmo 2. È interessante notare come una questione relativa agli angoli venga risolta mediante il ricorso alla funzione *dist* con il tramite di una proprietà (i triangoli inscritti in una semicirconferenza sono rettangoli) ed un teorema (di Pitagora).

**Algoritmo 2** - Test se un angolo è acuto**Input:** punti  $A, V, B$  che individuano l'angolo  $\widehat{V}$ **Output:** TRUE se e solo se l'angolo  $\widehat{V}$  è acuto

- 1:  $a \leftarrow dist(B, C)$
- 2:  $b \leftarrow dist(A, C)$
- 3:  $c \leftarrow dist(A, B)$
- 4:  $q \leftarrow c^2 < a^2 + b^2$
- 5: **return**  $q$

□

**7.5.4 Omotetie**

Dati due punti  $A, B$  ed un numero reale  $\lambda$  si definisce *punto omotetico* di  $B$  rispetto ad  $A$  di rapporto  $\lambda$  il punto  $B'$  situato sulla retta passante per  $A$  e  $B$  tale che sia  $\lambda$  il rapporto delle lunghezze dei due segmenti  $AB'$  ed  $AB$ ; tale punto viene indicato con la notazione funzionale

$$hom(A, B, \lambda)$$

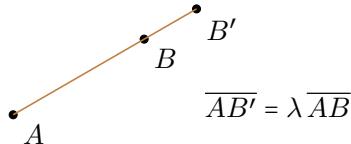


Figura 7.3: Omotetia di rapporto  $\lambda$  del punto  $B$  rispetto al punto  $A$ .

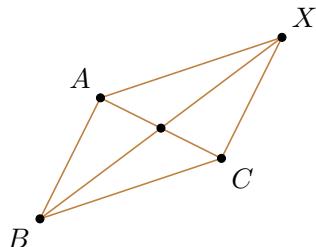
Al variare del parametro  $\lambda$  in  $\mathbb{R}$ , l'espressione  $hom(A, B, \lambda)$  fornisce un generico punto della retta passante per  $A$  e  $B$ ; per  $\lambda = 0$  si ottiene il punto  $A$ ; per  $\lambda = 1$  si ottiene il punto  $B$ ; per  $\lambda > 1$  si ottengono i punti della semiretta  $\overrightarrow{AB}$  oltre il punto  $B$ ; per  $\lambda < 0$  si ottengono punti sul prolungamento della semiretta di vertice  $A$  e dalla parte opposta a  $B$ ; per  $0 < \lambda < 1$  si ottengono tutti i punti compresi fra  $A$  e  $B$ . In generale,  $\lambda$  rappresenta il rapporto di omotetia del punto  $B'$  rispetto al centro  $A$ , assumendo che il segmento  $AB$  sia di lunghezza unitaria.

L'operazione  $hom$  presenta i seguenti frequenti casi particolari:

$$\begin{aligned} hom(A, B, 0) &= A \\ hom(A, B, 1) &= B \\ hom(A, B, \frac{1}{2}) &= med(A, B) \\ hom(A, B, 2) &= sym(A, B) \\ hom(A, B, -1) &= sym(B, A) \end{aligned}$$

**Problema 7.5.4** Ricorrendo alla funzione  $hom$ , dati tre punti  $A, B, C$ , determinare il punto  $X$  che tale che  $ABCX$  risulti un parallelogramma.

*Soluzione.* Il problema è descritto nella figura che segue dalla quale si ricava immediatamente che il punto  $X$  è il simmetrico di  $B$  rispetto al punto medio fra  $A$  e  $C$ .



Quindi

$$X \stackrel{\text{def}}{=} \hom(B, \hom(A, C, \frac{1}{2}), 2)$$

Questa espressione corrisponde a quella già trovata nel problema 7.5.1 espri-  
mando *sym* e *med* in funzione di *hom*.  $\square$

**Problema 7.5.5** Determinare la bisettrice di un angolo.

*Soluzione.* Il problema, nella sua formulazione essenziale corrispondente alla consegna, può essere descritto graficamente dalla figura 7.4.

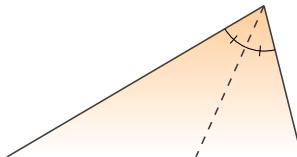


Figura 7.4: Problema della determinazione della bisettrice di un angolo.

Il problema può essere ricondotto al contesto del piano punteggiato, individuando i dati (angolo da bisecare) mediante 3 punti ed il risultato (bisettrice dell'angolo) mediante un punto di passaggio (diverso dal vertice). Il problema così riformulato è descritto nella figura 7.5.

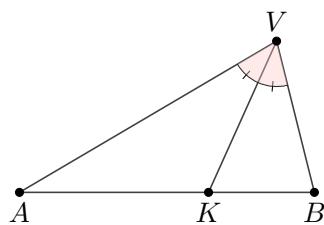


Figura 7.5: Determinazione della bisettrice  $AK$  dell'angolo  $AVB$ .

Facciamo ora un'incursione nella geometria euclidea per dimostrare che il punto  $K$  divide il lato  $AB$  in due segmenti di lunghezze proporzionali agli altri due lati, ossia

$$\overline{AK} : \overline{KB} = \overline{AV} : \overline{BV}$$

La dimostrazione si basa sulla figura 7.6 dove è tracciato il segmento  $BD$  parallelo alla bisettrice  $VK$  avente il vertice  $D$  sul prolungamento del lato  $AV$ . In base al teorema di Talete vale la proporzionalità  $\overline{AK} : \overline{AV} = \overline{KB} : \overline{VD}$ . Inoltre, i quattro angoli evidenziati in figura sono isometrici in quanto generati dalle due rette parallele  $KV$  e  $BD$  tagliate dalla trasversale  $BV$ ; di conseguenza  $\overline{BV} = \overline{DV}$ .

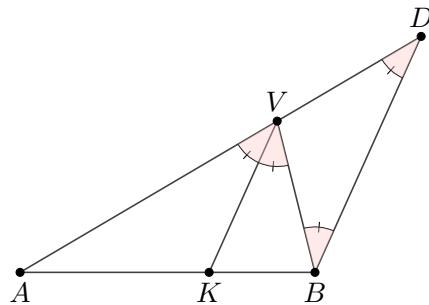


Figura 7.6: Dimostrazione della proprietà  $\overline{AK} : \overline{KB} = \overline{AV} : \overline{BV}$ .

Ritornando alla figura 7.5, applicando la proprietà del comporre alla proporzione  $\overline{AK} : \overline{KB} = \overline{AV} : \overline{BV}$  appena dimostrata, si ottiene

$$(\overline{AK} + \overline{KB}) : \overline{AK} = (\overline{AV} + \overline{BV}) : \overline{AV}$$

da cui

$$\overline{AB} : \overline{AK} = (\overline{AV} + \overline{BV}) : \overline{AV}$$

Da questa nuova proporzione si può calcolare il rapporto  $k$  fra le lunghezze dei segmenti  $AK$  e  $AB$ :

$$k = \frac{\overline{AK}}{\overline{AB}} = \frac{\overline{AV}}{\overline{AV} + \overline{BV}}$$

da cui

$$K = hom(A, B, k)$$

Il procedimento completo è descritto nell'algoritmo 3.

**Algoritmo 3** - Bisettrice di un angolo

**Input:** punti  $A, V, B$  che individuano l'angolo di vertice  $V$

**Output:** punto  $K$  che individua la bisettrice dell'angolo di vertice  $V$

- 1:  $\alpha \leftarrow \overline{AV}/(\overline{AV} + \overline{BV})$
- 2:  $K \leftarrow hom(A, B, k)$
- 3: **return**  $K$ .

□

**7.5.5 Rotazioni**

Dati due punti  $A, B$  e la misura  $\alpha$  di un angolo si definisce *punto ruotato* di  $B$  rispetto ad  $A$  di angolo  $\alpha$  il punto  $B'$  corrispondente di  $B$  che si ottiene ruotando il segmento  $AB$  di un angolo  $\alpha$  in senso antiorario attorno al vertice  $A$ ; tale punto viene indicato con la notazione funzionale

$$rot(A, B, \alpha)$$

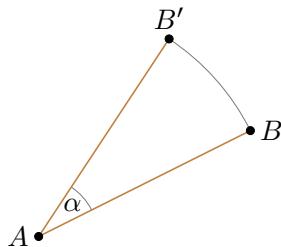


Figura 7.7: Rotazione di angolo  $\alpha$  del punto  $B$  rispetto al punto  $A$ .

**Osservazione.** La disponibilità delle funzioni di base (sui punti e sulle altre entità geometriche) equivale all'uso di particolari strumenti di misura e di costruzione geometrica; ad esempio la funzione *dist* equivale all'uso di una riga graduata; la funzione *hom* equivale all'uso di una riga di tracciamento e di una riga graduata; la funzione *rot* equivale all'uso di un compasso e di un goniometro.

L'uso combinato delle funzioni *hom* e *rot* permette di descrivere interessanti algoritmi geometrici, come confermano gli esempi ed i problemi che seguono.

**Esempio 7.5.1** - Ricorrendo alle funzioni *hom* e *rot* si può scrivere un'espressione che rappresenta il baricentro del quadrato di lato  $AB$  situato alla sinistra del segmento (orientato)  $AB$ . L'espressione è la seguente:

$$hom(A, rot(B, A, 90), \frac{1}{2})$$

□

## 7.6 L'algebra dei punti

Un interessante arricchimento del piano punteggiato consiste nel definire un punto di riferimento detto *origine* e denotato generalmente con  $O$ . In questa situazione il piano viene detto *piano vettoriale*.

La definizione del punto origine permette di definire delle ulteriori operazioni sui punti, come descritto a seguire.

- *addizione* fra due punti  $A$  e  $B$ : è il punto opposto all'origine che si ottiene costruendo il parallelogramma di lati  $OA$  ed  $OB$ , come descritto nella figura 7.8.

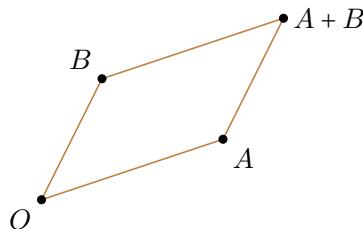


Figura 7.8: Addizione fra il punto  $A$  ed il punto  $B$ .

- *inverso* di un punto  $A$ : è il punto che addizionato a  $A$  da come risultato il punto origine  $O$ ; si tratta del punto simmetrico di  $A$  rispetto ad  $O$ . L'inverso del punto  $A$  viene denotato con  $-A$ . La figura 7.9 illustra la costruzione geometrica del punto  $-A$ .

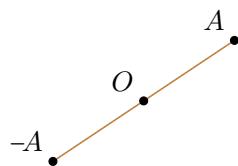


Figura 7.9: Inverso  $-A$  del punto  $A$ .

- *sottrazione* fra il punto  $A$  ed il punto  $B$ : è il punto definito come segue:

$$A - B \stackrel{\text{def}}{=} A + (-B)$$

La figura 7.10 illustra la costruzione geometrica del punto  $A - B$ .

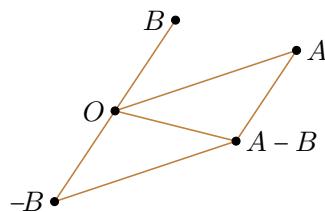


Figura 7.10: Sottrazione fra il punto  $A$  ed il punto  $B$ .

Da questa definizione conseguе che per ogni punto  $A$  vale la proprietà

$$A - A = O$$

- *moltiplicazione* di un punto  $A$  per un numero  $\lambda \in \mathbb{R}$ : è il punto della retta passante per  $O$  ed  $A$ , tale che sia  $|\lambda|$  il rapporto delle lunghezze dei segmenti  $OA'$  ed  $OA$ ; il punto si trova sulla semiretta  $OA$  se  $\lambda > 0$ ; altrimenti sulla semiretta opposta; in particolare il punto si trova sul segmento  $OA$  se  $0 < \lambda < 1$ ; il punto coincide con  $O$  se  $\lambda = 0$  e con  $A$  se  $\lambda = 1$ .

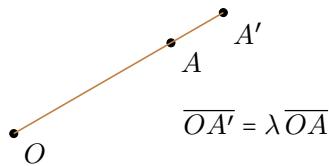


Figura 7.11: Moltiplicazione del punto  $A$  per il numero  $\lambda = 1.3$ .

Al posto di  $\lambda * A$  viene spesso utilizzata la notazione  $\lambda A$ . L'applicazione  $A \mapsto \lambda A$  di  $\mathcal{P}oint$  in sé stesso è detta *omotetia di rapporto*  $\lambda$ .

- *rotazione* del punto  $A$  di un angolo  $\alpha$ : è il punto che si ottiene ruotando il segmento  $OA$  di un angolo  $\alpha$  attorno all'origine  $O$ ; viene indicato con

$$A < \alpha$$

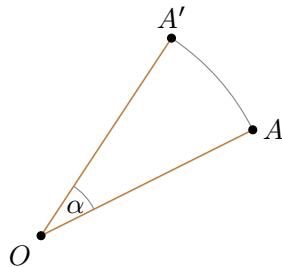


Figura 7.12: Rotazione del punto  $A$  di un angolo  $\alpha$ .

Le operazioni sui punti descritte nel precedente paragrafo godono di molte proprietà che sono riassunte nel seguente teorema.

**TEOREMA 3.** L'insieme  $\mathcal{P}oint$  dei punti del piano, con le operazioni sopra definite, soddisfa alle seguenti proprietà:

- $G_1$ .  $P + Q = Q + P$  per ogni  $P, Q \in \mathcal{P}oint$  [commutatività]
- $G_2$ .  $(P + Q) + R = P + (Q + R)$  per ogni  $P, Q, R \in \mathcal{P}oint$  [associatività]
- $G_3$ .  $P + O = O + P = P$  per ogni  $P \in \mathcal{P}oint$  [elemento neutro]

$G_4$ . per ogni  $P \in \text{Point}$  esiste  $-P \in \text{Point}$  tale che  $P + (-P) = O$  [inverso]

$G_5$ .  $\lambda(P + Q) = \lambda P + \lambda Q$  per ogni  $P, Q \in \text{Point}$  e  $\lambda \in \mathbb{R}$

$G_6$ .  $(\lambda + \mu)P = \lambda P + \mu P$  per ogni  $P \in \text{Point}$  e  $\lambda, \mu \in \mathbb{R}$

$G_7$ .  $(\lambda \mu)P = \lambda(\mu P)$  per ogni  $P \in \text{Point}$  e  $\lambda, \mu \in \mathbb{R}$

$G_8$ .  $1P = P$  per ogni  $P \in \text{Point}$

Le proprietà  $G_1, G_2, G_3, G_4$  possono essere sintetizzate affermando che la struttura  $(\text{Point}, +)$  costituisce un *gruppo abeliano* avente l'origine  $O$  come elemento neutro. Queste quattro proprietà costituiscono infatti gli assiomi dei gruppi abeliani. Le proprietà  $G_5, G_6, G_7, G_8$  (assieme a molte altre) discendono come conseguenza dalle prime quattro; la loro dimostrazione può essere fatta nel contesto astratto della teoria dei gruppi.

L'insieme di proprietà riportate nel Teorema 3 corrisponde alle usuali proprietà dell'algebra; questo permette di scrivere e di elaborare espressioni coinvolgenti i punti operando formalmente, senza preoccuparsi del significato delle trasformazioni, ottenendo, alla fine, dei risultati semanticamente corretti.

**Esempio 7.6.1** - Indicando con  $P, Q$  dei generici punti, si possono eseguire i seguenti calcoli sulle espressioni:

$$\begin{aligned} 2P + 3P &\rightarrow 5P \\ P + Q + P &\rightarrow 2P + Q \\ 3P - 2Q + P &\rightarrow 2(P - Q) \end{aligned}$$

□

**Problema 7.6.1** Determinare il punto medio  $M$  fra due dati punti  $A$  e  $B$ .

**Soluzione.** Dalle definizioni delle operazioni sui punti riportate sopra e dall'osservazione che il punto di intersezione delle diagonali del parallelogramma  $OA(A+B)B$  dimezza le diagonali, dalla figura 7.13 si riconosce facilmente che il punto medio cercato è dato dall'espressione

$$M = \frac{1}{2}(A + B)$$

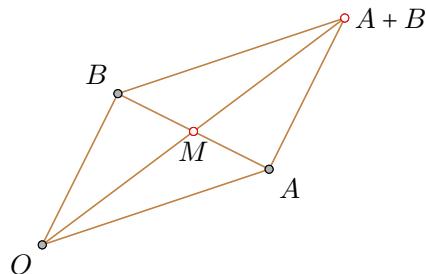
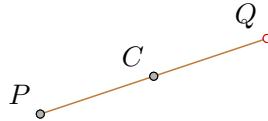


Figura 7.13: Calcolo del punto medio  $M$  fra  $A$  e  $B$ .

□

**Problema 7.6.2** Determinare il punto  $Q$  simmetrico del punto  $P$  rispetto al centro  $C$ .

*Soluzione.* Orientandosi sulla figura



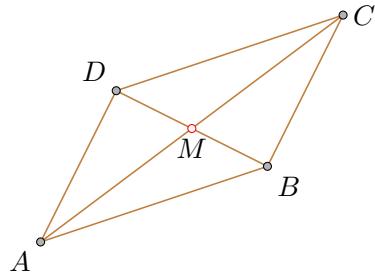
si può calcolare algebricamente il risultato  $Q$  come segue:

$$\begin{aligned}\frac{1}{2}(P+Q) &= C \\ P+Q &= 2C \\ Q &= 2C - P\end{aligned}$$

□

**Problema 7.6.3** Dimostrare che la sequenza di punti  $(A, B, C, D)$  costituisce i vertici di un parallelogramma se e solo se  $A + C = B + D$ .

*Soluzione.* In un parallelogramma le due diagonali si tagliano mutuamente a metà.



Il punto  $M$  d'incrocio è il punto medio delle due diagonali ed è dato da

$$M = \frac{A+C}{2} = \frac{B+D}{2}$$

Da ciò si deduce che  $A+C=B+D$ . □

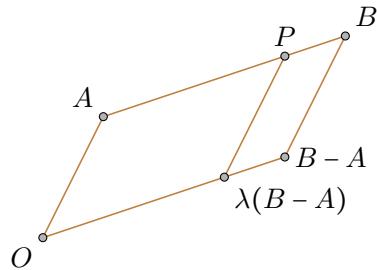
## 7.7 Proprietà delle operazioni sui punti

Le seguenti due proprietà mettono in relazione le operazioni *hom* e *rot* con le operazioni addizione, sottrazione e rotazione fra punti.

**PROPRIETÀ 2.** Siano  $A, B$  due punti distinti del piano. Vale la relazione

$$hom(A, B, \lambda) = A + \lambda(B - A)$$

*Dimostrazione.* Consideriamo la seguente figura:



Il punto  $\lambda(B - A)$  si trova sulla retta per  $O$  e  $B - A$ . In base alla regola del parallelogramma applicata all'addizione fra i due punti  $A$  e  $\lambda(B - A)$  si deduce che il punto  $P$  deve stare sulla retta per  $A$  e  $B$  in quanto i quattro punti  $O, \lambda(B - A), P$  e  $A$  sono i vertici di un parallelogramma.  $\square$

*Osservazione.* Al variare del parametro reale  $\lambda$ , l'espressione  $A + \lambda(B - A)$  genera tutti i punti appartenenti alla retta passante per  $A$  e  $B$ .

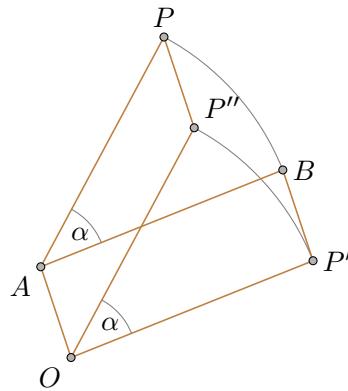
*Esempio 7.7.1 -* Una semplice applicazione di quanto sopra descritto permette di determinare il punto medio  $M$  fra due dati punti  $A, B$ : prendendo  $\lambda = \frac{1}{2}$  si ha:

$$M = A + \frac{1}{2}(B - A) = \frac{1}{2}A + \frac{1}{2}B = \frac{1}{2}(A + B)$$

**PROPRIETÀ 3.** Siano  $A, B$  due punti distinti del piano. Vale la relazione

$$\text{rot}(A, B, \alpha) = A + ((B - A) \triangleleft \alpha)$$

*Dimostrazione.* L'espressione  $\text{rot}(A, B, \alpha)$  rappresenta il punto  $P$  ottenuto dalla rotazione di un angolo  $\alpha$  del segmento  $AB$  attorno al punto  $A$ .



Consideriamo le due rette per  $O$  e parallele rispettivamente a  $AB$  ed  $AP$  e consideriamo i due parallelogrammi  $OP'BA$  e  $OP''PA$ . Per costruzione risulta  $P' = B - A$ . I due angoli  $P'OP''$  e  $BAP$  sono isometrici in quanto alterni generati dall'intersezione di rette parallele di due fasci. Dunque  $P'' = P' \triangleleft \alpha = (B - A) \triangleleft \alpha$ . Osservando che  $P = A + P''$ , si conclude.  $\square$

*Osservazione.* Fissato il punto origine  $O$  e due punti distinti  $A, B$  (diversi dall'origine), l'espressione

$$\lambda A + \mu B \quad \lambda, \mu \in \mathbb{R}$$

detta *combinazione lineare* fra  $A$  e  $B$ , genera, al variare di  $\lambda, \mu$  in  $\mathbb{R}$  tutti i punti del piano. Rimane così stabilita una corrispondenza biunivoca fra i punti del piano e le coppie  $(\lambda, \mu) \in \mathbb{R}^2$ . I valori di tale coppia vengono detti *coordinate* del punto generato. Imponendo delle condizioni alla variabilità della coppia  $(\lambda, \mu)$  rimane individuato un sottoinsieme dei punti del piano. In particolare imponendo la condizione

$$\lambda + \mu = 1$$

si descrivono tutti i punti della retta passante per  $A$  e  $B$ . Di conseguenza l'espressione

$$\lambda A + (1 - \lambda)B$$

descrive tutti i punti della retta passante per  $A$  e  $B$ .

## 7.8 Figure

Nel seguito chiameremo *figura* un insieme ben definito di punti. Ben definito significa che disponiamo di un criterio effettivo, algoritmico, per stabilire in modo univoco se un dato punto appartiene alla figura. Definito così il concetto di figura, si impongono nuove questioni:

- $Q_1$ : come definire (i punti che costituiscono) una figura  $\mathcal{F}$
- $Q_2$ : dato un punto  $P$  ed una figura  $\mathcal{F}$ , stabilire se  $P \in \mathcal{F}$
- $Q_3$ : generare i punti  $P$  appartenenti ad una figura  $\mathcal{F}$
- $Q_4$ : denotare delle specifiche istanze di figure

Queste questioni vengono indagate nei seguenti sottoparagrafi.

### 7.8.1 Definizione di figure

Una figura può essere definita in varie modalità:

- *intuitivamente*: in modo informale, una figura può essere definita in modo discorsivo, appellandosi a concetti già acquisiti, a definizioni già formulate o ad esperienze maturate. Ad esempio, con un atto di astrazione possiamo considerare una coppia non ordinata  $(A, B)$  di punti come un'entità che chiamiamo *segmento*, che denoteremo con  $AB$  e che possiamo pensare come l'insieme dei punti "compresi" fra  $A$  e  $B$ ; ovviamente questa non è una definizione in quanto l'attributo "compresi" dovrebbe essere adeguatamente definito, basandosi su termini già definiti. Sempre in modo informale possiamo definire come *retta* passante per due generici punti  $A$  e  $B$  come l'insieme dei punti "allineati" con  $A$  e  $B$  e una *circonferenza* come l'insieme dei punti "equidistanti" da un punto centrale. Similmente a come sopra si è giunti al concetto di segmento, possiamo considerare 3 punti come un'unica entità, pervenendo al nuovo concetto di *triangolo*. E qui si apre

un mondo di possibilità: di nuovi concetti, di nuove definizioni: bordo, interno-esterno, misura di area, angolo, ....

- *per proprietà*: una classica impostazione consiste nel definire una figura come *luogo geometrico*, ossia come insieme di punti soddisfacenti ad una data proprietà; per comodità si prende spesso una proprietà facilmente indagabile ed in particolare ci si basa sul concetto di *distanza fra due punti* o *allineamento fra due punti*. Se  $p$  è un predicato avente per argomenti punti e numeri, allora  $p$  individua una figura  $\mathcal{F}$  costituita dai punti (anche infiniti) soddisfacenti al predicato  $p$ , ossia

$$\mathcal{F} = \{P \in Point : p(P)\}$$

Notiamo che, se  $p_1$  e  $p_2$  sono due predicati che individuano due figure  $\mathcal{F}_1$  e  $\mathcal{F}_2$ , allora  $p_1 \wedge p_2$  è il predicato che individua i punti di *intersezione* fra le due figure. Analogamente, il predicato  $p_1 \vee p_2$  individua l'*unione* fra le due figure.

**Esempio 7.8.1** - Basandosi sulla funzione *dist* che valuta la distanza fra due punti, si possono definire le seguenti figure:

- *segmento* di estremi  $A$  e  $B$ : insieme dei punti  $P$  tali che

$$dist(P, A) + dist(P, B) = dist(A, B)$$

- *retta* passante per i punti  $A$  e  $B$ : insieme dei punti del segmento  $AB$  uniti all'insieme dei punti  $P$  tali che

$$|dist(P, A) - dist(P, B)| = dist(A, B)$$

- *circonferenza* di centro  $C$  e passante per il punto  $A$ : insieme dei punti  $P$  tali che:

$$dist(P, C) = dist(A, C)$$

Queste definizioni suggeriscono interessanti generalizzazioni verso spazi di dimensioni superiori e verso livelli di astrazione più avanzati (*spazi metrici*), come si vedrà nel prossimo capitolo. □

- *mediante espressione*: basandosi sui punti e sulle loro operazioni si possono definire delle figure geometriche mediante un'espressione che, in funzione di alcuni punti base e di uno o più parametri numerici, fornisce, al variare dei parametri, tutti i punti della figura. Questo metodo è utile anche per generare i punti di una figura.

**Esempio 7.8.2** - Ricorrendo alle operazioni *hom*, *rot* è possibile definire le entità geometriche di retta, segmento e circonferenza come luogo geometrico di punti soddisfacenti ad una proprietà definita mediante queste due operazioni:

- *retta* passante per due punti  $A$  e  $B$ : luogo geometrico dei punti  $P$  che si ottengono, al variare di  $\lambda$  in  $\mathbb{R}$ , mediante l'espressione  $hom(A, B, \lambda)$
- *segmento* di estremi i due punti  $A$  e  $B$ : luogo geometrico dei punti  $P$  che si ottengono, al variare di  $\lambda$  in  $[0, 1]$ , mediante l'espressione  $hom(A, B, \lambda)$
- *circonferenza* di centro  $A$  e passante per  $B$ : luogo geometrico dei punti  $P$  che si ottengono, al variare di  $\alpha$  nell'intervallo  $[0, 2\pi]$ , mediante l'espressione  $rot(A, B, \alpha)$

□

*Osservazione.* Fissati due generici punti distinti  $A$  e  $B$  l'espressione

$$rot(A, hom(A, B, \lambda), \alpha)$$

al variare della coppia  $(\alpha, \lambda)$  in  $\mathbb{R}^2$ , descrive tutto il piano; fissando il valore di  $\alpha$  si ottiene il luogo geometrico dei punti appartenenti alla retta passante per i punti  $A$  e  $B$ ; fissando il valore di  $\lambda$  si ottiene il luogo geometrico costituito dai punti della circonferenza di centro  $A$  e raggio  $dist(A, B)$ .

### 7.8.2 Denotazione di figure

Una figura può essere definita mediante un'espressione avente per argomenti punti e numeri. Per descrivere questi argomenti viene solitamente adottata la notazione funzionale. Vengono descritte a seguire le seguenti figure geometriche più ricorrenti:

- $Segment(A, B)$ : segmento avente per vertici i punti  $A$  e  $B$
- $Line(A, B)$ : retta passante per i punti  $A$  e  $B$
- $Ray(A, B)$ : semiretta avente vertice il punto  $A$  e passante per il punto  $B$
- $Circle(C, P)$ : circonferenza di centro  $C$  e passante per il punto  $P$
- $Circle(C, r)$ : circonferenza di centro  $C$  e raggio  $r$
- $Circle(C, A, B)$ : circonferenza di centro  $C$  e raggio  $\overline{AB}$

Fissata una retta di riferimento per la misura degli angoli, le figure lineari (segmenti, rette, raggi) possono essere definite anche in base ad un argomento che misura l'angolo di inclinazione rispetto a questa retta di riferimento, come descritto a seguire:

- $Segment(A, l, \alpha)$ : segmento avente un vertice sul punto  $A$ , una lunghezza  $l$  ed avente angolo  $\alpha$  di inclinazione rispetto alla retta di riferimento
- $Line(A, \alpha)$ : retta passante per il punto  $A$  ed avente angolo  $\alpha$  di inclinazione rispetto alla retta di riferimento
- $Ray(A, \alpha)$ : semiretta avente vertice il punto  $A$  ed avente angolo  $\alpha$  con una retta prefissata di inclinazione rispetto alla retta di riferimento

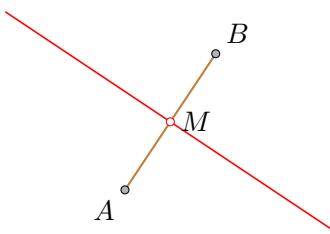
I punti che compaiono come argomenti nei costruttori sopra descritti vengono detti *punti base*.

### 7.8.3 Espressione di figure

Ricorrendo ai costruttori definiti sopra ed alle operazioni sui punti si possono scrivere espressioni che descrivono specifiche figure geometriche.

**Problema 7.8.1** Determinare l'asse del segmento di vertici  $A, B$ .

*Soluzione.* L'asse può essere costruito usando il punto di appoggio costituito dal punto medio del segmento  $AB$ .



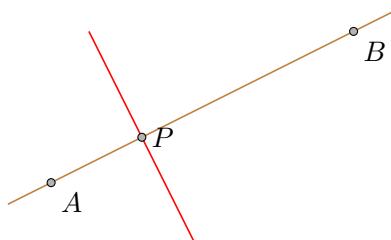
Dall'analisi della figura precedente si deduce che l'espressione che fornisce l'asse è:

$$\text{Line}(\text{med}(A, B), \text{rot}(\text{med}(A, B), A, 90))$$

□

**Problema 7.8.2** Determinare la retta perpendicolare alla retta  $AB$  e passante per il punto  $P$  della retta.

*Soluzione.* La soluzione deriva direttamente dalla figura



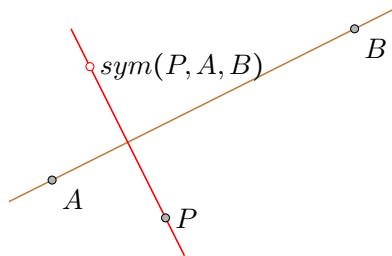
dalla quale si ricava che l'espressione della retta cercata è

$$\text{Line}(P, \text{rot}(P, A, 90))$$

□

**Problema 7.8.3** Determinare la retta perpendicolare alla retta  $AB$  e passante per il punto  $P$  esterno ad essa.

*Soluzione.* La soluzione immediata deriva dall'osservazione che la retta cercata deve passare per il punto  $P$  e per il punto simmetrico a  $P$  rispetto alla retta  $AB$ :



L'espressione della retta cercata è pertanto

$$\text{Line}(P, \text{sym}(P, A, B))$$

L'ipotesi che il punto  $P$  sia esterno alla retta  $AB$  è necessaria affinché i punti  $P$  e  $\text{sym}(P, A, B)$  siano distinti e quindi risulti definita la retta per essi.  $\square$

## 7.9 Il "triangolo"

Il triangolo è la figura che ha il ruolo principale in tutta la geometria, da quella classica greca a tutte le successive diverse definizioni ed assiomatizzazioni. Coerentemente con l'impostazione di questo capitolo, un triangolo risulta individuato dai suoi tre vertici.

**Problema 7.9.1** Stabilire se un punto  $P$  è interno ad un triangolo di dati vertici  $A, B, C$ .

**Soluzione.** Ricorrendo alla funzione *dist* ed utilizzando la formula di Erone si può calcolare l'area di un triangolo di cui sono noti i tre vertici. A questo punto basta osservare (figura 7.14) che se il punto è esterno la somma delle aree dei tre triangoli  $\triangle PAB$ ,  $\triangle PBC$ ,  $\triangle PAC$  risulta maggiore dell'area del triangolo  $\triangle ABC$ .

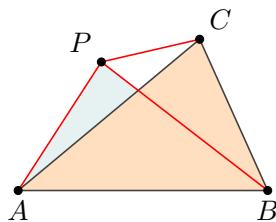


Figura 7.14: Controllo se il punto  $P$  è esterno al triangolo  $ABC$ .

Quindi la condizione che il punto  $P$  è esterno al  $\triangle ABC$  si può esprimere come segue:

$$\text{area}(ABP) + \text{area}(BCP) > \text{area}(ABC)$$

$\square$

**Problema 7.9.2** Dati i tre vertici di un triangolo, determinarne il baricentro.

*Soluzione.* Basandosi sulla sua definizione (il baricentro di un triangolo è il punto d'incontro delle mediane) e su una sua proprietà (il baricentro si trova su una mediana ed è tale che la sua distanza dal vertice è doppia della sua distanza dal punto medio) si può determinare il baricentro di un triangolo di vertici  $A, B, C$  mediante la seguente espressione:

$$\hom(A, \hom(B, C, \frac{1}{2}), \frac{2}{3})$$

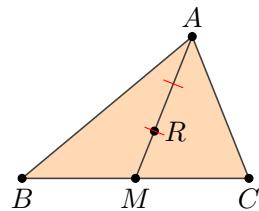


Figura 7.15: Calcolo del baricentro  $R$  del triangolo  $ABC$ .

In modo alternativo, basandosi sulla corrispondente figura 7.15, il calcolo del baricentro  $R$  di un triangolo di vertici  $A, B, C$  può essere svolto con i seguenti calcoli, dove con  $M$  si denota il punto medio fra  $B$  e  $C$ :

$$\begin{aligned} M &= \frac{1}{2}B + \frac{1}{2}C \\ R &= \frac{1}{3}A + \frac{2}{3}M \\ &= \frac{1}{3}A + \frac{2}{3}\left(\frac{1}{2}B + \frac{1}{2}C\right) \\ &= \frac{1}{3}(A + B + C) \end{aligned}$$

□

## ESERCIZI

**7.1** Dimostrare che nel sistema assiomatico di Tarski vale la seguente implicazione:

$$\mathcal{F}(A, B, C) \wedge \mathcal{F}(B, C, D) \implies \mathcal{F}(B, C, D)$$

**7.2** In un articolo <sup>4</sup> è stato proposto un insieme di operazioni di seguito elencate:

1.  $A \oplus B$  : punto medio del segmento  $AB$
2.  $A \otimes B$  : centro del quadrato di lato  $AB$  in modo tale che  $A, B$ , e  $A \otimes B$  siano orientati in verso antiorario

Analizzare le proprietà (associatività, commutatività, distributività) di questo insieme di operazioni.

**7.3** Analizzare le proprietà del seguente insieme di operazioni sui punti del piano:

1.  $A \odot B$  : punto posto sulla perpendicolare in  $B$  al segmento  $AB$ , in senso antiorario, ad una distanza dalla retta per  $A$  e  $B$  uguale alla distanza fra  $A$  e  $B$ .
2.  $A \otimes B$  : punto simmetrico di  $A$  rispetto al centro  $B$

**7.4** Mediante le relazioni *equ*, *fra*, *par*, *iso*, esprimere la condizione che

1. i tre punti  $A, B, C, D$  costituiscono i vertici di un triangolo di area nulla e perimetro non nullo
2. i tre punti  $A, B, C, D$  costituiscono i vertici di un triangolo isoscele
3. i quattro punti  $A, B, C, D$  sono allineati
4. i quattro punti  $A, B, C, D$  sono allineati ed in ordine
5. i quattro punti  $A, B, C, D$  costituiscono i vertici di un rombo
6. i quattro punti  $A, B, C, D$  costituiscono i vertici di un quadrato

**7.5** Dimostrare che

$$med(A, B) \equiv int(A, B, int(A, D, B, C), int(A, C, B, D))$$

**7.6** Stabilire il significato geometrico della seguente condizione sui quattro punti  $A, B, C, D$ :

$$(sym(B, A, C) = D) \wedge (sym(A, D, B) = C)$$

**7.7** Stabilire il significato geometrico della seguente espressione condizione sui quattro punti  $A, B, C, D$ :

$$int(A, B, C, sym(C, A, B))$$

**7.8** Esprimere la funzione *sym* in funzione della funzione *rot* e viceversa.

<sup>4</sup>Puppi Mario, *Una costruzione geometrica e la sua algebra*, Archimede, LIV (2002), pag. 150-159

**7.9** Esprimere la relazione di interposizione  $fra(P, A, B)$  di un punto  $P$  ad un segmento  $AB$  mediante le altre proprietà.

**7.10** Dati due punti  $A$  e  $B$ , dimostrare che

$$hom(A, B, 2) = hom(B, A, -1)$$

**7.11** Definire una funzione  $oltre(A, B, k)$  che calcola il punto posizionato ad una distanza  $k$  oltre  $B$  sulla semiretta  $AB$ .

**7.12** Definire una funzione  $tratto(A, B, k)$  che calcola il punto posizionato ad una distanza  $k$  oltre  $A$  sulla semiretta  $AB$ .

**7.13** Dimostrare che la funzione  $f$  descritta nel problema 7.5.1 soddisfa alla proprietà

$$f(A, B, C) = f(C, D, f(A, B, D))$$

**7.14** Determinare un punto  $C$  interno ad un segmento  $AB$  in modo da suddividerlo in proporzioni auree, ossia in modo tale che  $AB : AC = AC : CB$ .

**7.15** Determinare la distanza di un punto  $P$  dalla retta passante per i punti  $A, B$ .

**7.16** Dimostrare che la seguente sequenza di assegnazioni ha l'effetto di scambiare i due punti  $P, Q$ :

$$P \leftarrow P + Q$$

$$Q \leftarrow P - Q$$

$$P \leftarrow P - Q$$

**7.17** Dati due punti  $A, B$ , determinare altri due punti  $C, D$  in modo che il quadrilatero  $ABCD$  risulti un quadrato di diagonali  $AC$  e  $BD$ .

**7.18** Dati tre punti  $A, B, C$ , determinare un quarto punto  $D$  in modo che i quattro punti  $ABCD$ , nell'ordine, formino un parallelogramma.

**7.19** Sia  $\mathcal{A} = \{P_1, P_2, \dots, P_n\}$  un insieme di punti del piano. Stabilire cosa rappresenta l'espressione

$$\frac{1}{n} \sum_{i=1}^n P_i$$

**7.20** Basandosi sulla funzione  $dist$  che determina la distanza fra due generici punti del piano:

1. stabilire se 3 dati punti sono allineati.
2. determinare la distanza di un punto dalla retta passante per due punti

**7.21** Dati i vertici di un triangolo, determinarne il perimetro e l'area.

**7.22** Dati i vertici di un triangolo, stabilire se esso è

1. scaleno
2. isoscele

3. equilatero
4. rettangolo

**7.23** Dati i vertici di due triangoli, decidere se essi sono

1. isometrici
2. equivalenti
3. simili

**7.24** Decidere se un punto appartiene

1. ad un segmento
2. ad una retta
3. ad una circonferenza

**7.25** Decidere se un punto è interno

1. ad una circonferenza
2. ad un triangolo
3. ad un quadrilatero
4. ad una striscia individuata da un segmento con i vertici sul bordo ed ortogonale alla striscia

**7.26** Decidere se un segmento è completamente contenuto all'interno di una circonferenza.

**7.27** Decidere se

1. due segmenti sono allineati
2. due segmenti sono paralleli.
3. due rette sono parallele

**7.28** Determinare la distanza di un punto da

1. un segmento
2. una retta
3. una circonferenza
4. un cerchio
5. un triangolo

**7.29** Determinare il più piccolo segmento che

1. contiene 3 dati punti allineati
2. contiene 4 dati punti allineati

**7.30** Decidere se si intersecano

1. due segmenti
2. due circonferenze
3. due cerchi
4. due triangoli

**7.31** Decidere se un dato quadrilatero è

1. circoscrittibile da una circonferenza
2. inscrittibile in una circonferenza

**7.32** Data una sequenza di quattro punti, decidere se il quadrilatero corrispondente è convesso. Suggerimento: ogni punto deve essere esterno al triangolo formato dagli altri tre punti.

**7.33** Data la sequenza dei vertici di un quadrilatero, determinarne l'area ed il perimetro.

**7.34** Determinare la *retta passante per un dato punto e parallela ad una data retta*. Suggerimento: usare la funzione  $f$  descritta nel problema 7.5.1.

# 8

---

## METRIZZARE

---

*In generale, il compito della matematica e dei matematici consiste nell'esprimere le regolarità visibili nel mondo fisico e intellettuale e nel trovare nuovi modelli strutturali non percepibili dall'intuizione diretta e dal senso comune.*

M. Gromov, dal discorso al *Premio Balzan 1999*

Agli inizi del secolo scorso, nel 1906, M. Fréchet ha scoperto una struttura matematica molto semplice ma assolutamente generale che si fonda sul concetto intuitivo di *distanza*. Il concetto è stato formalizzato, sempre agli inizi del XX secolo, dal matematico tedesco Felix Hausdorff. Tale struttura è nota con il nome di *spazio metrico* e traduce in un contesto assiomatico l'idea intuitiva di distanza fra due elementi di un generico insieme. Un concetto vago come quello di *grado di somiglianza* fra (le immagini di) due volti di persone può essere definito in modo rigoroso facendo ricorso al concetto di spazio metrico. Proprio per la sua generalità, tale struttura si presenta in moltissime argomentazioni teoriche ed in molte applicazioni pratiche.

## 8.1 Il concetto di distanza

Nonostante sia relativamente recente, il concetto di spazio metrico ha contribuito a molte e feconde applicazioni. Calcolare la distanza fra due volti è un'operazione tipica di alcuni sistemi di ricerca di un volto in una banca dati di volti già registrati; misurare la distanza fra l'immagine di due iridi dell'occhio o di due impronte digitali è già un'applicazione sulla quale si basano alcuni sistemi di sicurezza (ad esempio di una banca) che si sbloccano quando la persona che si pone davanti viene riconosciuta come l'utente abilitato all'accesso; essere in grado di misurare la distanza fra due movimenti (non solo della loro traiettoria ma comprensivi di tutti gli attributi di dinamicità) potrebbe divenire in un prossimo futuro un meccanismo per abilitare un telefono o le chiavi dell'automobile o della casa. Testimonianza di questa situazione è data dal seguente estratto di un discorso pronunciato dal matematico Mikhael Gromov in occasione della consegna del premio Balzan nel 1999:

*Permettetemi di illustrarvi due esempi che hanno a che vedere con il mio campo d'attività. Cominciamo con la nozione corrente di distanza, riferita a due oggetti, due stati di un sistema fisico o biologico, o due idee astratte più o meno lontane fra loro. In questo caso, un matematico parlerebbe di uno spazio astratto degli oggetti in questione, detti punti, e considererebbe la distanza come l'attribuzione di un determinato numero positivo a ogni coppia di punti nello spazio. Nello spazio tridimensionale che tutti conosciamo, la distanza fra due punti è data dalla misura del segmento più corto che li collega. Ma questa nozione avrebbe un valore puramente accademico se abitaste in una zona montuosa, dove una definizione di distanza senz'altro più appropriata sarebbe la lunghezza del percorso più breve esistente fra due luoghi assegnati. Un esempio più astratto è offerto dalla linguistica matematica: in questo caso, i nostri punti sono delle frasi, vale a dire sequenze di lettere, ad esempio quelle dell'alfabeto inglese. Ora, supponiamo che vogliate definire la distanza fra due frasi. Naturalmente, la definizione deve essere adatta allo scopo che avete in mente. In questo caso, una definizione estremamente semplice ma altrettanto utile è fornita dalla metrica di Hamming, secondo cui si prendono in considerazione frasi di una determinata lunghezza - diciamo, ad esempio, esattamente di cento lettere. La distanza fra due frasi di questo tipo si misura contando le posizioni in cui esse presentano lettere diverse. Supponiamo ora di avere davanti a noi uno spazio metrico di questo tipo, vale a dire una collezione di numeri che rappresentano le distanze. Come possiamo ottenere un'immagine geometrica partendo da questi numeri? Supponiamo, ad esempio, di isolare  $10^3$  punti su una zona montuosa e scrivere una tabella di  $10^6$  numeri corrispondenti alle distanze (più precisamente  $(10^6 - 10^3)/2$ , in quanto la distanza è simmetrica e si annulla su punti identici). Possiamo determinare, ad esempio, le dimensioni del monte più alto semplicemente concentrandoci su questi  $10^6$  numeri? Oppure, disponendo di due tabelle cosiddette, siamo in grado di dire se i due paesaggi sono simili o se invece sono diversi? Quest'ultima domanda ci introduce al livello d'astrazione successivo: abbiamo bisogno di definire una distanza fra spazi metrici assegnati. Dunque, gli spazi stessi sono considerati come punti, che tutti insieme formano un grande oggetto astratto, lo spazio degli spazi metrici.*

Ma per quanto astratto possa sembrare, non possiamo fare a meno di occuparci di questi spazi nella matematica pura nonché nelle sue applicazioni. Ad esempio, se volessimo creare un programma informatico in grado di distinguere i volti umani o le espressioni del volto, dovremmo codificare ogni volto mediante una tabella di numeri, ad esempio, immaginandolo come una specie di paesaggio con la distanza definita dal percorso più breve. Dopodiché, dovremmo definire in qualche modo intelligente una nozione di distanza fra i volti. Il programma deve dirci se due volti, convertiti in spazi metrici, si assomigliano o meno. Naturalmente, non potremmo lavorare direttamente con i numeri. Nemmeno nello spazio linguistico saremmo in grado di scrivere tutti i numeri necessari: sono circa  $26^{200}$  - molti di più di tutte le particelle elementari che si trovano nell'universo! Perciò dovremmo sviluppare un linguaggio in grado di esprimere caratteristiche essenziali degli spazi, i cosiddetti invarianti metrici. Dopodiché, potremmo confrontare gli spazi sulla base di questi invarianti. Non esiste una ricetta universale per creare tale linguaggio. Noi tentiamo ogni strada, attingendo liberamente dall'intuizione geometrica basata sulla nostra esperienza dello spazio tridimensionale. Ciò ci permette una buona partenza, ma abbiamo ancora molta strada da percorrere.

## 8.2 Gli spazi metrici

Il concetto di spazio metrico generalizza la nozione di distanza geometrica di due punti su una retta, su un piano e nello spazio. Un insieme costituisce uno *spazio metrico* se per ogni coppia di elementi dell'insieme è definita una funzione a valori numerici che rappresenta la *distanza* fra due generici elementi dell'insieme. Coerentemente con il significato usuale di distanza, si impone che tale funzione soddisfi a delle specifiche proprietà. Naturalmente, al fine di avere una generalizzazione coerente, gli assiomi che stanno alla base di questa struttura devono rispettare le proprietà di cui gode l'usuale nozione di distanza. Formalmente tale concetto viene descritto mediante la seguente definizione.

**DEFINIZIONE 3.** Uno *spazio metrico* è una coppia  $(E, d)$  dove  $E$  è un generico insieme e  $d$  è una funzione

$$d : E \times E \rightarrow \mathbb{R}$$

soddisfacente alle seguenti proprietà <sup>1</sup>:

- $M_1$ .  $d(x, y) = 0$  se e solo se  $x = y$ , per ogni  $x, y \in E$
- $M_2$ .  $d(x, y) = d(y, x)$  per ogni  $x, y \in E$
- $M_3$ .  $d(x, z) \leq d(x, y) + d(y, z)$  per ogni  $x, y, z \in E$

La proprietà  $M_3$  è detta *diseguaglianza triangolare*, con evidente analogia al caso geometrico. In generale, gli elementi di  $E$  si chiamano *punti*, la funzione  $d$  si chiama *metrica* ed il numero reale  $d(x, y)$  si chiama *distanza* fra  $x$  ed  $y$ .

<sup>1</sup>Se al posto della condizione  $M_1$  si prende la meno esigente  $M'_1$ .  $d(x, x) = 0$  per ogni  $x \in E$ , si parla di *spazio pseudometrico*.

Una immediata conseguenza dei tre assiomi sopra definiti è rappresentata dal seguente importante

**TEOREMA 4.** Una metrica è una funzione non negativa.

*Dimostrazione.* Ponendo  $z = x$  nella  $M_3$ , si ha  $d(x, x) \leq d(x, y) + d(y, x)$ ; per la  $M_1$  si ha  $d(x, x) = 0$  e, per la  $M_2$ , si ha  $d(y, x) = d(x, y)$  e quindi  $0 \leq 2 d(x, y)$  da cui  $d(x, y) \geq 0$  e per la generalità dei punti  $x$  e  $z$  si conclude.  $\square$

La tematica relativa agli spazi metrici riguarda principalmente due aspetti: *definire* una distanza, ossia specificare un criterio che definisce la distanza rispettando i vincoli relativi ad un generico spazio metrico; *misurare* la distanza fra due dati oggetti in base ad un prefissato criterio di misura di distanza.

Nella teoria e nelle applicazioni si incontrano con frequenza degli interessanti esempi di spazi metrici.

*Esempio 8.2.1 -* Sia  $A$  un generico insieme. Indipendentemente dalla natura degli elementi di  $A$  e da eventuali strutture vigenti su  $A$ , si può definire su  $A$  la seguente metrica (detta *metrica banale* o *metrica discreta*):

$$d(x, y) = \begin{cases} 0 & \text{se } x = y \\ 1 & \text{se } x \neq y \end{cases}$$

*Esempio 8.2.2 -* Sulla retta numerica, ossia prendendo  $E = \mathbb{R}$ , si definisce la seguente metrica:

$$d(x, y) = |x - y|$$

Naturalmente, questa definizione di metrica presuppone che su  $\mathbb{R}$  sia definita l'operazione di sottrazione e l'operazione di valore assoluto.

*Esempio 8.2.3 -* L'insieme *Date* delle date, ciascuna delle quali individuata mediante una terna (*giorno, mese, anno*), può essere dotato della struttura di spazio metrico definendo

$$d(x, y) = \text{numero di giorni che intercorrono fra le due date } x \text{ ed } y$$

Ad esempio:  $d((20, \text{novembre}, 2019), (3, \text{dicembre}, 2019)) = 13$ . Questa metrica viene frequentemente usata nelle applicazioni di tipo commerciale.

*Esempio 8.2.4 -* L'insieme *String* delle stringhe costituite da una sequenza di caratteri di lunghezza fissata, può essere dotato della struttura di spazio metrico definendo

$$d(x, y) = \text{numero di caratteri diversi in } x \text{ ed } y \text{ in posizioni corrispondenti}$$

Ad esempio:  $d(\text{"romano"}, \text{"ramino"}) = 2$ . Nel caso di stringhe di lunghezze diverse si può adottare questa definizione di distanza considerando diversi i caratteri sui quali una stringa eccede in lunghezza l'altra; ad esempio:  $d(\text{"calibro"}, \text{"libro"}) = 7$ ; evidentemente questa è una definizione poco utile; più interessante sarebbe una funzione determina la distanza considerando un

allineamento ottimale fra le due stringhe. Questa metrica, detta di *Hamm*ing, sarà presentata più avanti, assieme ad altre metriche sulle stringhe, più interessanti e più utili.

**Esempio 8.2.5 -** L'insieme  $\mathbb{Z}^2$  delle coppie  $(m, n)$  di numeri interi può essere metrizzato mediante la *metrica del cavallo degli scacchi* definendo la distanza fra due caselle di coordinate  $(m_1, n_1)$  e  $(m_2, n_2)$  come il minimo numero di mosse del cavallo per spostarsi da una all'altra.

**Esempio 8.2.6 -** Nell'insieme  $\mathcal{C}([a, b])$  delle funzioni reali di una variabile reale definite e continue nell'intervallo  $[a, b]$  si possono definire le seguenti metriche:

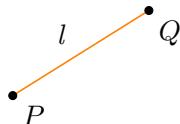
$$d_1(f, g) = \max_{x \in [a, b]} |f(x) - g(x)|$$

$$d_2(f, g) = \int_a^b |f(x) - g(x)| dx$$

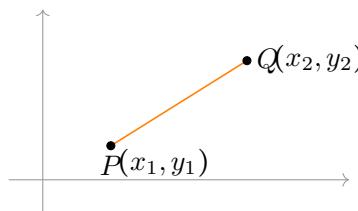
### 8.3 Metrizzare il piano

L'insieme che più naturalmente si presta ad essere metrizzato è costituito dal piano punteggiato. Negli esempi che seguono sono descritte diverse metrizzazioni. Oltre alla banale e poco utile metrica descritta nell'esempio 8.2.1, si possono definire altre metriche che si basano sulle proprietà specifiche del piano. Alcune possibili metriche sul piano punteggiato sono descritte negli esempi che seguono.

**Esempio 8.3.1 -** Il modo più naturale per metrizzare il piano consiste nel fissare un'unità di misura e definire la distanza fra due generici punti  $P$  e  $Q$  come la lunghezza  $l$  del segmento  $\overline{AB}$  congiungente i due punti. Si parla in questo caso di *distanza euclidea*.



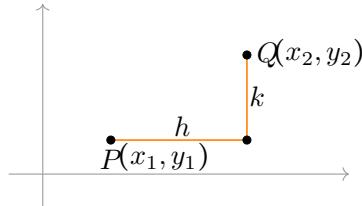
**Esempio 8.3.2 -** Consideriamo il piano punteggiato sul quale sia definito un sistema di riferimento cartesiano ortogonale; in questo modo un generico punto del piano risulta individuato mediante le sue coordinate, stabilendo una corrispondenza biunivoca fra i punti del piano e l'insieme  $\mathbb{R}^2$  delle coppie di numeri reali. Consideriamo due generici punti  $P = (x_1, y_1)$ ,  $Q = (x_2, y_2)$ .



In questo contesto la distanza euclidea corrisponde alla lunghezza del segmento che congiunge i due punti  $P$  e  $Q$  ed è espressa da (applicando il teorema di Pitagora)

$$d_E(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

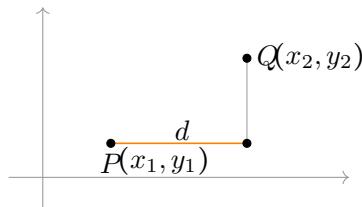
*Esempio 8.3.3* - Nel piano dotato di un sistema di riferimento cartesiano ortogonale la *distanza Manhattan* è definita come la lunghezza del più breve percorso formato da segmenti paralleli agli assi, come descritto nella figura che segue, dove  $d(P, Q) = h + k$ .



Oltre al percorso evidenziato in figura esiste un altro percorso di stessa lunghezza minima composto dai segmenti paralleli a quelli indicati; si otterrebbe la stessa distanza se si ammette come percorso una qualsiasi spezzata formata da segmenti paralleli agli assi. Per entrambi questi percorsi la distanza Manhattan fra i due punti  $P = (x_1, y_1)$ ,  $Q = (x_2, y_2)$  è espressa mediante l'espressione

$$d_M(P, Q) = |x_1 - x_2| + |y_1 - y_2|$$

*Esempio 8.3.4* - Un altro modo per definire la distanza fra due punti del piano dove sia definito un sistema di riferimento cartesiano ortogonale è indicato nella figura e nella corrispondente definizione che seguono (*distanza del massimo*).

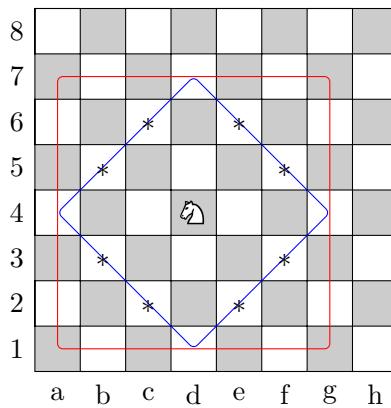


$$d_m(P, Q) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$$

*Osservazione.* La metrica Manhattan e la metrica del massimo risultano di particolare interesse computazionale in quanto risultano facilmente calcolabili poiché richiedono il calcolo di operazioni elementari come l'addizione, la sottrazione ed il valore assoluto.

*Esempio 8.3.5* - Come applicazione delle metriche definite negli esempi precedenti consideriamo il problema di stabilire se una casella di una scacchiera

è sotto attacco di un cavallo. Individuiamo ciascuna casella mediante una coppia di coordinate intere; ad esempio, la casella all'angolo in basso a sinistra è individuata, al posto della tradizionale notazione scacchistica  $a1$ , mediante la coppia di coordinate  $(1, 1)$ . Nella figura che segue è riportato un cavallo bianco e sono indicate con un asterisco le posizioni che esso attacca. Con la linea blu sono indicate le caselle aventi distanza Manhattan  $d_M$  dal cavallo bianco pari a 3 mentre in colore rosso sono indicate le caselle aventi una distanza  $d_m$  del massimo pari a 3.



Un cavallo posto nella posizione  $A = (i, j)$  attacca la posizione  $B = (k, l)$ ,  $i, j, k, l \in \{1, \dots, 8\}$  se è soddisfatta la condizione

$$(d_M(A, B) = 3) \wedge (d_m(A, B) \neq 3)$$

## 8.4 Metrizzare gli spazi n-dimensionali

Una caratteristica linea di tendenza della Matematica e della Scienza in generale è quella della generalizzazione dei concetti. Per tradizione scolastica la Geometria viene solitamente ambientata e studiata nel piano (bidimensionale). In particolare viene definito un sistema di assi cartesiani mediante il quale un punto risulta individuato dalla coppia  $(x, y)$  delle sue coordinate. Generalizzando in modo spontaneo, svincolandosi dal piano bidimensionale ed anche dallo spazio tridimensionale, si possono considerare gli spazi  $n$ -dimensionali nei quali un *punto* risulta individuato da una  $n$ -pla di numeri; per questo motivo lo spazio viene denotato con  $\mathbb{R}^n$ .

Dati due generici punti  $P = (x_1, \dots, x_n)$ ,  $Q = (y_1, \dots, y_n)$  dello spazio  $n$ -dimensionale  $\mathbb{R}^n$  si possono definire le seguenti metriche:

$$\begin{aligned}
 d_1(P, Q) &= \sum_{i=1}^n |x_i - y_i| \\
 d_2(P, Q) &= \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \\
 d_p(P, Q) &= \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad p \in \mathbb{R}, p \geq 1 \\
 d_\infty(P, Q) &= \lim_{p \rightarrow \infty} \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} = \max_{i \in \{1, \dots, n\}} |x_i - y_i|
 \end{aligned}$$

Si riconosce facilmente che la metrica  $d_1$  è una generalizzazione della metrica Manhattan  $d_M$ ,  $d_2$  una generalizzazione della metrica euclidea  $d_E$  e  $d_\infty$  una generalizzazione della metrica  $d_m$  viste nel precedente paragrafo. La  $d_p$  è detta *metrica di Minkowsky*.

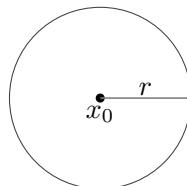
## 8.5 Definizioni in uno spazio metrico

La nozione di spazio metrico rappresenta una naturale generalizzazione dell'usuale concetto di distanza fra due punti (della retta, del piano o dello spazio). Per tradizione e per ragioni tipografiche uno spazio metrico viene solitamente rappresentato graficamente con riferimento al piano usuale. Tale modalità di rappresentazione offre il vantaggio di suggerire delle importanti definizioni derivate da usuali concetti del piano geometrico, quali *segmenti*, *diametri*, *linee*, ... .

In un generico spazio metrico (in particolare nel piano punteggiato caratterizzato dalla distanza euclidea, dal quale si prende spunto per generalizzare) molte entità geometriche quali segmenti, circonferenze, rette, possono essere definite come *luogo geometrico (di punti)*, ossia come insieme di punti del piano soddisfacenti ad una data proprietà. Per comodità conviene prendere una proprietà facilmente indagabile, basata sulla nozione di distanza; lasciandosi guidare dalle analoghe definizioni nel piano, si possono allora definire le rette, i segmenti e le circonferenze come segue.

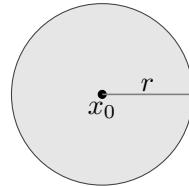
**DEFINIZIONE 4.** In uno spazio metrico  $(E, d)$  dicesi *circonferenza* di centro  $x_0 \in E$  e *raggio*  $r \in \mathbb{R}_+$  l'insieme dei punti equidistanti  $r$  dal centro  $x_0$ , ossia l'insieme

$$\{x \in E : d(x, x_0) = r\}$$



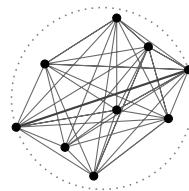
Dicesi *cerchio* o *disco* di centro  $x_0 \in E$  e raggio  $r \in \mathbb{R}_+$  l'insieme

$$\{x \in E : d(x, x_0) \leq r\}$$



Dicesi *diametro* di un sottoinsieme  $A$  di uno spazio metrico  $(E, d)$  l'estremo superiore dell'insieme delle distanze fra tutte le coppie di suoi generici elementi, ossia il numero

$$\text{diam}(A) = \sup_{x, y \in A} d(x, y)$$



Un insieme dicesi *limitato* se ha diametro finito.

Dicesi *segmento* di estremi  $a$  ed  $b$  il luogo geometrico dei punti  $x$  tali che

$$d(x, a) + d(x, b) = d(a, b)$$



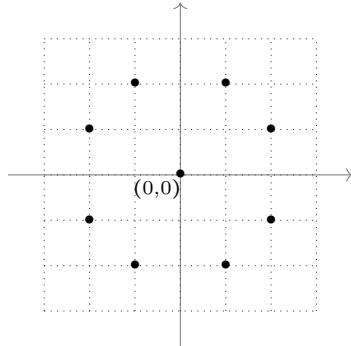
Dicesi *retta* passante per i due punti  $a, b \in E$  il luogo geometrico dei punti  $x \in E$  tali che, posto  $\alpha = d(a, x), \beta = d(b, x), \gamma = d(a, b)$ , si ha

$$\max(\alpha, \beta, \gamma) = (\alpha + \beta + \gamma)/2$$



*Osservazione.* I punti che compaiono nelle figure che illustrano la precedente definizione devono essere intesi come elementi di un generico spazio metrico; possono rappresentare: punti del piano, punti dello spazio  $n$ -dimensionale, stringhe di caratteri, immagini di un volto.

*Esempio 8.5.1 -* La figura che segue illustra il disco di  $\mathbb{Z}^2$  di centro  $(0,0)$  e raggio 1 con la *metrica del cavallo*.



## 8.6 Distanza fra figure di uno spazio metrico

Coerentemente con la terminologia della geometria, si chiama *figura* un sottoinsieme di punti di uno spazio metrico. Ad esempio sono figure le seguenti: gli insiemi singoletti formati da un solo punto dello spazio metrico; le circonferenze, le rette, gli insiemi dei punti delimitati da una circonferenza.

In generale l'insieme delle figure di uno spazio metrico  $(E, d)$  è un insieme, che viene denotato con  $\mathcal{F}_E$ , ed è costituito da  $\wp(E)$ , insieme delle parti di  $E$ . Vogliamo ora definire dei criteri per stabilire la distanza fra due generiche figure, ossia vogliamo metrizzare l'insieme  $\mathcal{F}_E$ . Basandosi sulla funzione  $\delta$  che definisce la distanza fra un punto ed un insieme, si può definire la distanza fra due qualsiasi figure adottando la seguente

**DEFINIZIONE 5.** Sia  $(E, d)$  uno spazio metrico,  $\mathcal{F}_E$  l'insieme delle figure di  $E$ ,  $\delta : E \times \mathcal{F}_E \rightarrow \mathbb{R}$  la funzione definita da  $\delta(x, A) \stackrel{\text{def}}{=} \inf_{y \in A} d(x, y)$ . Dicesi *metrica di Hausdorff* la funzione  $d_H : \mathcal{F}_E^2 \rightarrow \mathbb{R}$  definita da

$$d_H(A, B) \stackrel{\text{def}}{=} \max\{\sup_{x \in A} \delta(x, B), \sup_{y \in B} \delta(y, A)\}$$

*Esempio 8.6.1 -* L'insieme delle entità geometriche del piano risulta metrizzato considerando la metrica indotta dalla metrica euclidea mediante la definizione di distanza fra due insiemi di uno spazio metrico. La figura 8.1 illustra la distanza euclidea fra un rettangolo ed un ellisse del piano.

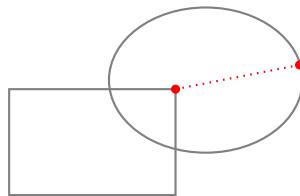


Figura 8.1: Distanza fra due figure.

Dato uno spazio metrico  $E$ , un sottoinsieme  $A \subseteq E$ , un punto  $x \in E$ , la precedente definizione di distanza fra figure suggerisce una coerente definizione di *distanza* fra  $x$  ed  $A$ , definita da

$$\delta(x, A) \stackrel{\text{def}}{=} d_H(\{x\}, A)$$

ossia

$$\delta(x, A) \stackrel{\text{def}}{=} \inf_{y \in A} d(x, y)$$

In pratica,  $\delta(x, A)$  rappresenta la distanza del punto di  $A$  più vicino ad  $x$ . Ad essere rigorosi, la scrittura  $\delta(x, A)$  risulta impropria, data la non omogeneità degli argomenti  $x$  ed  $A$ ; si riacquista la coerenza formale intendendo questa scrittura come una abbreviazione di  $\delta(\{x\}, A)$ .

## 8.7 Criteri per definire la distanza

In molte applicazioni un criterio utilizzato per definire distanza fra due elementi consiste nel trasformare un elemento nell'altro e misurare la complessità della trasformazione; a questo scopo viene prefissato un insieme di trasformazioni elementari e si definisce come distanza il minor numero di trasformazioni elementari necessario per trasformare un elemento nell'altro. Per garantire la condizione di simmetria della metrica, ogni operazione elementare di trasformazione, nell'insieme delle trasformazioni elementari ammesse, ha una simmetrica operazione (ad esempio: *inserimento-eliminazione*, *alza-abbassa*, *ruota a sinistra-ruota a destra*). Questo metodo per la definizione della distanza dà luogo ad interessanti algoritmi per il calcolo della stessa.

*Esempio 8.7.1* - Consideriamo una sequenza  $a = [a_1, a_2, \dots, a_n]$  ed una sua permutazione  $b = [b_1, b_2, \dots, b_n]$ . Ammettiamo come operazione di trasformazione lo scambio fra due generici elementi. Con queste premesse si può definire la distanza fra  $a$  e  $b$  come il minor numero di scambi necessari per trasformare una sequenza nell'altra.

## 8.8 Metriche sulle stringhe

Una tipica funzione degli elaboratori di testi è costituita dalla funzione di correzione automatica che consiste nel ricercare in un dizionario le parole, segnalando quelle non appartenenti al dizionario. Una funzionalità collegata a questa consiste nel proporre un insieme di possibili parole (appartenenti

ad un prefissato dizionario) *vicine* alla parola non ritrovata nel dizionario, in modo da suggerire allo scrivente delle opzioni sulla scelta della parola da inserire al posto di quella non ritrovata. Problemi analoghi sorgono in biologia molecolare per determinare il grado di similarità di due sequenze di DNA, nei sistemi di riconoscimento vocale e nei sistemi basati sulla ricerca di similarità tra immagini, suoni, testi, etc. Tutte queste problematiche possono essere uniformate, definite in modo preciso e risolte dotando l'insieme delle stringhe di una struttura di spazio metrico.

Applichiamo ora le precedenti considerazioni relative agli spazi metrici al caso in cui l'insieme  $E$  sia costituito dall'insieme delle stringhe su un dato alfabeto. In questo caso il concetto di distanza fra stringhe esprime un indice di similarità fra due stringhe.

Si possono adottare diversi criteri per misurare la distanza fra due stringhe. Per motivazioni di carattere pratico si considera una particolare classe di funzioni di distanza che fanno riferimento a delle funzioni di trasformazione fra stringhe. Fra queste le più interessanti e frequenti sono le seguenti:

$I$ : *inserimento* nella stringa  $s$  alla posizione  $k$  del carattere  $c$ . In notazione funzionale si scrive  $I(s, k, c)$ . Ad esempio, inserendo nella stringa "libro" alla posizione 4 il carattere 'e' si ottiene la stringa "libero", ossia  $I("libro", 4, 'e') = "libero"$ .

$D$ : *cancellazione* dalla stringa  $s$  del carattere di posizione  $k$ . In notazione funzionale si scrive  $I(s, k, c)$ . Ad esempio, cancellando dalla stringa "libro" il carattere di posizione 4 si ottiene la stringa "libro", ossia  $D("libro", 4) = "libro"$ .

$S$ : *sostituzione* nella stringa  $s$  del carattere di posizione  $k$  con il carattere  $c$ . In notazione funzionale si scrive  $S(s, k, c)$ . Ad esempio, sostituendo nella stringa "libro" il carattere di posizione 3 con il carattere 't', si ottiene la stringa "litro", ossia  $S("libro", 3, 't') = "litro"$ .

$T$ : *trasposizione* nella stringa  $s$  del carattere di posizione  $k$  con il carattere successivo. In notazione funzionale si scrive  $T(s, k)$ . Ad esempio, trasponendo nella stringa "capra" il carattere di posizione 3 con il successivo), si ottiene la stringa "carpa", ossia  $T("capra", 3) = "carpa"$ .

Siamo ora pronti per dare la seguente

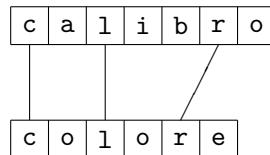
**DEFINIZIONE 6.** Fissato un insieme  $\mathcal{T}$  di operazioni di trasformazione fra stringhe, dicesi *distanza* fra due stringhe  $x$  ed  $y$  il minor numero di operazioni di trasformazione necessarie e sufficienti per trasformare la stringa  $x$  nella stringa  $y$ .

Per opportuni insiemi  $\mathcal{T}$  di funzioni di trasformazione la nozione di distanza sopra definita costituisce una funzione soddisfacente agli assiomi di spazio metrico. I casi più interessanti sono di seguito elencati.

**Distanza di Hamming** : prendendo  $\mathcal{T} = \{sostituzione\}$  ed assumendo l'ipotesi  $|x| = |y|$  si ottiene la distanza di Hamming fra due stringhe, denotata

soltamente con  $d_H$ ; essa rappresenta il numero di caratteri in cui le due stringhe si differenziano. Ad esempio,  $d_H("lavoro", "tavolo") = 2$ . La metrica di *Hamming* risulta di una certa utilità in alcune applicazioni, ad esempio nella gestione delle interrogazioni nelle *basi di dati*.

**Distanza della più lunga sottosequenza comune** : sono ammesse le operazioni  $\mathcal{T} = \{\text{inserimento}, \text{cancellazione}\}$ . Il nome di questa funzione di distanza deriva dal fatto che essa si basa sulla lunghezza della più lunga sequenza di caratteri accoppiabili fra le due stringhe ed esprime il numero di caratteri non accoppiabili. Pertanto, vale evidentemente la relazione  $d(x, y) \leq |x| + |y|$ . Ad esempio,  $d("calibro", "colore") = 7$  come si desume dal seguente schema di corrispondenza fra i caratteri.



## 8.9 La distanza di edit

Nel 1965 dallo scienziato russo Vladimir Levenshtein definì una interessante metrica per determinare il grado di similarità di due stringhe; questa metrica esprime il numero di errori di battitura che *separano* una stringa (considerata errata) da un'altra (considerata corretta)<sup>2</sup>. I tipici errori atomici di digitazione con la tastiera sono:

- digitazione di un tasto in più
- mancata digitazione di un tasto
- digitazione di un tasto sbagliato

La *distanza di edit* fra due stringhe è definita come il minor numero di operazioni di *inserimento*, *cancellazione*, *sostituzione* di caratteri necessarie a trasformare una stringa nell'altra. Questa distanza corrisponde all'insieme di trasformazioni  $\mathcal{T} = \{\text{inserimento}, \text{cancellazione}, \text{sostituzione}\}$ .

*Esempio 8.9.1* - La stringa **albero** può essere trasformata nella parola **libri** mediante la seguente sequenza di trasformazioni elementari:

$$\text{albero} \xrightarrow{D(1)} \text{lbero} \xrightarrow{I(2,i)} \text{libero} \xrightarrow{D(4)} \text{libro} \xrightarrow{S(5,i)} \text{libri}$$

La lunghezza di questa sequenza di trasformazioni è pari a 4 e, poiché non ne esistono di più corte, la distanza di edit fra le due stringhe **albero** e **libri** è uguale a 4.

<sup>2</sup>Al di là della convenzionale attribuzione asimmetrica *errata/corretta*, le due stringhe hanno un ruolo simmetrico, coerentemente con l'assioma di simmetria degli spazi metrici.

Il seguente procedimento adotta la tecnica della *programmazione dinamica* per determinare la distanza di edit fra due stringhe e le sequenze minimali di operazioni elementari di edit che permettono di trasformare una stringa in un'altra.

Siano  $x = x_1 \dots x_m$ ,  $y = y_1 \dots y_n$  due stringhe. Per trasformare  $x$  in  $y$  si può adottare una delle seguenti tre trasformazioni:

- accodare  $y_n$  alla fine di  $x$  :  $x \rightarrow x_1 \dots x_m y_n$  e poi trasformare  $x_1 \dots x_m$  in  $y_1 \dots y_{n-1}$
- eliminare  $x_m$  da  $x$  :  $x \rightarrow x_1 \dots x_{m-1}$  e poi trasformare  $x_1 \dots x_{m-1}$  in  $y_1 \dots y_n$
- cambiare  $x_m$  in  $y_n$  (se diversi):  $x \rightarrow x_1 \dots x_{m-1} y_n$  e poi trasformare  $x_1 \dots x_{m-1}$  in  $y_1 \dots y_{n-1}$

Ciò suggerisce uno schema algoritmico di tipo ricorsivo caratterizzato dal seguente sottoproblema di trasformazione di un prefisso di  $x$  in un prefisso di  $y$ :

$$\text{Trasformare } x_1 \dots x_i \text{ in } y_1 \dots y_j$$

Per la soluzione di questo sottoproblema costruiamo una matrice  $M[0..m][0..n]$  definita come segue (gli indici 0 corrispondono alla stringa vuota):

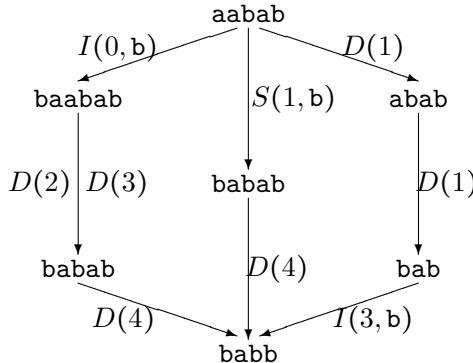
$$M[i][j] = \begin{cases} \text{minimo numero di operazioni elementari di edit richieste per} \\ \text{trasformare } x_1 \dots x_i \text{ in } y_1 \dots y_j, \text{ per ogni } 0 \leq i \leq m, 0 \leq j \leq n, \\ \text{ossia } M[i][j] = d_E(x_1 \dots x_i, y_1 \dots y_j) \end{cases}$$

La costruzione della matrice  $M$  si basa sulle seguenti proprietà:

- $M[0][j] = j$ , in quanto l'unico modo per trasformare la stringa vuota  $\epsilon$  in  $y_1 \dots y_j$ , consiste nell'aggiungere i  $j$  caratteri  $y_1, \dots, y_j$
- $M[i][0] = i$ , in quanto l'unico modo per trasformare  $x_1 \dots x_i$  nella stringa vuota consiste nell'eliminare gli  $i$  caratteri  $x_1, \dots, x_i$
- per  $i, j \geq 1$ :  $M[i][j] = \min\{M[i][j-1] + 1, M[i-1][j] + 1, M[i-1][j-1] + c(x_i, x_j)\}$ , dove  $c(x_i, x_j) = 0$  se  $x_i = x_j$ , 1 altrimenti

Dalla definizione della matrice  $M$  risulta che il valore  $M[m][n]$  costituisce la distanza di edit fra le due stringhe  $x$  ed  $y$ . Inoltre, ogni percorso dalla posizione  $(m, n)$  alla posizione  $(0, 0)$  determina una sequenza di operazioni di edit che trasformano una stringa in un'altra. Si può dimostrare che la complessità di tempo dell'algoritmo è  $O(mn)$  e, per lunghezze equivalenti delle due stringhe, la complessità dell'algoritmo è quadratica.

**Esempio 8.9.2 -** Determiniamo la distanza di edit fra le due stringhe **aabab** nella stringa **babb** e la sequenza delle operazioni elementari che trasformano una stringa in un'altra. La trasformazione fra le due stringhe può essere descritta mediante il grafo riportato nella figura 8.2.

Figura 8.2: Alcuni percorsi di trasformazione della stringa **aabab** in **babb**.

Nella tabella 8.3 è riportata la matrice della distanza di edit fra le due stringhe.

|            | $\epsilon$ | b        | a | b | b        |
|------------|------------|----------|---|---|----------|
| $\epsilon$ | 0          | <u>1</u> | 2 | 3 | 4        |
| a          | <u>1</u>   | <b>1</b> | 1 | 2 | 3        |
| a          | 2          | 2        | 1 | 2 | 3        |
| b          | <u>3</u>   | <u>2</u> | 2 | 1 | 2        |
| a          | <u>4</u>   | <b>3</b> | 2 | 2 | 2        |
| b          | 5          | 4        | 3 | 2 | <b>2</b> |

Figura 8.3: Matrice della distanza di edit fra le due stringhe **aabab** e **babb**. I numeri in grassetto **1** e **3** sono stati calcolati in funzione dei corrispondenti numeri contigui evidenziati in sottolineato. Il numero **2**, posto all'estremità in basso a destra della matrice, costituisce la distanza di edit fra le due stringhe.

## 8.10 Metriche equivalenti

Intuitivamente, due metriche equivalenti esprimono il fatto che due punti vicini rispetto ad una metrica risultano vicini anche rispetto all'altra; l'informatica, sensibile alle questioni di efficienza degli algoritmi, sceglie ed applica le metriche che vengono calcolate più efficientemente. Questa osservazione viene sfruttata nei programmi di grafica per ricercare velocemente gli elementi grafici vicini ad un dato punto. Formalmente questo concetto viene espresso per mezzo della seguente

**DEFINIZIONE 7.** Due metriche  $d_1$  e  $d_2$  su un insieme  $E$  si dicono *equivalenti* se esistono due numeri reali positivi  $a$ ,  $b$  tali che

$$a d_1(x, y) \leq d_2(x, y) \leq b d_1(x, y) \text{ per ogni } x, y \in E$$

Dato uno spazio metrico  $(E, d)$ , un generico sottoinsieme non vuoto  $A$  di  $E$  può essere considerato spazio metrico dotandolo della metrica di  $E$  ristretta al

dominio  $A \times A$ ; tale metrica viene detta *metrica indotta*. Pertanto, un generico sottoinsieme di uno spazio metrico verrà considerato tacitamente uno spazio metrico dotato della metrica indotta.

*Esempio 8.10.1 -* Le metriche  $d_1$ ,  $d_2$ ,  $d_p$  e  $d_\infty$  definite su  $\mathbb{R}^n$  nel paragrafo 8.4 sono equivalenti. Nel caso  $n = 1$  queste metriche risultano uguali.

## 8.11 Prodotto di spazi metrici

Un problema di particolare rilevanza pratica consiste nel determinare la distanza fra due elementi composti da elementi di spazi metrici. Il problema trova soluzione nel contesto del prodotto di due spazi metrici. E' naturale attendersi che le metriche definite sugli spazi componenti possano indurre una metrica sullo spazio prodotto. In effetti, il problema trova risposta affermativa (multipla) in base al seguente

**TEOREMA 5.** Siano  $(E, d_E)$ ,  $(F, d_F)$  due spazi metrici. L'insieme prodotto  $E \times F$  risulta spazio metrico adottando, ad esempio, una delle seguenti metriche:

1.  $d_{E \times F}((x_1, y_1), (x_2, y_2)) = d_E(x_1, x_2) + d_F(y_1, y_2)$
2.  $d_{E \times F}((x_1, y_1), (x_2, y_2)) = \max\{d_E(x_1, x_2), d_F(y_1, y_2)\}$
3.  $d_{E \times F}((x_1, y_1), (x_2, y_2)) = \sqrt{d_E^2(x_1, x_2) + d_F^2(y_1, y_2)}$

Si lascia per esercizio la facile dimostrazione che le funzioni sopra definite soddisfano agli assiomi di spazio metrico. Il risultato di questo teorema è facilmente estendibile al caso di prodotti di più spazi metrici.

## 8.12 Interrogazioni negli spazi metrici

Gli insiemi dotati di una struttura di spazio metrico costituiscono un ambiente ideale dove eseguire delle interrogazioni basate su un criterio di similarità; queste forme di elaborazione possono essere considerate come una generalizzazione dei procedimenti di ricerca basati sul criterio di uguaglianza.

Dato uno spazio metrico  $E$  ed un elemento  $x_0 \in E$ , le forme più tipiche di interrogazioni assumono le seguenti forme:

- $Q_1$ : Determinare l'elemento  $x \in E$  più vicino a  $x_0$ .
- $Q_2$ : Assegnato un numero naturale positivo  $k$ , determinare i  $k$  elementi di  $E$  più vicini a  $x_0$ .
- $Q_3$ : Fissato un numero reale positivo  $r$  (raggio) determinare gli elementi di  $E$  che hanno una distanza da  $x_0$  inferiore o uguale a  $r$ .

Sebbene possa sembrare strano, le questioni teoriche come quelle relative agli spazi metrici hanno moltissime applicazioni pratiche, come attestano i casi descritti nel seguente esempio.

*Esempio 8.12.1 -* In un motore di ricerca l'utente digita la parola **logarimo**; poiché tale parola non è memorizzata nel ricco dizionario a disposizione del sistema, esso ipotizza che l'utente abbia fatto qualche errore di digitazione e,

dopo pochi millesimi di secondo, viene presentata una lista di parole sensate: **algoritmo**, **logaritmo**, ... e l'utente viene sollecitato a sceglierne una fra queste: il sistema ha determinato l'insieme delle parole del suo dizionario avente una distanza di edit minore o uguale a 3 e le ha presentate all'utente.

*Esempio 8.12.2 -* Un'applicazione di un'agenzia immobiliare invia periodicamente all'utente iscritto la lista delle case in vendita vicine ai suoi desideri: il sistema determina nel suo database di case in vendita quelle che si trovano all'interno del *cerchio* di *centro* e *raggio* definito dai dati forniti inizialmente dall'utente relativi alle caratteristiche della casa cercata (posizione, metratura, costo ed altre).

## ESERCIZI

**8.1** Dimostrare che i tre assiomi di uno spazio metrico riportati nel paragrafo 8.2 possono essere sostituiti dai seguenti due che assorbono la proprietà di simmetria:

$$M_1. \quad d(x, x) = 0 \text{ per ogni } x \in E$$

$$M_2. \quad d(x, y) \leq d(y, z) + d(z, x) \text{ per ogni } x, y, z \in E$$

**8.2** Dimostrare che le metriche definite negli esempi riportati nel paragrafo 8.2 soddisfano agli assiomi di spazio metrico.

**8.3** Sia  $X$  un insieme finito e  $E = \mathcal{P}(X)$  l'insieme delle parti di  $X$ . Dimostrare che la seguente funzione su  $E \times E$  definisce una metrica su  $E$  (l'operatore  $|\cdot|$  denota la cardinalità dell'insieme):

$$d(A, B) = |A| + |B| - 2|A \cap B|$$

**8.4** Dare una coerente definizione di *centro* e *raggio* di uno spazio metrico. Per ciascuno di questi due termini descrivere un esempio nel caso di un insieme di stringhe.

**8.5** Descrivere graficamente il *cerchio* e la *circonferenza* di raggio unitario secondo le metriche  $d_1$ ,  $d_2$ ,  $d_{p=3}$ ,  $d_\infty$  riportate nel paragrafo 8.4 riferite al piano  $\mathbb{R}^2$ .

**8.6** Determinare la distanza fra due date  $(a_1, m_1, g_1)$  e  $(a_2, m_2, g_2)$ , essendo  $(a, m, g)$  la terna che esprime anno, mese, giorno.

**8.7** Si consideri lo spazio  $\mathbb{Z}^2$  con la metrica definita da

$$d(x, y) = \text{numero min. di mosse del cavallo per andare da } x \text{ a } y$$

1. Dimostrare che si tratta di una metrica.
2. Scrivere un algoritmo per calcolare la distanza fra due punti.
3. Determinare il cerchio di raggio 2 e centro  $(0, 0)$ .
4. Analizzare la metrica indotta sul sotto spazio  $E = \{1, 2, \dots, n\}^2$ ; in particolare si analizzi il caso  $n = 3$
5. Calcolare la distanza fra  $(0, 0)$  e  $(1, 1)$  e fra  $((0, 0)$  e  $(0, 1)$ .

**8.8** Determinare i cerchi di raggio 2 e di raggio 3 in  $\mathbb{Z}^2$  con la metrica della *distanza del cavallo degli scacchi*.

**8.9** Nel gioco degli scacchi, con riferimento ad un dato pezzo ed alla situazione di scacchiera senza pezzi che impediscono il movimento, si definisce la distanza fra due caselle come il numero minimo di mosse che deve compiere il pezzo per andare da una casella di coordinate  $(i, j)$  ad un'altra casella di coordinate  $(k, l)$ . Determinare le distanze per ciascuno dei seguenti pezzi: cavallo, alfiere, torre, regina, re.

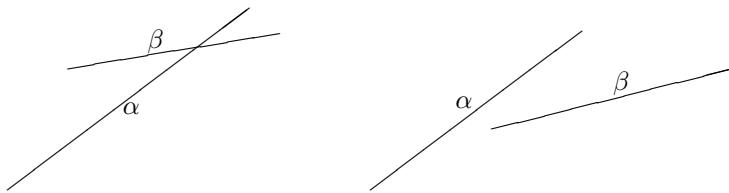
**8.10** Si consideri lo spazio  $E = \mathbb{R}^2$  dove si interpreta una coppia  $(a, b) \in E$  come intervallo  $[a, b]$  della retta reale. Si definisce

$$d(x, y) = d([a_1, b_1], [a_2, b_2]) = ||a_1 - b_1| - |a_2 - b_2||$$

Stabilire se si tratta di una metrica.

**8.11** Determinare la distanza di Hausdorff fra due intervalli  $[a_1, b_1]$  e  $[a_2, b_2]$  della retta reale.

**8.12** Evidenziare graficamente la distanza di Hausdorff fra i due segmenti  $\alpha$  e  $\beta$  riportati nella seguente figura:



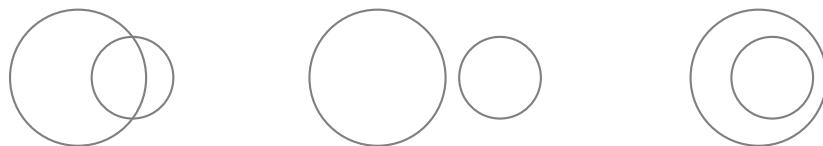
Descrivere degli algoritmi per determinare la distanza di Hausdorff fra due segmenti, nel contesto del piano punteggiato e del piano cartesiano.

**8.13** Si considerino nel piano euclideo le seguenti tre coppie di rettangoli, aventi i lati reciprocamente paralleli:



Disegnare per ciascuno dei tre casi il segmento di lunghezza pari alla *distanza* fra i rettangoli di ciascuna coppia. Determinare tali distanze in funzione delle lunghezze  $(a, b)$ ,  $(c, d)$  dei lati dei due rettangoli e della distanza  $d$  fra i loro baricentri.

**8.14** Si considerino nel piano euclideo le seguenti tre coppie di circonferenze:



Disegnare per ciascuno dei tre casi il segmento di lunghezza pari alla *distanza* fra le varie coppie di circonferenze. Determinare la distanza fra due circonferenze in funzione della lunghezza  $r_1$  ed  $r_2$  dei raggi e della distanza  $d$  fra i centri.

**8.15** Per ciascuno dei sei casi descritti nei due precedenti esercizi 8.13 e 8.14 disegnare un segmento di lunghezza pari alla distanza fra le varie coppie di figure, considerandole *piene*.

**8.16** Scrivere un algoritmo per determinare la distanza di Hausdorff fra una coppia di elementi delle seguenti tipologie:

1. punto - punto
2. punto - segmento
3. punto - retta

4. punto - circonferenza
5. segmento - segmento
6. segmento - retta
7. segmento - circonferenza
8. retta - retta
9. retta - circonferenza

**8.17** Determinare il *diametro* di un insieme di punti in uno spazio metrico.

**8.18** Un criterio spontaneo ed intuitivo per definire la distanza fra due figure  $A$  e  $B$  di uno spazio metrico consiste nel definire la seguente funzione:

$$\delta(A, B) \stackrel{\text{def}}{=} \inf_{x, y \in A} d(x, y)$$

Dimostrare che questa funzione, pur risultante di indubbio interesse e significato per molte applicazioni, non costituisce una metrica e quindi non permette di beneficiare di tutti i risultati e le proprietà validi per gli spazi metrici. Suggerimento: osservare, ad esempio, che  $(A \cap B \neq \emptyset) \Rightarrow (\delta(A, B) = 0)$ .

**8.19** Decidere se la seguente funzione sulle stringhe soddisfa agli assiomi di spazio metrico (con  $|\alpha|$  viene denotata la lunghezza della stringa  $\alpha$ ):

$$d(x, y) = |||x| - |y|||$$

**8.20** Dimostrare che la funzione distanza di edit  $d_E$  definisce effettivamente una metrica, ossia soddisfa agli assiomi di *spazio metrico*.

**8.21** Determinare nello spazio metrico  $(\{a, b, c\}^*, d_E)$  il cerchio di raggio unitario e centro la stringa **acaab**.

**8.22** Determinare la distanza di edit fra le due stringhe **a<sup>h</sup>, b<sup>k</sup>**.

**8.23** Determinare la matrice di edit delle due stringhe "libro" ed "albero". Basandosi sulla matrice di edit, determinare la (minima) sequenza di trasformazioni per trasformare una stringa nell'altra.

**8.24** Scrivere un algoritmo per determinare la distanza di edit fra due stringhe. Testare l'algoritmo codificandolo in un linguaggio di programmazione.

**8.25** Determinare la sequenza di operazioni che trasforma una stringa in un'altra secondo la distanza di edit.

**8.26** Dati due testi (documenti) costituiti da parole, definire un *indice di affinità* compreso fra 0 ed 1, che indichi la *vicinanza* fra i due documenti, valutando (opportunamente) la distanza di edit fra le parole dei due documenti. Tale indice deve valere 1 nel caso i due documenti siano uguali; nella valutazione si considerino solo le parole di almeno 4 caratteri. Scrivere un algoritmo che risolve il problema.

**8.27** Dimostrare che ammettendo l'insieme di trasformazioni  $T = \{inserimento\}$  non risulta sempre possibile trasformare una stringa  $x$  in una stringa  $y$ . Nel caso in cui sia possibile la funzione di distanza non rispetta comunque gli assiomi di spazio

metrico (non è simmetrica). In generale si ha  $d(x, y) = |y| - |x|$  oppure  $d(x, y) = +\infty$  nel caso in cui la stringa  $x$  non sia trasformabile nella stringa  $y$ . Questa funzione  $d$ , pur non rispettando gli assiomi di spazio metrico, viene comunque detta *episode distance*.

**8.28** Analizzare le considerazioni sopra riportate al paragrafo 8.1. evidenziando le idee importanti al fine di progettare un algoritmo per determinare il grado di vicinanza fra due impronte digitali.



# 9

---

## APPROSSIMARE

---

*La radice di 2 è stata una vera e propria cavia, utilizzata per mettere alla prova, descrivere e promuovere concetti nuovi.*

*Benoit Rittaud, La favolosa storia della radice quadrata di due*

C'è un punto in cui la struttura finita e discreta delle macchine si incontra con l'infinito ed il continuo di alcuni enti matematici. È questo il punto in cui la certezza dell'esistenza di alcuni enti non è sufficiente e deve cedere il passo all'effettiva costruibilità degli stessi, passando il testimone dell'indagine agli algoritmi di approssimazione.

Un problema strettamente connesso alla *determinazione* di un numero riguarda la possibilità di *esprimere* in modo finitistico il risultato, anche nel caso si tratti di un numero decimale illimitato (esprimibile come frazione o numero periodico (esempio  $2/3$ ) o di un numero irrazionale (esempio  $\sqrt{2}$ ) o trascendente (esempio  $\pi$ ). Queste due problematiche trovano soluzione analizzando il concetto di *approssimazione* nel contesto degli spazi metrici ed utilizzando alcune tecniche algoritmiche che ne derivano.

Nei processi di approssimazione viene enfatizzato il ruolo di collaborazione fra uomo e computer: l'uomo mette l'idea ed il computer la forza. Il calcolatore dà forza vitale al concetto teorico di convergenza e rende concrete e visibili alcune caratteristiche degli algoritmi, quali la velocità di convergenza.

## 9.1 Vi presento la $\sqrt{2}$

La  $\sqrt{2}$  è un famoso personaggio della Matematica, non altrettanto noto come i suoi famosi colleghi  $i$ ,  $e$ ,  $\phi$  e  $\pi$  (che hanno avuto il privilegio di avere un nome proprio). È un personaggio molto antico: è nato oltre due millenni fa presso il mondo scientifico e filosofico dell'antica Grecia, dopo un lungo e difficile travaglio; appena apparso sulla scena venne quasi rinnegato, come un figlio indesiderato. Ci vollero duemila anni affinché venisse ospitato a pieno titolo nel mondo dei numeri. È ancora oggi un individuo interessante in quanto stimola considerazioni ed approfondimenti sia dal punto di vista matematico che da quello informatico. Il matematico francese Benoit Rittaud nel capitolo introduttivo del suo libro *La favolosa storia della radice quadrata di due* dice:

*La radice quadrata di 2, che vale approssimativamente 1,414213562, è, secondo la definizione attualmente più in voga, il numero che, moltiplicato per se stesso, dà 2. È anche La radice del quadrato di dimensioni pari a 2, ovvero la lunghezza del lato di un quadrato di area 2. È questo carattere geometrico che fa di questa radice un punto di partenza, un'origine. Entrambe le definizioni potrebbero farci pensare di avere a che fare con un numero buono solamente a esprimere la soluzione di un problema di geometria per degli studenti cui si chieda di imparare che l'area  $A$  di un quadrato di lato  $a$  è data dalla formula  $A = a^2$ . In realtà, non solo i due aspetti (quello algebrico e quello geometrico) hanno numerosissime conseguenze in direzioni spesso inattese, ma la radice quadrata di 2 ammette ulteriori definizioni, e anche queste danno vita a ramificazioni che si estendono ben oltre il semplice calcolo dell'area di un quadrato. È così che i campi in cui interviene questo numero almeno quattro volte milleenario nella storia del pensiero sono di una varietà pressoché infinita. [...] Una prima caratteristica interessante della radice di 2 è che essa rappresenta una porta aperta su interi settori della matematica, sia antica che moderna: la geometria e la teoria dei numeri, ma anche la logica, l'algebra, l'aritmetica, l'analisi e, più recentemente, l'algoritmica, le strutture di dati, i numeri gadici e la dinamica simbolica. [...] Una volta stabilita l'importanza pratica della radice di 2 si pone il problema di calcolarne il valore effettivo.*

La definizione  $\sqrt{2}$  è quel numero che moltiplicato per se stesso dà 2 come risultato, apparentemente innocua, sembra autosufficiente, conclusiva e non lascia presagire alcuna problematica. Sfruttando delle proprietà che discendono direttamente dalla sua definizione, questo oggetto può essere manipolato in modo formale; ad esempio, senza neanche conoscere il valore di  $\sqrt{2}$  ma affidandosi solamente alla sua definizione, si possono eseguire dei calcoli quali:

$$\begin{aligned} \sqrt{2} \cdot \sqrt{2} &\rightarrow 2 \\ \sqrt{2} \cdot \frac{1}{\sqrt{2}} &\rightarrow 1 \\ \frac{1}{\sqrt{2}} &\rightarrow \frac{\sqrt{2}}{\sqrt{2} \cdot \sqrt{2}} \rightarrow \frac{\sqrt{2}}{2} \rightarrow \frac{1}{2}\sqrt{2} \end{aligned}$$

## 9.2 Alla ricerca della $\sqrt{2}$

I problemi iniziano a sorgere quando si considera l'obiettivo *Determinare la  $\sqrt{2}$* . Ovviamente  $\sqrt{2}$  non è la *soluzione* del problema, è semplicemente una notazione, un *simbolo* che *denota* quel numero che moltiplicato per se stesso da 2, in modo analogo a come possiamo usare  $3/4$  per denotare quel numero che otteniamo eseguendo l'operazione *3 diviso 4*. Determinare un oggetto significa esibirlo, in un contesto ben precisato, in un formato ed in una notazione specifica del contesto considerato. Questo quesito richiede implicitamente di definire l'insieme nel quale la  $\sqrt{2}$  viene cercata. I greci pensavano che tutti i numeri fossero esprimibili come numeri razionali, come rapporto di due numeri naturali. La  $\sqrt{2}$  non è invece esprimibile come numero razionale.

Con le premesse date, *il problema della  $\sqrt{2}$*  solleva le seguenti questioni (che saranno affrontate e risolte più avanti):

1. dimostrare che  $\sqrt{2}$  non è esprimibile mediante un numero razionale
2. determinare in quale insieme si trova  $\sqrt{2}$
3. determinare una frazione che *approssima bene* il valore  $\sqrt{2}$
4. determinare e descrivere un procedimento che, basandosi solo sulle quattro operazioni aritmetiche fondamentali sulle frazioni, fornisca le varie cifre decimali di  $\sqrt{2}$
5. dato un segmento di lunghezza unitaria nel piano, costruire con riga e compasso un segmento di lunghezza  $\sqrt{2}$

È facile mostrare che una tale  $\sqrt{2}$  non è esprimibile come frazione: un risultato che risale alla scuola pitagorica ed era ben noto a Euclide. Una dimostrazione del risultato pitagorico, considerata dal matematico e teorico dei numeri Paul Erdős come una delle più belle di tutta la matematica, è riportata a seguire.

*Dimostrazione.* Supponiamo per assurdo che esistano due numeri naturali  $p$  e  $q$  tali che

$$2 = \left(\frac{p}{q}\right)^2 = \frac{p^2}{q^2}$$

Possiamo supporre che la frazione sia ridotta, ovvero che  $p$  e  $q$  siano primi fra di loro. Dalla precedente relazione si ottiene

$$p^2 = 2 \cdot q^2$$

Ne segue che 2 divide  $p^2$ , e quindi  $p$  è pari. Quindi  $p = 2 \cdot k$  per qualche  $k \in \mathbb{N}$ . Di conseguenza otteniamo:

$$2 \cdot k^2 = q^2$$

Allora anche  $q$  è pari, in contraddizione con il fatto che  $p$  e  $q$  siano coprimi. Dunque deve essere falsa l'ipotesi iniziale, cioè  $\sqrt{2}$  non può essere razionale.  $\square$

Dimostrato che  $\sqrt{2}$  non può essere un numero razionale, si tratta ora di accertarsi che tale numero esiste fra i numeri reali. Ciò è garantito dal seguente

**TEOREMA 6 (Esistenza di  $\sqrt{2}$ ).** Nell'insieme dei numeri reali  $\mathbb{R}$  esiste la radice quadrata di 2, cioè un numero positivo il cui quadrato è esattamente 2.

La dimostrazione di questo teorema, che qui non viene riportata, può essere basata usando soltanto le quattro operazioni e i principi fondamentali dell'ordinamento di  $\mathbb{R}$ .

La  $\sqrt{2}$  ha anche una semplice definizione geometrica: è la lunghezza della diagonale del quadrato di lato unitario. Questa lineare definizione ed il corrispondente metodo di costruzione con riga e compasso fa intuire che questo numero si trova più a suo agio in ambito geometrico.

I greci traducevano i numeri in ambito geometrico, mediante segmenti la cui lunghezza rappresentava il numero stesso; in particolare determinare un numero significava costruire un segmento la cui lunghezza era il numero. In tale contesto, determinare  $\sqrt{2}$  significava costruire con riga e compasso un segmento di lunghezza  $\sqrt{2}$  a partire da un segmento di lunghezza unitaria. La figura 9.1 descrive una costruzione.

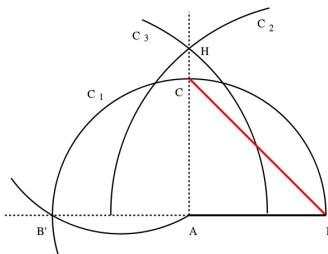


Figura 9.1: La costruzione di un segmento di lunghezza  $\sqrt{2}$  a partire dal segmento di lunghezza unitaria  $AB$ .

### 9.3 Calcolo della $\sqrt{2}$

1, 4142135623 73095 04880 16887 24209 69807 85696 è il valore di  $\sqrt{2}$  approssimato a quaranta cifre decimali. Un'approssimazione di questo numero è riportata in un antico testo matematico indiano, il Sulbasutras, scritto nel periodo 800-600 a.C., che afferma: *Aumenta la lunghezza [del lato] della sua terza parte, poi aggiungi la sua dodicesima parte, infine sottra un trentaquattresimo della sua dodicesima parte.* Tradotta numericamente, questa approssimazione diventa.<sup>1</sup>

$$1 + \frac{1}{3} + \frac{1}{12} - \frac{1}{12 \cdot 34} = \frac{577}{408} \approx 1.4142156862745$$

I babilonesi, usando un sistema di numerazione in base 60 (vedi figura 9.2), diedero una approssimazione di  $\sqrt{2}$  leggermente migliore, tramite l'espressione

$$1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3} = \frac{30547}{21600} \approx 1,4142129629629$$

<sup>1</sup>Qui e nel seguito vengono indicate in grassetto le cifre decimali corrette.

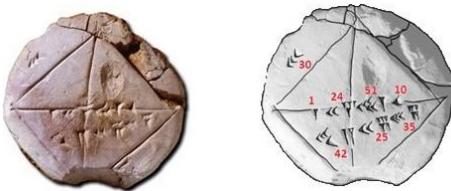


Figura 9.2: Su una tavoletta d'argilla, attualmente conservata presso la Yale University (collezione Yale, YBC 7289), ritrovata in una città della Mesopotamia, risalente al periodo della dinastia di Hammurabi (1900–1600 a.C.), è riportato un valore approssimato della  $\sqrt{2}$ . Sulla tavoletta sono tracciate le diagonali di un quadrato di lato 30; sulla diagonale sono scritti due gruppi di numeri: 1, 24, 51, 10 e 42, 25, 35 che, in base 60, corrispondono a delle approssimazioni di  $\sqrt{2}$  e  $30\sqrt{2}$ . Da queste espressioni si è dedotto che i Babilonesi conoscevano, oltre ad una buona approssimazione di  $\sqrt{2}$ , il teorema di Pitagora, circa 1300 anni prima dello scopritore ufficiale.

Con gli sviluppi ed i progressi dell'Analisi, a metà del XVIII sec. Eulero, utilizzando delle rappresentazioni tramite somme e prodotti infiniti delle funzioni *seno* e *coseno*, ricavò le seguenti formule<sup>2</sup>

$$\sqrt{2} = \frac{7}{5} \left( 1 + \sum_{k=1}^{\infty} \frac{(2k-1)!!}{100^k \cdot k!} \right)$$

$$\sqrt{2} = \prod_{k=0}^{\infty} \frac{(4k+2)^2}{(4k+1)(4k+3)} = \left( \frac{2 \cdot 2}{1 \cdot 3} \right) \left( \frac{6 \cdot 6}{5 \cdot 7} \right) \left( \frac{10 \cdot 10}{9 \cdot 11} \right) \left( \frac{14 \cdot 14}{13 \cdot 15} \right) \dots$$

$$\sqrt{2} = \prod_{k=0}^{\infty} \left( 1 + \frac{1}{4k+1} \right) \left( 1 - \frac{1}{4k+3} \right) = \left( 1 + \frac{1}{1} \right) \left( 1 - \frac{1}{3} \right) \left( 1 + \frac{1}{5} \right) \left( 1 - \frac{1}{7} \right) \dots$$

## 9.4 Calcolo della $\sqrt{a}$

Nel corso della storia sono stati ideati molti algoritmi per il calcolo delle cifre di  $\sqrt{2}$ . Alcuni di questi si prestano ad essere generalizzati per il calcolo della radice quadrata di un generico numero.

### Metodo di bisezione

Un elementare precedimento di calcolo approssimato della  $\sqrt{2}$  può essere descritto con le seguenti parole:

<sup>2</sup>Con  $n!!$  si denota il *semifattoriale* del numero naturale  $n$  definito come segue:  $n!! = 2 \cdot 4 \cdot 6 \cdots n$  se  $n$  è pari,  $n!! = 1 \cdot 3 \cdot 5 \cdots n$  se  $n$  è dispari.

Per determinare la radice quadrata di un numero naturale  $a$  determiniamo inizialmente due valori  $x_1$  ed  $x_2$  che siano un'approssimazione di  $\sqrt{a}$  e tali che sia  $x_1 < \sqrt{a} < x_2$ . Successivamente calcoliamo la media  $t = \frac{x_1+x_2}{2}$  e verifichiamo se  $t$  è minore o maggiore di  $\sqrt{a}$ ; questa verifica può essere fatta confrontando  $t^2$  con  $a$  (visto che  $\sqrt{a}$  non è noto). Se  $t$  è maggiore di  $\sqrt{a}$ , allora  $t$  sostituirà il valore in eccesso; se è minore sostituirà l'altro valore. Si ripete questo procedimento fino a che si ottiene un valore approssimato soddisfacente.

Prima di procedere oltre si consiglia di leggere con attenzione il procedimento e di applicarlo al calcolo di un valore approssimato di  $\sqrt{7}$  utilizzando una semplice calcolatrice dotata delle 4 operazioni aritmetiche di base.

La precedente formulazione del procedimento di calcolo della radice quadrata risulta soddisfacente per un operatore umano, ma inadeguata per essere tradotta mediante un linguaggio di programmazione al fine di essere sottomesa ad un calcolatore digitale, in quanto molti aspetti rimangono indefiniti (scelta dei valori iniziali di approssimazione, numero di volte da ripetere il ciclo del calcolo). Nell'algoritmo 1 che segue questi aspetti vengono definiti; in particolare viene adottato il criterio di terminare il processo di calcolo quando si sono ottenuti due valori approssimati molto vicini fra loro.

---

### **Algoritmo 1 - Radice quadrata di un numero naturale**

---

**Input:** numero naturale  $a$

**Output:** radice quadrata approssimata di  $a$

```

1: $\epsilon \leftarrow 10^{-5}$ \triangleright margine di approssimazione
2: $x_1 \leftarrow (1 + a)/2$
3: $x_2 \leftarrow a$
4: while $(x_2 - x_1) > \epsilon$ do
5: $t \leftarrow (x_1 + x_2)/2$
6: if $(t * t) < a$ then
7: $x_1 \leftarrow t$
8: else
9: $x_2 \leftarrow t$
10: end if
11: end while
12: return $(x_1 + x_2)/2$
```

---

### **Metodo babilonese**

Uno dei metodi più usati per il calcolo della radice quadrata di un numero  $a$  deriva da una generalizzazione del metodo ideato dai babilonesi nel XVIII secolo a.C. per il calcolo della  $\sqrt{2}$ ; il metodo è stato più recentemente analizzato da Newton alla fine del 1600.

L'idea è la seguente. Il problema di calcolare la  $\sqrt{2}$  coincide con il problema di trovare la soluzione dell'equazione  $x^2 = a$ . Tale equazione può essere scritta

nella forma  $x = a/x$ . Se  $x_0$  non è la soluzione esatta di questa equazione ma una sua approssimazione, allora i due valori  $x_0$  e  $a/x_0$  saranno delle approssimazioni di  $\sqrt{2}$ , una per difetto e l'altra per eccesso. La soluzione effettiva sarà quindi compresa fra questi due valori. La media  $x$  fra questi due valori vale

$$x_1 \leftarrow \frac{1}{2} \left( x_0 + \frac{a}{x_0} \right)$$

e costituirà quindi un'approssimazione migliore. Il procedimento può essere sintetizzato mediante il seguente schema iterativo:

$$\begin{aligned} x_0 &\leftarrow p \\ x_k &\leftarrow \frac{1}{2} \left( x_{k-1} + \frac{a}{x_{k-1}} \right) \end{aligned}$$

dove  $p$  è una generica approssimazione iniziale, ad esempio 1. Si può dimostrare che tale successione converge a  $\sqrt{2}$ . Notiamo che, se  $x_0$  è un numero razionale, tutti i valori della successione  $(x_0, x_1, x_2, \dots, x_k, \dots)$  sono numeri razionali. Si può dimostrare che, indipendentemente dal valore di partenza  $p$ , la successione  $(x_k)$  converge a  $\sqrt{a}$ . Benché il valore di partenza  $x_0$  non influenzi sensibilmente la velocità di convergenza della successione, un valore consigliabile è:

$$x_0 \leftarrow \frac{1+a}{2}$$

Sono stati fatti studi particolari per la determinazione di buoni valori di partenza  $x_0$  (in funzione di  $a$ ); in particolare sono state individuate delle espressioni polinomiali per ottenere valori di partenza ottimi per  $x_0$ . Il procedimento può essere arrestato quando il risultato si è sufficientemente stabilizzato oppure adottando il criterio di terminazione eseguendo il ciclo un prefissato numero  $n$  di volte, come indicato nell'algoritmo 2.

---

**Algoritmo 2 - Radice quadrata di un numero naturale**


---

**Input:** numero naturale  $a$ , numero  $n$  di iterazioni

**Output:** radice quadrata approssimata di  $a$

```

1: $n \leftarrow 5$ \triangleright numero di iterazioni
2: $x \leftarrow (1+a)/2$
3: for n times
4: $x \leftarrow \frac{1}{2} \left(x + \frac{a}{x} \right)$
5: end for
6: return x

```

---

## 9.5 Frazioni continue

Le *frazioni continue* sono classici oggetti della teoria dei numeri; si ritengono tradizionalmente ideate da Euclide in connessione al suo ben noto algoritmo per il calcolo del massimo comune divisore fra due numeri naturali; con il formalismo moderno vennero utilizzate la prima volta nel 1500 ad opera di due matematici italiani, Bombelli (1530) che espresse lo sviluppo di  $\sqrt{13}$  mediante una frazione continua infinita e Cataldi (1548-1626) che riuscì ad esprimere  $\sqrt{18}$  mediante una frazione continua infinita; vennero studiate sistematicamente solo nel 1700 da Eulero ed in seguito da altri grandi matematici come Jacobi, Lagrange, Galois e Gauss. In tempi recenti sono state utilizzate all'interno di algoritmi di calcolo per le approssimazioni di numeri reali e in relazione alla teoria del caos.

Le *frazioni continue semplici* sono espressioni della forma

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\dots + \cfrac{1}{a_n}}}}}$$

dove  $a_0$  è un numero intero e gli altri  $a_i$ ,  $i \geq 1$ , sono interi positivi. Gli elementi  $a_0, a_1, a_2, \dots, a_n$  si dicono *termini* della frazione continua o *quozienti parziali*. Una frazione continua viene spesso denotata mediante la sequenza dei suoi termini, ossia con

$$[a_0, a_1, a_2, \dots, a_n]$$

La notazione appena descritta è univoca se si impone il vincolo  $a_n \geq 2$ . Questa notazione suggerisce direttamente la modalità di rappresentazione di una frazione continua mediante una sequenza semplice di numeri interi.

Una frazione continua si dice *finita* quando il suo sviluppo al denominatore si ferma, ed essa rappresenta un numero razionale. Si possono considerare anche frazioni continue *infinite*, cioè il cui sviluppo al denominatore non termina mai. Frazioni continue infinite rappresentano numeri irrazionali.

Una frazione continua finita si può riscrivere facilmente come un rapporto tra due numeri. Viceversa, qualunque numero razionale si può scrivere come frazione continua finita. Il procedimento, basato su un'applicazione dell'algoritmo di Euclide per il calcolo del massimo comune divisore, è descritto nel seguente caso particolare:

$$\frac{71}{25} \rightarrow 2 + \frac{21}{25} \rightarrow 2 + \frac{1}{\frac{25}{21}} \rightarrow 2 + \frac{1}{1 + \frac{4}{21}} \rightarrow 2 + \frac{1}{1 + \frac{1}{\frac{21}{4}}} \rightarrow 2 + \frac{1}{1 + \frac{1}{5 + \frac{1}{4}}}$$

Pertanto la frazione  $\frac{71}{25}$  è rappresentata in frazione continua dalla sequenza  $[2, 1, 5, 4]$ .

L'interesse computazionale delle frazioni continue è dovuto al fatto che si possono considerare delle scritture infinite quali

$$[1, 1, 1, 1, 1, \dots]$$

$$[1, 2, 1, 2, 1, 2, \dots]$$

$$[3, 2, 3, 4, 1, 3, 4, 1, 3, 4, 1, \dots]$$

Queste scritture manifestano una evidente struttura ripetitiva e si prestano ad essere espresse con notazione a periodo, come segue:

$$[\bar{1}]$$

$$[\bar{1, 2}]$$

$$[3, 2, \bar{3, 4, 1}]$$

Evidenziando la parte non periodica e la parte periodica, queste espressioni possono essere rappresentate mediante le seguenti sequenze:

$$[[], [1]]$$

$$[[], [1, 2]]$$

$$[[3, 2], [3, 4, 1]]$$

$$[[2, 5, 1], [7]]$$

Questa creatività (o semplice meccanica generalizzazione) di scrittura ci porta a varcare il limite dei numeri razionali e ci conduce nel mondo dei numeri reali: si può dimostrarre che frazioni continue infinite convergono a numeri reali. Molti interessanti ed utili numeri reali vengono calcolati proprio facendo ricorso ad algoritmi che operano su frazioni continue.

Il procedimento di trasformazione di una frazione in frazione continua non può essere utilizzato in generale per ottenere una frazione continua a partire da un numero irrazionale. Esistono però dei metodi alternativi. Nel caso particolare di  $\sqrt{2}$  si può scrivere

$$\sqrt{2} = 1 + \sqrt{2} - 1 = 1 + \frac{(\sqrt{2} - 1)(\sqrt{2} + 1)}{(\sqrt{2} + 1)} = 1 + \frac{1}{1 + \sqrt{2}}$$

da cui

$$\sqrt{2} = 1 + \frac{1}{1 + \sqrt{2}}$$

Si delinea così lo sviluppo di  $\sqrt{2}$  in frazione continua:

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{\ddots}}}}$$

In forma sintetica:

$$\sqrt{2} = [1, \bar{2}]$$

Una scrittura come questa non esprime il fatto che conosciamo il valore di  $\sqrt{2}$  o che lo conosciamo meglio; esprime solamente una proprietà che permette di calcolare numericamente il valore di  $\sqrt{2}$  con una approssimazione desiderata. Per sfruttare computazionalmente l'espressione di  $\sqrt{2}$  in frazione continua è necessario tradurre l'espressione in un algoritmo. A questo scopo aggiungiamo 1 ad entrambi i lati ed analizziamo la struttura dell'espressione per individuare il passo ricorsivo:

$$1 + \sqrt{2} = 2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{\ddots}}}}$$

Analizzando la struttura dell'espressione che esprime la frazione continua si evidenzia il seguente processo che genera una successione di numeri razionali che converge alla  $\sqrt{2}$ . Questo processo può essere tradotto nell'algoritmo 3.

---

### Algoritmo 3 - Radice quadrata di 2 con frazione continua

---

**Output:** radice quadrata approssimata di 2

```

1: x \leftarrow 2
2: for n times
3: x \leftarrow $2 + \frac{1}{x}$
4: end for
5: return x - 1
```

---

Nell'algoritmo 3 il numero *n* di iterazioni può essere un argomento di input oppure una costante definita all'interno dell'algoritmo.

## 9.6 Velocità di convergenza

Come atteso, in un algoritmo approssimato, maggiore è il numero di iterazioni, migliore sarà la precisione del risultato. Da un semplice confronto dei precedenti algoritmi per il calcolo della  $\sqrt{2}$  si nota una diversa velocità di convergenza al valore corretto. La tabella 9.1 riporta la lista di alcuni termini della successione.

| $n$ | Algoritmo 1    | Algoritmo 2               | Algoritmo 3               |
|-----|----------------|---------------------------|---------------------------|
| 0   | <b>1</b>       | 1.5                       | <b>1</b>                  |
| 1   | <b>1.5</b>     | <b>1.4166666666666666</b> | <b>1.5</b>                |
| 2   | <b>1.25</b>    | <b>1.414215686274509</b>  | <b>1.4</b>                |
| 3   | <b>1.375</b>   | <b>1.414213562374689</b>  | <b>1.4166666666666666</b> |
| 4   | <b>1.4375</b>  | <b>1.414213562373095</b>  | <b>1.413793103448276</b>  |
| 5   | <b>1.40625</b> | <b>1.414213562373095</b>  | <b>1.414285714285714</b>  |

Tabella 9.1: Valori della successione per il calcolo di  $\sqrt{2}$  mediante il metodo di bisezione, il metodo babilonese ed il metodo delle frazioni continue; in grassetto sono evidenziate le cifre corrette.

Da una superficiale analisi della tabella 9.1 emerge che nel caso dell'algoritmo 1 di bisezione la convergenza è molto lenta mentre è particolarmente veloce per l'algoritmo 1 babilonese. Questo fatto trova precisa definizione nel concetto di velocità di convergenza descritto nella seguente definizione.

**DEFINIZIONE 8.** Si dice che una successione  $(x_k)$ , generata da un procedimento numerico, converge ad un limite  $\alpha$  con *ordine*  $p$  se esiste un indice  $\bar{k}$  ed una costante positiva  $C$  tale che

$$\frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} \leq C \quad \text{per ogni } k > \bar{k}$$

Se  $p = 1$  la velocità di convergenza è detta *lineare*, se  $p = 2$  *quadratica*. Nel caso di convergenza con velocità quadratica il numero di decimali corretti raddoppia ad ogni iterazione. È questo il caso dell'algoritmo babilonese per il calcolo della radice quadrata.

## 9.7 Calcoli infiniti

Oltre che essere applicati per il calcolo approssimato di particolari valori numerici, i procedimenti di approssimazione vengono applicati per valutare il valore al quale converge un calcolo che coinvolge un numero infinito di operandi, come descritto nel problema che segue.

**Problema 9.7.1** Calcolare la seguente somma infinita

$$\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}}$$

**Soluzione.** Questo problema può essere affrontato con diversi approcci. Un approccio, che chiamiamo *matematico*, fa ricorso a tecniche matematiche tipiche dell'Analisi Matematica. I passi sono i seguenti:

1. definizione ricorsiva della successione delle somme parziali:

$$\begin{aligned} s_0 &= \sqrt{1} \\ s_n &= \sqrt{1 + s_{n-1}} \end{aligned}$$

2. dimostrazione dell'esistenza del limite (finito)

$$\lim_n s_n$$

3. calcolo del valore  $s$  del limite

Un approccio, che possiamo chiamare *informatico*, si innesta sulle considerazioni appena svolte sopra, sostituendo il punto 3 con un algoritmo affettivo per il calcolo del limite; basandosi sulla conoscenza che la successione  $\{s_n\}$  è convergente, l'algoritmo 4 calcola un valore approssimato interrompendo il ciclo di calcolo quando saranno stati calcolati due valori distanti meno di una quantità  $\varepsilon = 10^{-k}$ , il che equivale ad aver determinato un valore approssimato con  $k$  cifre decimali esatte.

---

**Algoritmo 4** - Calcolo di  $\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}}$

---

```

1: $k \leftarrow 5$
2: $\varepsilon \leftarrow 10^{-k}$
3: $r \leftarrow 0$
4: $s \leftarrow \sqrt{1}$
5: while $|s - r| > \varepsilon$ do
6: $r \leftarrow s$
7: $s \leftarrow \sqrt{1 + s}$
8: end while
9: return s

```

---

Questo problema, nonostante l'apparenza, può essere risolto con un approccio *alternativo* che utilizza un arsenale matematico molto ridotto, quello fornito dal biennio di un istituto superiore. Indicando con  $x$  la quantità incognita che rappresenta la sommatoria sopra ed elevando al quadrato entrambi i

membri dell'equazione si ottiene la seguente equazione:

$$x^2 = 1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}$$

Notando che al secondo membro compare ancora la quantità originale, denotata con  $x$ , si ricava l'equazione

$$x^2 = 1 + x$$

che fornisce la soluzione (scartando quella negativa)

$$x = \frac{1 + \sqrt{5}}{2}$$

Osserviamo che difficilmente si riesce a riconoscere questo numero fra le cifre decimali del numero ottenuto mediante l'algoritmo 4.

## 9.8 Convergenza negli spazi metrici

Disporre di un algoritmo per il calcolo approssimato di un numero permette di calcolare il numero con una approssimazione desiderata ma lascia aperti molti problemi di ordine pratico quali: La successione dei valori calcolati converge? Converge al numero cercato? Converge sempre? Con quale velocità converge? Quante approssimazioni servono per avere una prefissata precisione? Gli errori di approssimazione del calcolo influenzano la convergenza? Tutte queste questioni trovano una adeguata risposta se il problema viene sufficiente generalizzato in un adeguato contesto formale. Gli spazi metrici sono il naturale ambiente per studiare la convergenza di successioni che emergono in molti procedimenti iterativi tipici dell'analisi numerica. Nel contesto degli spazi metrici, con riferimento a molte problematiche di tipo algoritmico (che saranno affrontate nel prossimo paragrafo) acquista particolare importanza il concetto di *convergenza* di una successione, oggetto della seguente definizione.

**DEFINIZIONE 9.** Sia  $(E, d)$  uno spazio metrico,  $(x_k)_{k \in \mathbb{N}}$  una successione di  $E$ . Si dice che la successione considerata *converge* all'elemento  $\bar{x} \in E$  e si scrive

$$\lim_n x_n = \bar{x}$$

oppure

$$x_n \rightarrow \bar{x}$$

se per ogni  $\epsilon > 0$  esiste  $k \in \mathbb{N}$  tale che  $d(x_n, \bar{x}) < \epsilon$  per ogni  $n > k$ .

Si può notare che la nozione di convergenza in uno spazio metrico può essere fondata sulla nozione di convergenza topologica in  $\mathbb{R}$  definendo che

$$x_n \rightarrow \bar{x} \text{ in } E \quad \text{se e solo se} \quad d(x_n, \bar{x}) \rightarrow 0 \text{ in } \mathbb{R}$$

Intuitivamente,  $x_n \rightarrow \bar{x}$  se al crescere di  $n$  l'elemento  $x_n$  si avvicina indefinitamente all'elemento  $\bar{x}$ .

La seguente definizione stabilisce una importante caratterizzazione delle successioni in uno spazio metrico.

**DEFINIZIONE 10.** In uno spazio metrico  $(E, d)$  una successione  $(x_n)_{n \in \mathbb{N}}$  dicesi *successione di Cauchy* se per ogni  $\epsilon > 0$  esiste  $k \in \mathbb{N}$  tale che  $d(x_m, x_n) < \epsilon$  per ogni  $m, n > k$ .

Si noti che la nozione di successione di Cauchy non fa alcun riferimento all'elemento limite.

È evidente che ogni successione di Cauchy è convergente. Non è invece sempre vero il viceversa; ad esempio ci sono delle successioni di Cauchy in  $\mathbb{Q}$  che non convergono ad alcun valore di  $\mathbb{Q}$  (successioni che convergono ad un numero irrazionale).

Si può prendere spunto dalla precedente osservazione per stabilire un'importantissima caratteristica degli spazi metrici, oggetto della seguente definizione.

**DEFINIZIONE 11.** Uno spazio metrico  $E$  dicesi *completo* se ogni successione di Cauchy converge ad un elemento di  $E$ .

## 9.9 Procedimenti iterativi indefiniti

Una delle applicazioni più significative dell'Informatica riguarda i procedimenti iterativi della Matematica. Tali procedimenti vengono applicati per la ricerca di un elemento  $x$  appartenente ad un dato spazio di ricerca  $A$  e soddisfacente ad una data condizione. Spesso l'elemento  $x$  cercato è un numero reale ma può essere anche un punto (di una retta o del piano), una sequenza di numeri, una matrice o, più in generale, un elemento di uno spazio metrico.

La ricerca viene eseguita generando una successione  $(x_k) = (x_0, x_1, \dots, x_k, \dots)$ . Se, sotto opportune ipotesi, la successione  $(x_k)$  converge ad un valore  $\bar{x}$ , tale valore viene assunto come risultato della ricerca. Affinché abbia senso parlare di convergenza, è necessario che lo spazio  $A$  sia dotato di una metrica. Il metodo algoritmico si dice *convergente*, secondo una data metrica  $d$ , al valore  $\bar{x}$  se

$$\lim_{k \rightarrow \infty} d(x_k, \bar{x}) = 0$$

ossia se per ogni  $\varepsilon \geq 0$ , esiste  $n : (k > n) \Rightarrow d(x_k, \bar{x}) < \varepsilon$ .

Un *passo* del procedimento può essere visto come una trasformazione

$$\Phi : x_k \mapsto x_{k+1}$$

In generale, un algoritmo iterativo si presenta nella seguente forma:

---

**Algoritmo 5 - Algoritmo iterativo**


---

- 1:  $x_k \leftarrow$  valore iniziale  $x_0$
  - 2: **while**  $x_k$  non è una sufficiente approssimazione **do**
  - 3:    determina un altro  $x_{k+1} \leftarrow \Phi(x_k)$
  - 4: **end while**
  - 5: **return**  $x_k$
- 

Notiamo che i *parametri* caratterizzanti un algoritmo iterativo sono i seguenti:

- valore di partenza  $x_0$  : può essere considerato come un dato di input oppure un dato intrinseco al procedimento (ricavabile, eventualmente, dagli altri dati di input).
- funzione di iterazione  $\Phi$  : è caratteristica del metodo algoritmico adottato
- criterio di arresto del processo iterativo

## 9.10 Criteri di arresto

Consideriamo un generico procedimento iterativo per risolvere un dato problema. Indicando con  $\bar{x}$  un risultato del problema, è chiaro che, secondo la metrica adottata, deve risultare

$$\lim_{k \rightarrow \infty} x_k = \bar{x}$$

Anche nell'ipotesi di convergenza di un algoritmo, rimane un fondamentale problema che riguarda la scelta del criterio (stopping-test) da adottare nell'arresto del processo di elaborazione. In generale, il calcolo della successione  $(x_k)$  viene iterato fino a quando ci si è sufficientemente *avvicinati* (secondo la vigente metrica  $d$ ) al valore  $\bar{x}$ , ossia

$$d(x_k, \bar{x}) \leq \varepsilon$$

con  $\varepsilon$  sufficientemente piccolo. Questo criterio di arresto ha il difetto che il valore  $\bar{x}$  non è noto a priori e, pertanto, risulta impraticabile. Un criterio alternativo, che si dimostra efficace nella maggioranza dei casi, consiste nell'arrestare il processo iterativo quando la soluzione si è sufficientemente *stabilizzata*, ossia quando due valori consecutivi della successione  $(x_k)$  sono sufficientemente vicini, ossia

$$d(x_k, x_{k-1}) \leq \varepsilon$$

Questo ultimo criterio si traduce nel seguente algoritmo:

---

**Algoritmo 6** - Metodo iterativo stazionario lineare di primo grado

---

```

1: $x_h \leftarrow x_0$
2: $x_k \leftarrow \Phi(x_h)$
3: while $d(x_h, x_k) \geq \varepsilon$ do
4: $x_h \leftarrow x_k$
5: $x_k \leftarrow \Phi(x_h)$
6: end while
7: return x_k

```

---

Un altro importante accorgimento riguarda la scelta della funzione di trasformazione  $\Phi$  per avere una veloce convergenza. Inoltre, utilizzando i numeri in rappresentazione finita (come avviene nei calcolatori) servono delle precauzioni che garantiscano la convergenza del metodo.

### 9.11 L'errore nei processi di calcolo

Il concetto di *errore* ha diversi significati: può avere una valenza morale quando denota una deviazione da una condotta ritenuta corretta; in Matematica può consistere in una inconsapevole maldestra applicazione di una regola di calcolo, di una formula o di un procedimento; nel seguito utilizzeremo tale termine per denotare una conscia deviazione da un risultato considerato corretto; ad esempio, la frase *La frazione  $\frac{355}{113}$  rappresenta il rapporto fra la lunghezza di una circonferenza e del suo raggio con un errore inferiore a 0.01*.

Si parla di errore anche quando, in un processo di calcolo infinito convergente si arresta il processo ad una data iterazione. Nei processi di calcolo automatico altre situazioni di errore insorgono dal fatto che i numeri che si ottengono nei calcoli sono approssimati a causa della loro rappresentazione finita in memoria.

In Matematica è sentita l'esigenza di *misurare* l'errore; questa misura dell'errore viene realizzata dotando l'ambiente di una metrica  $d$  ed esprimendo l'errore  $E$  come la distanza fra il valore effettivo  $v_e$  ed un valore approssimato  $v_a$ :

$$E = d(v_e, v_a)$$

Molto spesso l'ambiente di riferimento è costituito da un insieme numerico, nel qual caso la definizione precedente si esprime semplicemente con

$$E = |v_e - v_a|$$

## 9.12 Computabilità dei numeri

Un apparente limite di cui sembrano soffrire gli elaboratori consiste nella difficoltà di gestire in modo numerico (e non solamente simbolico) i numeri reali irrazionali. Ciò dipende dal fatto che la rappresentazione dei numeri avviene in modo finito mentre i numeri irrazionali richiedono un'espansione numerica decimale non finita. In questi casi, come avviene in molti algoritmi numerici, si approssima l'*infinito attuale* con un *infinito potenziale*, secondo l'idea descritta nella seguente definizione.

**DEFINIZIONE 12.** Un numero reale  $x$  si dice *computabile* se esiste un algoritmo  $A : \mathbb{N} \rightarrow \{0, 1, \dots, 9\}$  tale che  $A(n)$  sia l' $n$ -esima cifra decimale di  $x$ , assumendo che  $A(0) = \lfloor x \rfloor$ ; in altri termini deve essere calcolabile (mediante un procedimento effettivo) la seguente funzione:

$$f(n) = \begin{cases} \lfloor x \rfloor & \text{se } n = 0 \\ n\text{-esima cifra decimale di } x & \text{se } n > 0 \end{cases}$$

Equivalentemente, la computabilità di un numero reale  $x$  può essere basata sulla calcolabilità della seguente funzione:

$$g(n) = \begin{cases} \lfloor x \rfloor & \text{se } n = 0 \\ \text{numero formato dalle prime } n \text{ cifre decimali di } x & \text{se } n > 0 \end{cases}$$

Risulta abbastanza evidente e facile da dimostrare che tutti i numeri razionali sono computabili. La computabilità effettiva dei numeri reali con espansione decimale illimitata e non periodica richiede invece che venga fornito un procedimento algoritmico che genera una successione convergente. Esistono però numeri irrazionali non computabili.

**Esempio 9.12.1 -** Il numero  $\pi$  è computabile. Esistono infatti molte formule che, mediante sommatorie infinite, forniscono approssimazioni sempre più raffinate alle cifre di  $\pi$ . Una di queste è la seguente:

$$\pi = 4 \sum_{k=1}^{\infty} (-1)^{k+1} \frac{1}{2k-1} = 4 \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right)$$

Un'altra, molto più efficace dal punto di vista computazionale, fu scoperta verso il 1910 dal matematico indiano Ramanujan:

$$\pi = \frac{9801}{\sqrt{8}} \left( \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}} \right)^{-1}$$

Questa formula fornisce 8 nuovi decimali esatti ad ogni termine della serie. Formule più complesse e più efficaci furono scoperte alla fine del secolo scorso.

## ESERCIZI

**9.1** Analizzare le similitudini e le differenze di significato fra le seguenti scritture:

$$\frac{3}{4} \quad 3/4 \quad 0.75 \quad \frac{1+2}{2*2}$$

**9.2** Dimostrare che tutti i numeri razionali sono computabili.

**9.3** Dimostrare che esistono numeri reali non computabili. Suggerimento: utilizzare un procedimento diagonale analogo a quello utilizzato per dimostrare che l'insieme dei numeri reali non è numerabile.

**9.4** Dimostrare che, per ogni  $x \in \mathbb{R}_+$ ,  $\sqrt{x}$  è computabile.

**9.5** Scrivere un algoritmo per il calcolo di  $\sqrt{2}$  basandosi sul metodo babilonese descritto nel paragrafo 9.3.

**9.6** Scrivere degli algoritmi per il calcolo di  $\sqrt{2}$  basati sulle rappresentazioni mediante prodotti descritti alla fine del paragrafo 9.3.

**9.7** Calcolare la radice  $n$ -esima di un dato numero positivo  $a$ , basandosi sul seguente schema iterativo:

$$\begin{aligned} x_0 &= a \\ x_k &= \frac{1}{n} \left[ (n-1)x_{k-1} + \frac{a}{(x_{k-1})^{n-1}} \right] \end{aligned}$$

La successione  $(x_0, x_1, \dots, x_k, \dots)$  converge al valore  $\sqrt[n]{a}$ .

**9.8** Sia  $P(n, l)$  un poligono regolare di  $n$  lati di lunghezza  $l$ . Dimostrare che i seguenti limiti tendono ad una circonferenza:

$$\lim_{n \rightarrow \infty} P(l, n)$$

$$\lim_{n \rightarrow \infty} P(l/n, n)$$

Determinare il raggio di tali circonference.

**9.9** Calcolare un valore approssimato con 3 cifre decimali corrette a cui tende il rapporto fra due termini consecutivi della successione dei numeri di Fibonacci definita come segue:

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 1 \\ a_n &= a_{n-2} + a_{n-1} \quad \text{per } n \geq 2 \end{aligned}$$

**9.10** Dati due numeri qualsiasi  $a, b$ , si consideri la successione

$$x_0 = a$$

$$x_1 = b$$

$$x_n = x_{n-1} + x_{n-2}$$

Questa successione costituisce una generalizzazione della successione di Fibonacci. Dimostrare e verificare sperimentalmente che il rapporto fra due termini consecutivi

tende al *rappporto aureo*  $\phi$ , ossia:

$$\lim_{n \rightarrow \infty} \frac{x_n}{x_{n-1}} = \frac{\sqrt{5} - 1}{2} \approx 1.618033$$

**9.11** Calcolare un valore approssimato con 3 cifre decimali corrette a cui tende la serie convergente

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots$$

**9.12** Valutare in modo iterativo (lento) un valore approssimativo di  $\pi$  in base alla seguente formula:

$$\pi = 4 \sum_{k=1}^{\infty} (-1)^{k+1} \frac{1}{2k-1} = 4 \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right)$$

Determinare quanti addendi sono necessari per avere una precisione a meno di  $10^{-p}$  ( $p$  cifre decimali corrette). Ottimizzare l'algoritmo raggruppando a due a due i termini della serie:  $\frac{1}{2k+1} - \frac{1}{2k+3} = \frac{2}{4k^2+8k+3}$  e verificare sperimentalmente il miglioramento di efficienza del procedimento.

**9.13** Valutare in modo approssimato il valore della costante  $\pi$  (pigreco), basandosi sul seguente sviluppo in serie:

$$\pi = \sqrt{\sum_{k=1}^{\infty} \frac{1}{k^2}}$$

**9.14** Scrivere un algoritmo corrispondente alla seguente espressione della  $\sqrt{2}$  mediante una frazione continua:

$$\begin{aligned} \sqrt{2} = 1 + & \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{\ddots}}}} \\ & \end{aligned}$$

**9.15** Calcolare mediante un algoritmo il valore limite al quale tende la seguente frazione continua:

$$\begin{aligned} 1 + & \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{\ddots}}}} \\ & \end{aligned}$$

Dimostrare che la successione tende al rapporto aureo  $\phi = \frac{1+\sqrt{5}}{2}$  (per questa dimostrazione serve un ragionamento extra-algoritmico).

9.16 Calcolare

$$\sqrt{2 + \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots}}}}$$

9.17 Determinare il valore al quale converge la seguente espressione:

$$1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}}$$

9.18 Discutere comparativamente i tre approcci *matematico*, *informatico*, *alternativo* presentati nel problema 9.7.1, con particolare riguardo ai seguenti aspetti: conoscenze richieste, efficienza del procedimento ed eleganza.

# 10

---

## FORMALIZZARE

---

*È possibile concepire un sistema formale che imiti tutta la realtà ? In senso lato, la risposta potrebbe sembrare affermativa. Si potrebbe pensare, ad esempio, che la realtà stessa non sia altro che un sistema formale molto complicato.*

D. R. Hofstadter, *Gödel, Escher, Bach*

Agli inizi del secolo scorso il matematico tedesco David Hilbert (1862-1943) introdusse il concetto di *sistema formale* con l'obiettivo di fornire dei criteri rigorosi in base ai quali si potesse stabilire se una data dimostrazione matematica potesse essere considerata corretta. Sulla base dei brillanti risultati conseguiti dalla logica simbolica, Hilbert pensava che tutte le asserzioni matematiche fossero decidibili e, con questa convinzione, intraprese il lavoro di codificazione delle regole del ragionamento matematico. Il sogno di Hilbert venne però infranto da alcuni sconcertanti risultati di Church, Gödel e Turing negli anni trenta.

Nello studio dei sistemi formali ci si limita alla specificazione delle configurazioni di simboli *formalmente corrette*, indipendentemente dal *significato* che può essere attribuito loro. Naturalmente, si cerca di definire come formalmente corrette le configurazioni che risultano utili in base al significato che si intende attribuire loro.

Nonostante l'argomento sia teorico, i sistemi formali trovano interessanti applicazioni in diversi campi fra i quali: teoria dei linguaggi, programmazione logica, intelligenza artificiale, grafica.

## 10.1 I sistemi formali

In modo approssimativo un *sistema formale* può essere definito come un insieme di configurazioni costituite mediante simboli di un prefissato insieme, rispettando delle regole di costruzione. Un metodo per precisare le configurazioni formalmente corrette consiste in una loro elencazione completa. Questo metodo è praticabile solo se le configurazioni sono in un numero finito. Un'alternativa consiste nel precisare un insieme finito di simboli componenti, un insieme finito di configurazioni di partenza corrette ed un insieme finito di regole mediante le quali generare tutte e sole le configurazioni ammissibili. È proprio questo l'oggetto della seguente articolata definizione.

**DEFINIZIONE 13.** Un *sistema formale* (in seguito *s.f.*) è una terna  $\mathcal{F} = (A, S, R)$  dove  $A$ , detto *alfabeto*, è un insieme finito e non vuoto di elementi detti *simboli* o *atomi*;  $S$  è un sottoinsieme privilegiato di configurazioni di partenza, dette *assiomi*;  $R$  è un insieme finito di *regole di inferenza* o *regole di riscrittura* o *produzioni* che specificano come generare delle nuove configurazioni a partire dagli assiomi. Un'aggregazione di simboli è detta *configurazione*. Si dicono *teoremi* le configurazioni che si ottengono applicando le regole di inferenza in un numero finito di passi, a partire da assiomi o teoremi precedentemente ricavati. Le configurazioni che non sono ottenibili vengono dette *non-teoremi*. Con  $Th(\mathcal{F})$  si denota l'insieme dei teoremi del s.f.  $\mathcal{F}$ . Molto spesso le configurazioni in un sistema formale sono delle *sequenze di caratteri* e vengono denominate *stringhe* o *parole* o *frasi* o *formule*.

La figura 10.1 illustra schematicamente gli insiemi di configurazioni che costituiscono gli assiomi, i teoremi ed i non-teoremi.

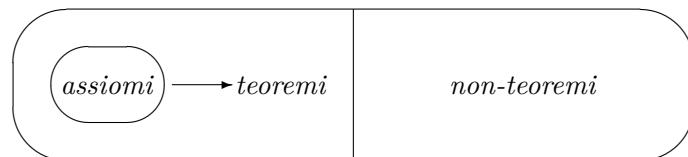


Figura 10.1: Visione insiemistica degli *assiomi*, *teoremi* e *non-teoremi*.

Con la scrittura

$$x \rightarrow y$$

si denota la regola che stabilisce che dalla configurazione  $x$  si può ricavare la configurazione  $y$ , ossia la regola afferma che "Se  $x$  è un assioma o un teorema, allora  $y$  è un teorema". Questa notazione può essere generalizzata ad una scrittura della forma

$$x_1, \dots, x_n \rightarrow y$$

Con riferimento a questa scrittura, gli assiomi o i teoremi  $x_1, \dots, x_n$  si dicono *premesse* e  $y$  la *conclusione*.

Se esiste una catena di regole della forma

$$x \rightarrow y_1 \rightarrow y_2 \rightarrow \cdots \rightarrow y$$

si scrive

$$x \xrightarrow{*} y$$

Se  $x, y, \alpha, \beta$  sono stringhe ( $y, \alpha, \beta$  eventualmente vuote), se  $x \rightarrow y$  è una regola e se  $\alpha x \beta$  è una stringa contenente come sottostringa la stringa  $x$ , allora si può ricavare la stringa  $\alpha y \beta$  e si scrive

$$\alpha x \beta \Rightarrow \alpha y \beta$$

Una tale trasformazione viene detta **derivazione**.

Qualora, per un dato sistema formale, la descrizione degli assiomi e/o delle regole risulti troppo laboriosa o, quando gli assiomi e/o le regole siano infiniti, impraticabile, si ricorre a delle forme generalizzate mediante le quali si descrivono delle intere classi di assiomi e regole (*schemi di assiomi e schemi di regole*). A questo scopo vengono utilizzate delle scritture che coinvolgono delle **variabili** (dette anche **simboli non terminali**), ossia dei metasimboli non appartenenti all'alfabeto di base del sistema formale. Questo meccanismo risulterà chiaro analizzando i sistemi formali descritti negli esempi che seguono.

**Esempio 10.1.1 -** Consideriamo il seguente sistema formale:

$$\text{Alfabeto} = \{+, -\}$$

$$\text{Assiomi} = \{-+\}$$

$$\text{Regole} = \{x+ \rightarrow x+-, +x \rightarrow +xx, x-+y \rightarrow x+y, x-+y \rightarrow x+-y\}$$

dove con  $x$  e  $y$  si denotano due qualsiasi stringhe di  $+$  e  $-$ . In questo caso i simboli  $x$  e  $y$  non sono simboli dell'alfabeto, ma variabili ausiliarie che denotano generiche stringhe di  $+$  e  $-$  e servono per definire le regole. Ad esempio, la regola  $x+ \rightarrow x+-$  precisa che se una stringa termina con  $+$  allora si può generare una nuova stringa aggiungendo  $-$  alla fine.

**Esempio 10.1.2 -** Una tipica applicazione dei sistemi formali è costituita dalla definizione di stringhe da accettarsi come *formalmente corrette*. Queste problematiche si incontrano nello studio delle *grammatiche generative* e dei *linguaggi formali*. Il seguente sistema formale definisce le stringhe di parentesi tonde  $($  e  $)$  bilanciate secondo le usuali convenzioni dell'Algebra:

$$\text{Alfabeto} = \{(, )\}$$

$$\text{Assiomi} = \{x\}$$

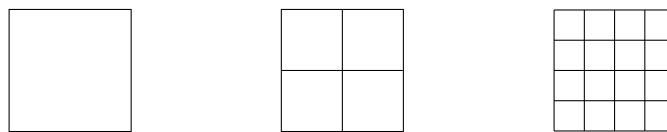
$$\text{Regole} = \{x \rightarrow (), x \rightarrow (x), x \rightarrow xx\}$$

Si noti qui che l'unico assioma è costituito da una variabile; in casi come questo, i teoremi del sistema formale sono costituiti dalle stringhe di simboli terminali dell'alfabeto ottenibili applicando le regole a partire dall'assioma ed iterando

il procedimento fino a che si arriva ad una stringa composta di soli simboli terminali.

Generalizzando la definizione di sistema formale data precedentemente, oltre ai sistemi formali basati su stringhe di caratteri, si possono considerare dei sistemi formali basati su altri elementi componenti, ad esempio figure.

*Esempio 10.1.3 -* Consideriamo il seguente sistema formale composto da figure; l'assioma è costituito da un quadrato; la regola generatrice è la seguente: *Suddividi ciascun quadrato in quattro parti uguali mediante due segmenti paralleli ai lati.* Sono teoremi di questo sistema formale le seguenti figure:



## 10.2 Dimostrazioni nei sistemi formali

Nel contesto dei sistemi formali si presentano problematiche della seguente forma:

*Dato il sistema formale  $S$  ed la stringa  $t$ , decidere se  $t$  è un teorema di  $S$ .*

Problematiche di questo tipo possono essere affrontate mediante una MdT. In questo caso la descrizione del sistema formale  $S$  e la stringa  $t$  costituiscono i *dati* del problema (e pertanto vanno scritti inizialmente sul nastro). La realizzazione di una siffatta MdT è stato uno dei sogni di molti matematici, filosofi e teorici informatici nei primissimi decenni del secolo scorso; tale sogno venne infranto nel 1931 ad opera di Gödel, come espresso nel suo famoso *teorema di incompletezza*. Data questa premessa, acquista utilità la seguente definizione: *Dimostrare* un teorema in un sistema formale significa esibire i passi attraverso i quali si genera una data stringa, a partire dagli assiomi. Nel caso si tratti di un non-teorema bisogna dimostrare che non esiste alcun percorso; spesso questa situazione viene realizzata *ragionando esternamente* al sistema formale.

*Esempio 10.2.1 -* Con riferimento all'esempio 10.1.1 consideriamo il problema di dimostrare la stringa  $-$ , ossia generare tale stringa partendo dall'assioma  $-+$  ed applicando una sequenza di regole. La dimostrazione può procedere applicando le regole del sistema formale fino ad arrivare a dimostrare la stringa obiettivo; se non si riesce, può significare che i tentativi sono stati male orientati, oppure può essere che la stringa obiettivo non sia raggiungibile. Questa seconda alternativa è dimostrabile solo se si riesce a *ragionare fuori del sistema formale*, ossia applicando dei ragionamenti che non si basano sulle regole del sistema. Soluzione del problema: ci si può facilmente convincere che non esiste alcuna regola che elimina il segno  $+$  dall'assioma di partenza; pertanto la stringa  $-$  non è un teorema.

*Esempio 10.2.2 -* Con riferimento all'esempio 10.1.2 la stringa  $((())((x))$  può essere dimostrata mediante la seguente catena di derivazioni:

$$\begin{aligned} x &\rightarrow xx \Rightarrow (x)x \Rightarrow ((x))x \Rightarrow ((x))(x) \\ &\Rightarrow ((x))(xx) \Rightarrow ((x)((x))x) \Rightarrow ((x)((x))) \end{aligned}$$

### 10.3 Sistemi formali e significato

Un sistema formale si comporta come un *generatore di stringhe*. Per rendere utile ed interessante un sistema formale si cerca di attribuire ai *teoremi formali* un qualche *significato*; spesso si cerca di attribuire dei significati ad alcune configurazioni di base e definire le regole di produzione dei teoremi formali in modo che riflettano un qualche significato. Si afferma che un teorema è *significativo* se riflette, per una qualche interpretazione dei simboli che vi compaiono, una corrispondenza con una qualche porzione della realtà. L'attribuzione di significato avviene stabilendo una corrispondenza biunivoca fra stringhe ed una porzione della realtà di interesse. Usando le parole di Hofstadter: “una interpretazione è dotata di significato nella misura in cui riflette fedelmente un isomorfismo con il mondo reale”.

Poiché un sistema formale deve riflettere un qualche isomorfismo con la realtà, le regole d'inferenza devono generare, nell'interpretazione assunta, solo teoremi significativi. Il rischio di definire dei sistemi formali per i quali non esistano delle interpretazioni significative è scongiurato poiché, come dice Hofstadter, “di solito chi costruisce un sistema formale ha già una interpretazione in mente”. E se la costruzione del sistema formale avviene su suggerimento del mondo reale, allora si tende a definirlo in modo che, nell'interpretazione assunta, tutti i teoremi siano veri e tutti i non-teoremi falsi. Bisogna comunque porre attenzione sul fatto che la manipolazione dei segni è indipendente dalla loro interpretazione.

Le funzioni di corrispondenza fra *realità* (e gli *oggetti reali*) e *sistema formale* (ed i *teoremi*), schematicamente descritte nella figura 10.2, vengono dette *rappresentazione* ed *interpretazione*.

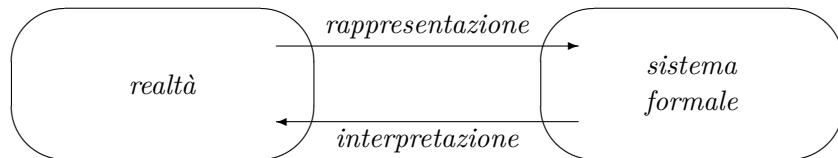


Figura 10.2: Le funzioni di corrispondenza fra realtà e sistema formale.

Molto spesso la realtà di interesse è costituita da oggetti del modo della matematica quali numeri, espressioni, figure, frasi, teoremi (della matematica) ed altri oggetti importanti per l'informatica quali programmi, macchine,

algoritmi, sistemi formali. Sfruttando questa corrispondenza biunivoca, un sistema formale può essere visto come un *generatore di oggetti matematici*.

L'ambiente degli oggetti matematici (numeri, espressioni, equazioni, teoremi, ...) rappresenta il terreno privilegiato di indagine mediante i sistemi formali. A tale scopo è necessario stabilire una corrispondenza fra gli oggetti della matematica e le configurazioni di simboli generate e manipolate nei sistemi formali. Incominciamo, nei prossimi esempi, dagli oggetti matematici più elementari e primordiali, i numeri naturali  $0, 1, 2, \dots$ .

*Esempio 10.3.1 -* Un numero naturale  $n$  può essere codificato in stringa usando gli usuali dieci caratteri cifra secondo la codifica decimale oppure mediante le due cifre binarie 0 e 1, secondo la notazione binaria o, addirittura, usando un singolo carattere, ad esempio  $|$ , mediante una sequenza di  $n + 1$  caratteri  $|$ , riservando la stringa  $|$  per la codifica del numero zero. Similmente, si può rappresentare una coppia  $(m, n)$  di numeri naturali mediante una sequenza di  $m + 1$  caratteri  $|$  separata con uno spazio (o altro carattere) da un sequenza di  $n + 1$  caratteri  $|$  a seguire. Questa è una delle possibili notazioni per i numeri naturali (notazione unaria) e sarà utilizzata più avanti quando si utilizzeranno le macchine di Turing. Naturalmente, al posto del simbolo  $|$  si potrebbe prendere qualsiasi altro simbolo. In modo simile si possono definire dei sistemi formali per codificare i numeri naturali in altre basi di numerazione. Quello che segue è un sistema formale che *descrive* l'insieme dei *numeri naturali*:

$$\text{Alfabeto} = \{| \}$$

$$\text{Assiomi} = \{| \}$$

$$\text{Regole} = \{x \rightarrow x| \}$$

È immediato riconoscere che le configurazioni che si possono ottenere in questo sistema formale sono tutte le sequenze di un arbitrario numero di trattini. Quest'insieme di configurazioni può rappresentare i numeri naturali (inteso come insieme privo di ogni struttura algebrica) in base alla seguente corrispondenza:

$$\begin{array}{rcl} | & \leftrightarrow & 0 \\ || & \leftrightarrow & 1 \\ ||| & \leftrightarrow & 2 \\ & \dots & \end{array}$$

Se si vuole limitarsi alla rappresentazione dei numeri naturali positivi si può adottare la seguente corrispondenza:

$$\begin{array}{rcl} | & \leftrightarrow & 1 \\ || & \leftrightarrow & 2 \\ ||| & \leftrightarrow & 3 \\ & \dots & \end{array}$$

I sistemi formali si prestano non solo a descrivere degli *oggetti* ma anche delle *operazioni*.

*Esempio 10.3.2 - I seguenti teoremi, costruiti sull'alfabeto  $\{\mathbf{s}, |\}$ , possono essere interpretati come operazione di successivo sui numeri naturali:*

$$T_1 = \{\mathbf{s}| = ||, \mathbf{s}|| = |||, \mathbf{s}||| = |||, \dots\}$$

da interpretarsi come “il successivo di 0 è 1”, “il successivo di 1 è 2” e così via. Un altro insieme di teoremi che possono essere interpretati come operazione di successivo sui numeri naturali è il seguente:

$$T_2 = \{|-||, ||-|||, |||-|||, \dots\}$$

In modo analogo a quanto visto nell'esempio precedente, si possono definire dei sistemi formali che rappresentano le quattro operazioni aritmetiche.

*Esempio 10.3.3 - (sistema PQ di Hofstadter) Consideriamo un esempio di sistema formale proposto da Hofstadter<sup>1</sup>. L'alfabeto consiste di soli tre simboli: P, Q, -. Gli assiomi sono costituiti dalle stringhe definite mediante lo schema*

$$xP-Qx-$$

dove  $x$  denota una generica stringa di caratteri -. Le regole di inferenza sono descritte dal seguente schema

$$xPyQz \rightarrow xPy-Qz-$$

dove  $x, y, z$  denotano generiche stringhe di trattini. In questo sistema formale si possono generare, ad esempio, i teoremi che compaiono nella seguente catena di produzioni:

$$-\mathbf{P}-\mathbf{Q}- \rightarrow -\mathbf{P}-\mathbf{Q}--$$

Per questo sistema formale una utile interpretazione consiste nel far corrispondere al segno P il significato di *addizione*, al segno Q il significato di *uguale* e ad una sequenza di trattini il corrispondente *numero naturale positivo* in notazione unaria.

*Esempio 10.3.4 - In modo alternativo, ma equivalente, a quanto descritto nell'esempio precedente, l'operazione di addizione sui numeri naturali, rappresentati in notazione unaria, può essere descritta mediante il seguente sistema formale i cui teoremi risultano di evidente interpretazione.*

$$\text{Alfabeto} = \{|, +, =\}$$

$$\text{Assiomi} = \{|+|=|\}$$

$$\text{Regole} = \{x+y=z \rightarrow x+y|=z|, x+y=z \rightarrow x|+y=z|\}$$

dove  $x, y, z$  sono sequenze di |. Le regole sopra possono essere espresse mediante l'unica regola  $x+y=z \rightarrow y+x=z$ .

<sup>1</sup>Douglas Hofstadter, *Godel, Escher, Bach: An Eternal Golden Braid* (1980)

L'idea di sistema formale può essere adattata per generare oggetti *reali* già interpretati (e non semplicemente una loro *codifica*), come illustra il seguente esempio.

*Esempio 10.3.5* - Assumiamo il numero 1 come assioma e consideriamo il seguente insieme di regole di generazione sui numeri naturali:

$$\begin{array}{ll} x \rightarrow x/2 & \text{se } x \text{ pari} \\ x \rightarrow 3x + 1 & \text{se } x \text{ dispari} \end{array}$$

È immediato constatare che i teoremi di questo sistema formale sono i numeri dell'insieme  $\{1, 2, 4\}$ . Questo sistema formale può essere generalizzato prendendo come assioma un generico numero naturale positivo. In questi casi si ottiene una sequenza di valori che termina la sua corsa sempre in 1 (e poi prosegue ciclicamente con  $1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow \dots$ ). Tale situazione è nota con il nome di “problema del  $3n + 1$ ”.

## 10.4 Il sistema formale MU

Il sistema formale *MU* venne originariamente proposto da Hofstadter nel libro già citato. I teoremi di questo sistema sono delle stringhe sull'alfabeto  $\{M, I, U\}$ . L'unico assioma è costituito dalla stringa *MI*. Le regole di generazione delle stringhe sono le seguenti quattro:

- (a) Se una stringa termina con *I*, si può accodare *U* alla fine.
- (b) Se una stringa inizia con *M*, si può accodare la sottostringa che segue tale *M*.
- (c) Se in una stringa è presente la sottostringa *III*, si può sostituirla con una *U*.
- (d) Se in una stringa è presente la sottostringa *UU*, si può eliminarla.

Formalmente tale sistema può essere definito come segue:

$$\text{Alfabeto} = \{ M, I, U \}$$

$$\text{Assiomi} = \{ MI \}$$

$$\text{Regole} = \{ xI \rightarrow xIU, Mx \rightarrow Mxx, xIIIy \rightarrow xUy, xUUy \rightarrow xy \}$$

dove  $x$  e  $y$  sono delle variabili che rappresentano delle generiche stringhe composte da simboli dell'alfabeto. La figura 10.3 descrive la generazione di alcune stringhe.

Il problema consiste nel decidere se *MU* è un teorema del sistema *MU*, ossia se la stringa *MU* è generabile a partire dalla stringa *MI* applicando le regole ammesse. È evidente anche in questo caso che un ragionamento interno al sistema, ossia l'applicazione diretta delle regole può condurre alla soluzione (ossia alla generazione della stringa *MU*); nel caso in cui l'applicazione diretta non ci guida al (supposto) teorema, rimane vivo il dubbio che *MU* possa non essere un teorema del sistema; in questo caso qualsiasi applicazione meccanica

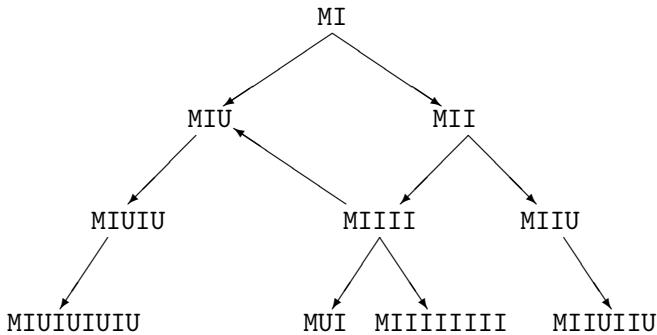


Figura 10.3: Generazione di alcune stringhe nel sistema *MU*.

delle regole (ossia rimanendo all’*interno* del sistema) non può darci alcuna informazione e non ci rimane altro che *uscire dal sistema*. Si invita a ragionare a risolvere il problema proposto. Come suggerimento per la soluzione del problema notiamo che ci sono regole che *allungano* e regole che *accorcianno*. A seguire è riportata la soluzione, in negativo, del problema del sistema MU. Anche in questo caso viene applicato un tipico ragionamento *esterno al sistema*.

*Dimostrazione.* La configurazione MU non contiene I. Quindi, nella generazione delle configurazioni, si devono applicare delle regole che eliminano la I di MI. Le regole che modificano il numero di I sono la (b) e la (c). La (b) raddoppia il numero delle I, mentre la (c) permette di eliminare le I di un multiplo di 3. Poichè queste due regole non modificano la divisibilità per 3 del numero di I, partendo da una I, si conclude che nessuna combinazione delle regole (b) e (c) può eliminare la I della configurazione iniziale MI.  $\square$

## 10.5 Giochi come sistemi formali

La definizione di sistema formale, essendo che le regole generative dei teoremi possono essere fissate con estrema libertà, può sembrare una sorta di gioco vuoto, senza alcun legame con la *realtà*. Questo potenziale scollegamento dalla realtà permette di dare la seguente definizione. Si chiamano *giochi* i sistemi formali che non hanno alcun legame con la realtà. In tali casi le regole del sistema formale costituiscono le *regole del gioco*. I sistemi formali possono essere usati per modellizzare dei giochi fra due persone. Ad esempio gli scacchi o giochi simili. Si parte da una configurazione e le trasformazioni (mosse) vengono proposte alternativamente dai due giocatori. Un gioco di solitario è analogo: si vince se si raggiunge una configurazione vincente.

*Esempio 10.5.1 -* Il gioco degli scacchi può essere pensato come un sistema formale nel quale i simboli dell’alfabeto sono i pezzi degli scacchi, l’assioma è la configurazione iniziale dei pezzi e le regole sono le regole di movimento dei pezzi, nel contesto della configurazione della scacchiera. In questo caso i teoremi sono le configurazioni potenzialmente raggiungibili nel corso di una partita a scacchi, muovendo i pezzi nel rispetto delle regole del gioco.  $\square$

## 10.6 Gli L-sistemi

Nel 1968 il biologo e matematico danese Aristid Lindenmeyer ha ideato e studiato una particolare classe di sistemi formali adatti a descrivere l'evoluzione di organismi multicellulari formati da filamenti lineari o ramificati (piante, fiori, alghe, ...). Tali formalismi, detti *sistemi di Lindenmeyer* o, semplicemente, *L-sistemi*, sono dei sistemi di riscrittura che permettono di descrivere la morfogenesi delle piante mediante un processo di elaborazione tipico dei sistemi formali. Un L-sistema è caratterizzato da un assioma di partenza e da una o più regole di riscrittura; le regole non devono ammettere alternative ed inoltre, ad ogni passo, devono essere applicate in modo forzato: vanno applicate *localmente* (ossia ad ogni simbolo della stringa) e *parallelamente* (ossia contemporaneamente ad ogni simbolo della stringa).

*Esempio 10.6.1* - L'L-sistema caratterizzato dall'assioma  $X$  e dalle regole di riscrittura  $X \rightarrow Y$ ,  $Y \rightarrow XYZ$ , al primo passo di riscrittura genera la stringa  $Y$ , al secondo passo la stringa  $XYZ$ , al terzo passo la stringa  $YXYZZ$  e così via.  $\square$

*Esempio 10.6.2* - Un noto esempio di L-sistema è fornito dalla *successione di Thue-Morse*, dal nome dei due matematici che l'hanno studiata). È definita dalle seguenti regole di riscrittura:

$$0 \rightarrow 01$$

$$1 \rightarrow 10$$

a partire dall'assioma  $0$ . Le prime stringhe generate da queste regole sono:

0

01

0110

01101001

0110100110010110

01101001100101101001011001101001

Si può notare che ciascuna stringa è costituita dalla stringa ottenuta al passo precedente concatenata con il suo complemento. Questa proprietà vale per tutte le stringhe della successione.  $\square$

*Esempio 10.6.3* - Nel 1202 il matematico italiano Leonardo Pisano, detto Fibonacci, pubblicò il suo trattato *Liber Abaci* che descriveva un famosissimo esempio di successione numerica nota come successione di Fibonacci:

$$0 \ 1 \ 1 \ 2 \ 3 \ 5 \ 8 \ 13 \ 21 \ 34 \ \dots$$

dove ciascun termine della successione è dato dalla somma dei due termini che precedono. La successione è pertanto descrivibile mediante le seguenti formule di ricorrenza:

$$\begin{aligned}a_0 &= 0 \\a_1 &= 1 \\a_n &= a_{n-1} + a_{n-2}\end{aligned}$$

Come studiato dallo stesso Fibonacci, questa successione descrive il processo riproduttivo delle coppie di governato dalle seguenti regole: si parte con una coppia  $F$  di conigli giovani, appena nati; trascorso un mese la coppia diventa fertile ( $G$ ); con cadenza mensile ciascuna coppia di conigli adulti genera una coppia  $F$  di conigli. Questo processo può essere descritto mediante le seguenti regole di riscrittura:

$$\begin{aligned}F &\rightarrow G \\G &\rightarrow GF\end{aligned}$$

Queste regole generano le seguenti stringhe:

$$F \rightarrow G \rightarrow GF \rightarrow GFG \rightarrow GF GG \rightarrow GF GG FG \rightarrow \dots$$

che possono essere interpretate come segue: all'inizio c'è una sola coppia di conigli giovani; dopo un mese tale coppia è diventata adulta ed in grado di generare; dopo un altro mese viene generata una nuova coppia di conigli giovani ( $GF$ ); dopo un altro mese la coppia matura genera una nuova coppia di figli ed i figli generati precedentemente diventano prolifici; dopo 5 mesi la popolazione complessiva di conigli è formata da 5 coppie di conigli maturi e 3 coppie di conigli giovani, ossia, complessivamente, formata da 8 coppie di conigli. Ad ogni mese il numero di coppie di conigli è dato dal corrispondente termine della successione di Fibonacci.  $\square$

*Esempio 10.6.4 -* L'L-sistema definito dalla regola di riscrittura  $F \rightarrow F-F++F-F$  genera, ai primi passi del processo di riscrittura, le seguenti stringhe:

$$\begin{aligned}&F \\&F-F++F-F \\&F-F++F-F-F-F++F-F++F-F-F-F++F-F\end{aligned}$$

Stringhe come queste, all'apparenza insignificanti, risultano interessanti se vengono interpretate geometricamente, nell'ambito della grafica della tartaruga. Interpretando come segue i simboli di queste stringhe, si ottengono delle figure note come *graftali*:

- $F$  : avanza di un tratto unitario
- $+$  : ruota in senso orario di un angolo  $\alpha$  fissato
- $-$  : ruota in senso antiorario di un angolo  $\alpha$  fissato

Eventuali caratteri diversi da quelli sopra riportati, ausiliari alla descrizione delle regole di riscrittura, vengono, in fase di interpretazione, trascurati e non considerati. Con queste interpretazioni, assumendo come angolo di rotazione un angolo di  $60^\circ$ , le stringhe sopra elencate possono essere geometricamente descritte dalle seguenti curve (istanze della *curva di von Kock*) riportata nella figura 10.4.



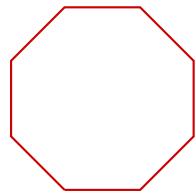
Figura 10.4: Le prime 3 istanze della curva di von Koch generate dalla regola di riscrittura  $F \rightarrow F-F++F-F$ .

□

### 10.6.1 L-sistemi per figure periodiche

Gli L-sistemi si prestano a descrivere figure periodiche aventi una componente che si ripete; tali figure vengono solitamente disegnate mediante il tradizionale ciclo *for* dei linguaggi di programmazione imperativi; usando gli L-sistemi il ciclo viene realizzato mediante una tecnica di *ricorsione in coda*, come si vede negli esempi 10.6.5 e 10.6.6.

*Esempio 10.6.5 - Un ottagono regolare*

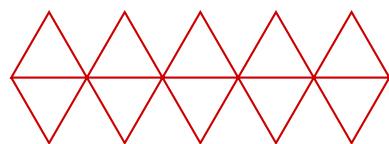


può essere generato dalla semplice regola di riscrittura

$$X \rightarrow F+X$$

ripetuta per 8 volte, usando  $\alpha = 45^\circ$  come angolo di rotazione. □

*Esempio 10.6.6 - La figura ripetitiva*



formata da una serie di coppie di triangoli equilateri aventi un lato in comune può essere descritta mediante un L-sistema caratterizzato dall'assioma X e dalla regola

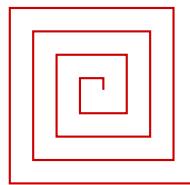
$$X \rightarrow F+F+F-F-FX$$

ripetendo la riscrittura per un numero di volte pari al numero di coppie di triangoli equilateri, usando  $\alpha = 120^\circ$  come angolo di rotazione. □

### 10.6.2 L-sistemi per figure ricorsive

Iterazione e ricorsione sono due aspetti mutuamente connessi che si prestano ad essere indagati analizzando la struttura ed i procedimenti di disegno di alcune figure.

*Esempio 10.6.7 - Una spirale quadrata, come quella descritta nella figura che segue:*



è una figura interessante in quanto si presta ad essere definita facilmente sia in modo iterativo che ricorsivo. Ipotizzando che si sviluppi dall'interno verso l'esterno, una spirale può essere definita nelle seguenti due modalità:

- *iterativa*: ripeti [ disegna un tratto, ruota di 90°, allunga il tratto ]
- *ricorsiva*: disegna un tratto, ruota di 90°, disegna una spirale con il primo tratto allungato

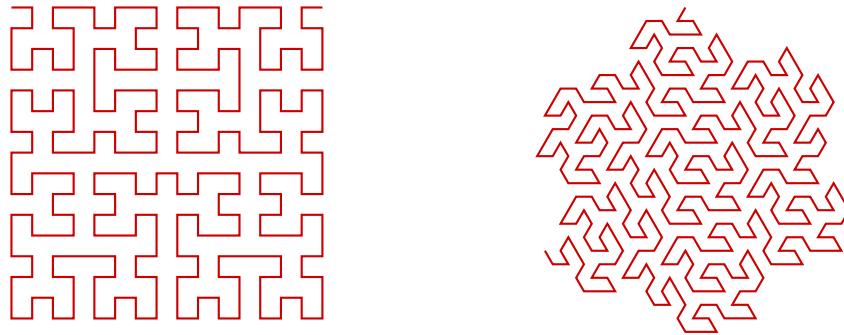
La struttura potenzialmente infinita e l'autosimilarità della spirale con una sua parte viene ben descritta dalla definizione ricorsiva e la natura ricorsiva degli L-sistemi ben si attaglia a questa definizione; adottando questa definizione, il disegno della spirale comporta che, ad ogni passo, si gestiscano i seguenti due sottoproblemi: allungare il segmento di un'unità, disegnare un'altra spirale; nel contesto degli L-sistemi la soluzione di questi due sottoproblemi si esprime mediante le seguenti due regole di riscrittura:

$$\begin{aligned} X &\rightarrow XY+ \\ Y &\rightarrow YF \end{aligned}$$

ripetendo il processo di riscrittura per un numero di volte pari al numero di lati della spirale. □

Gli L-sistemi manifestano la loro potenza espressiva nella descrizione di figure intrinsecamente ricorsive e difficilmente descrivibili in modo iterativo.

*Esempio 10.6.8 - Nella figura 10.5 sono riportate due classiche figure ottenute con degli L-sistemi le cui regole di riscrittura sono riportate in corrispondenza di ciascuna.  $\alpha$  denota la misura dell'angolo di rotazione nell'interpretazione dei segni + e -.*



*Assioma:* L

*Regole:*

$$L \rightarrow +RF-LFL-FR+$$

$$R \rightarrow -LF+RFR+FL-$$

*Angolo*  $\alpha = 90^\circ$

*Assioma:* X

*Regole:*

$$X \rightarrow X+YF++YF-FX-FXFX-YF+$$

$$Y \rightarrow -FX+YFYF++YF+FX-FX-Y$$

*Angolo*  $\alpha = 60^\circ$

Figura 10.5: Sistemi formali che generano la *curva di Hilbert* (a sinistra) e la *curva esagonale di Gosper* (a destra).

□

### 10.6.3 L-sistemi con ramificazioni

Negli L-sistemi, per descrivere delle strutture geometricamente composte da ramificazioni vengono utilizzati i due simboli speciali [ e ] che vengono interpretati, rispettivamente, come *inizio* e *fine* della ramificazione. Con queste convenzioni, la stringa  $x_1[y_1]x_2[y_2]\dots x_{n-1}[y_{n-1}]x_n$  rappresenta un albero avente per fusto la sequenza di elementi  $x_1, x_2, \dots, x_n$  e ramificazioni  $y_1, y_2, \dots, y_{n-1}$ ; tali ramificazioni possono essere a loro volta composte.

*Esempio 10.6.9* - La stringa  $a[b]c[d[e][f]g]h$  può essere interpretata graficamente mediante la figura 10.6.

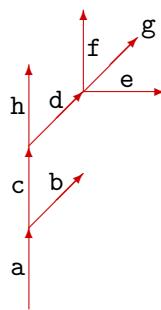
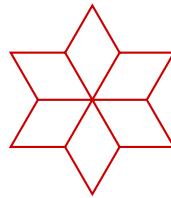


Figura 10.6: Interpretazione grafica della stringa  $a[b]c[d[e][f]g]h$ .

*Esempio 10.6.10* - Un fiocco di neve a 6 punte come il seguente



può essere generato dalla semplice regola di riscrittura

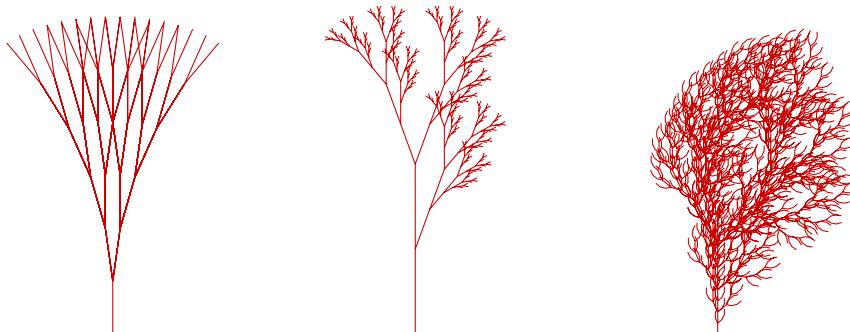
$$X \rightarrow [F+F++F]+X$$

ripetuta per 6 volte, usando  $\alpha = 60^\circ$  come angolo di rotazione.  $\square$

#### 10.6.4 L-sistemi che generano piante

Una delle più interessanti applicazioni degli L-sistemi consiste nella generazione di stringhe che, nell'interpretazione della grafica della tartaruga, riproducono delle figure assimilabili ad organismi vegetali e piante, dette *piante graftali*.

**Esempio 10.6.11 -** Nella figura 10.7 sono riportati alcuni esempi di piante graftali ottenute con degli L-sistemi descritti in corrispondenza di ciascuna;  $n$  indica il numero di iterazioni effettuate nell'applicazione delle regole ed  $\alpha$  denota la misura dell'angolo di rotazione nell'interpretazione dei segni + e -. Tali figure, per risultare più realistiche, potrebbero essere eventualmente arricchite graficamente con delle foglie alle estremità dei rami o con dei frutti o fiori opportunamente localizzati, con diversi spessori dei rami a seconda del loro livello di perifericità. Ancora, per rendere più realistiche le piante che vengono generate, si possono imporre delle ramificazioni verso sinistra e verso destra alternativamente di un dato angolo, eventualmente combinando l'uso di numeri casuali per definire la lunghezza dei rami e l'angolo fra di loro.



$$F \rightarrow F [+F] [-F]$$

$$\alpha = 8^\circ$$

$$n = 5$$

$$X \rightarrow F [+X] F [-X] +X$$

$$F \rightarrow FF$$

$$\alpha = 20^\circ$$

$$n = 6$$

$$F \rightarrow FF+ [+F-F-F]-[-F+F+F]$$

$$\alpha = 25^\circ$$

$$n = 4$$

Figura 10.7: Esempi di piante graftali generate da un L-sistema.

### 10.6.5 Una macchina per gli L-sistemi

Progettiamo ora una macchina per gestire gli L-sistemi. Chiamiamo una tale macchina con il termine *L-macchina*. Il nucleo principale di una L-macchina è costituito da un *L-trasformatore* di stringhe che agisce secondo lo schema illustrato nella figura 10.8.

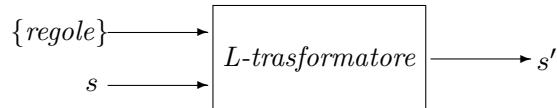


Figura 10.8: Schema di un *L-trasformatore*: ricevuto in ingresso un insieme di regole di trasformazione ed una stringa  $s$  da trasformare, viene generata in uscita una stringa  $s'$  trasformata secondo le regole specificate.

La macchina-trasformatore descritta sopra costituisce il motore della L-macchina descritta nella figura 10.9.

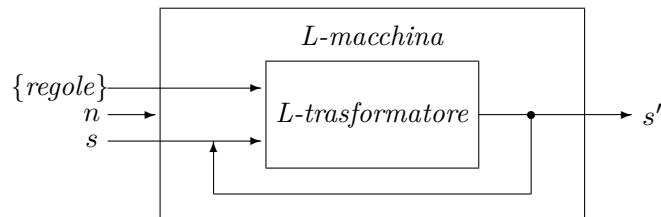


Figura 10.9: Schema di una *L-macchina*: ricevuta in ingresso una stringa  $s$  da trasformare ed un insieme di regole di trasformazione, il numero  $n$  dei passi di trasformazione da applicare, viene prodotta in uscita una stringa  $s'$  trasformata secondo le regole specificate.

Un *L-trasformatore* ed una *L-macchina* come descritte sopra possono essere realizzate mediante una funzione o blocco senza stato, a cui vengono forniti i dati in ingresso, producendo in uscita la stringa elaborata. In alternativa si possono realizzare delle macchine con stato, costituito dall'insieme delle regole di trasformazione che governano la trasformazione della stringa in ingresso. In questo caso le macchine vengono inquadrati come oggetti e le regole possono essere specificate come parametri al costruttore delle macchine oppure possono essere settate successivamente alla costruzione, mediante un apposito metodo d'istanza.

Nel caso in cui la stringa  $s'$  ottenuta dall'elaborazione di una L-macchina debba essere interpretata geometricamente mediante la grafica della tartaruga, la stringa  $s'$  va in ingresso ad un blocco *disegnatore*, collegato sequenzialmente alla macchina che genera la stringa, assieme a due ulteriori ingressi che rappresentano l'angolo di rotazione (per interpretare i simboli + e -) ed il passo di avanzamento (per interpretare il simbolo F).

## ESERCIZI

**10.1** Con riferimento ad un generico sistema formale, precisare cosa significano le seguenti affermazioni:

1.  $T$  è un non-teorema.
2.  $T$  non è un teorema.
3. Il teorema  $T$  non è dimostrabile.

Fra queste affermazioni, stabilire quali sono fra loro equivalenti.

**10.2** Stabilire in cosa si differenziano le seguenti due attività:

1. Dimostrare che  $T$  è un teorema del sistema formale  $F$ .
2. Generare il teorema  $T$  a partire dagli assiomi del sistema formale  $F$ .

Dire se il ragionamento interno e/o esterno al sistema formale può essere di aiuto nello svolgimento di queste due attività.

**10.3** Un sistema è costituito da fiori ( $F$ ) e rami ( $R$ ). Modellare mediante un L-sistema la seguente situazione:

- si parte da un *flore*
- ad ogni passo:
  - un *flore* diventa un *ramo*
  - ogni *ramo* produce un *flore*
  - due *rami* consecutivi diventano un unico *ramo*

Descrivere l'evoluzione del sistema in modo grafico. Calcolare quanti fiori e quanti rami sono presenti al passo  $n$ .

**10.4** Definire un sistema formale organizzato attorno ai seguenti elementi: *rami, foglie, fiori*.

**10.5** Si consideri il sistema formale dove l'insieme delle configurazioni è costituito dall'insieme dei punti di un segmento  $AB$ , l'insieme degli assiomi è costituito dagli estremi  $A, B$  del segmento; è fissata inoltre un'unica regola, a due premesse,

$$P, Q \rightarrow M$$

dove  $M$  è il punto medio del segmento  $PQ$ . Con riferimento al sistema formale sopra definito, dire come devono essere interpretate le seguenti affermazioni:

1.  $T$  è un teorema.
2.  $T$  è un non-teorema.
3. Dimostrare il teorema  $T$ .
4. Il teorema  $T$  non è dimostrabile.

Mediante degli alberi di dimostrazione, dimostrare i teoremi  $C, D, E, F$ , essendo  $\overline{AC} = \frac{1}{2}\overline{AB}$ ,  $\overline{AD} = \frac{3}{4}\overline{AB}$ ,  $\overline{AE} = \frac{3}{8}\overline{AB}$ ,  $\overline{AF} = \frac{5}{16}\overline{AB}$ . Quali sono i non-teoremi di questo sistema formale?

**10.6** Si hanno tre monete disposte allineate su un piano. Si parte dalla disposizione iniziale *Testa-Croce-Testa* e si vuole raggiungere la disposizione finale *Croce-Testa-Croce*. Sono ammesse le seguenti mosse:

1. si può rivoltare la moneta centrale quando e quante volte si vuole
2. si può rivoltare una delle due monete laterali se e solo se le altre due hanno la stessa faccia.

Descrivere un adeguato sistema formale che modellizzi il gioco proposto. Trovare una soluzione. Dimostrare che ogni configurazione è raggiungibile. Suggerimento: descrivere mediante i nodi di un grafo le diverse possibili disposizioni e mediante archi le possibili transizioni di configurazione.

**10.7** Analizzare il sistema formale descritto nell'esempio 10.1.3, modificando come segue la regola: *Suddividi un quadrato in quattro parti uguali mediante due segmenti ortogonali fra loro e paralleli ai lati*.

**10.8** Si definiscano due sistemi formali che producano gli insiemi di teoremi  $T_1$  e  $T_2$  descritti nell'esempio 10.3.2.

**10.9** Con riferimento al sistema formale costituito del gioco degli scacchi descritto nell'esempio 10.5.1, stabilire quali sono i *simboli* dell'alfabeto, cosa sono gli *assiomi*, i *teoremi*, i *non-teoremi* e cosa significa *dimostrare un dato teorema*. Esprimere le affermazioni riportate nell'esercizio 3 che precede, particolarizzandole a questo caso di sistema formale. Stabilire quali delle seguenti decisioni possono essere risolte in base all'*analisi* di una data configurazione del sistema:

1. il Bianco ha lo stesso numero di pedoni del Nero
2. il Bianco ha effettuato l'arrocco
3. la posizione del bianco è preferibile

**10.10** Con riferimento alla terza figura riportata nell'esempio 10.6.11, spiegare perché, nonostante l'apparente bilanciamento di rotazioni a destra (+) e rotazioni a sinistra (-) nelle regole di definizione della figura, l'albero risulta piegato a destra.

**10.11** Spiegare perché non è possibile dimostrare, in un dato sistema formale, che una data configurazione non è un teorema, ragionando solo con le regole del sistema formale considerato.

**10.12** Definire un sistema formale i cui teoremi siano costituiti dalle stringhe di cifre rappresentanti i *numeri naturali espressi in notazione binaria*.

**10.13** Definire un sistema formale i cui teoremi siano interpretabili come *moltiplicazione sui numeri naturali*.

**10.14** Definire un sistema formale  $F$  le cui configurazioni sono costituite dalle stringhe composte da un numero pari di simboli +. Dimostrare, nel sistema formale  $F$ , il teorema +++++.

**10.15** Definire un sistema formale le cui configurazioni sono costituite dalle stringhe palindrome di + e -, ossia dall'insieme  $\{+, -, ++, --, +++, ++-, ---, --+, +++, +--+,\dots\}$ . Dimostrare, nel sistema formale descritto, il teorema +--+--.

**10.16** Definire un sistema formale i cui teoremi sono costituiti da stringhe di parentesi tonde bilanciate, disposte a *buccia di cipolla*; ad esempio sono teoremi i seguenti:  $()$ ,  $(( ))$ ,  $(((( )))$ .

**10.17** Definire un sistema formale i cui teoremi sono costituiti dalle sequenze di parentesi bilanciate, in modo da rispettare le usuali regole delle espressioni algebriche; ad esempio sono teoremi le seguenti configurazioni:  $( )$ ,  $( () )$ ,  $( () ) ( )$  mentre non lo sono le seguenti:  $( ) ( )$ ,  $( ( ) )$ . Descrivere un algoritmo per decidere, scandendo una sola volta da sinistra a destra, se una data configurazione di parentesi è un teorema del sistema formale sopra definito.

**10.18** Stabilire la figura che viene generata dal seguente L-sistema:

$$\text{Assioma} = \{X\}$$

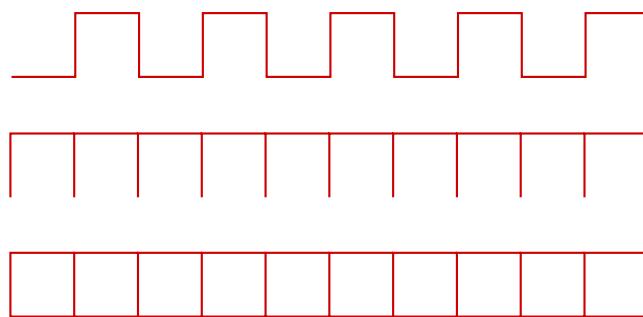
$$\text{Regole} = \{X \rightarrow F-F-F-F, F \rightarrow F-F+F+F+F-F-F-F+F\}$$

$$\text{Angolo } \alpha = 90^\circ$$

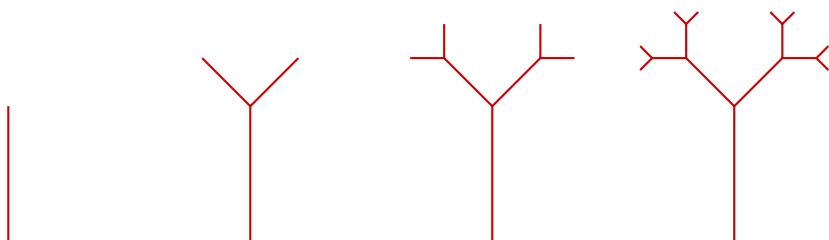
**10.19** Codificare mediante stringhe dell'alfabeto  $\{F, +, -, [, ]\}$  le figure sotto descritte, adottando la tecnica degli L-sistemi.



**10.20** Definire degli L-sistemi che generino le figure ad *onda quadra*, ad *arco multiplo* ed a *quadrati adiacenti*, come descritto a seguire.

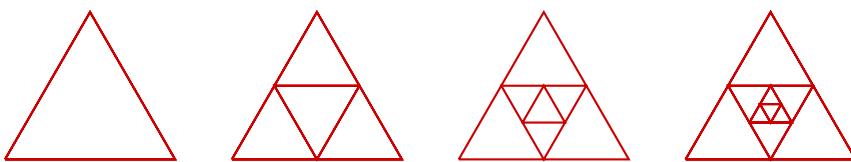


**10.21** Definire un L-sistema che generi un *albero binario* aventi rami lunghi la metà del ramo padre; le prime tre istanze sono descritte nella figura che segue.

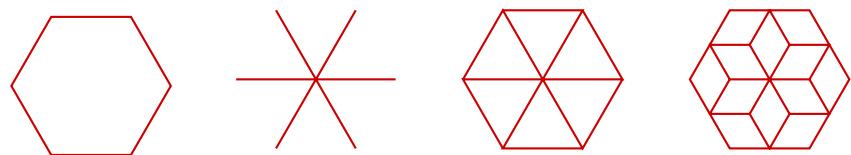


**10.22** Una *quadrifolia* è una stella di mare avente 4 steli di lunghezza unitaria, uniti in centro e disposti ad angolo retto. Ad ogni ciclo di vita ogni estremità di una quadrifolia genera 3 steli disposti ad angolo retto, ciascuno lungo metà del ramo padre. Descrivere un sistema formale che, con l'interpretazione sopra descritta, generi delle quadrifolie, ai vari stadi dei cicli di vita.

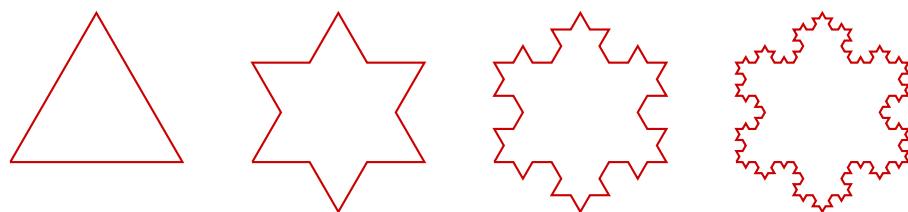
**10.23** Definire un L-sistema che generi una sequenza di triangoli equilateri, ciascuno avente come vertici i punti medi del triangolo precedente, come si vede nella figura che segue che riporta alcuni passi della generazione.



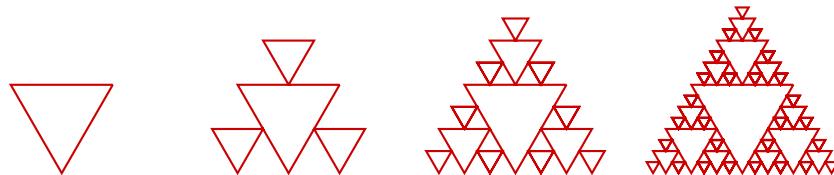
**10.24** Descrivere degli L-sistemi che generino le seguenti figure.



**10.25** Definire un L-sistema che generi il *fiocco di von Koch* ottenuto generando la curva di von Koch su ciascuno dei tre lati di un triangolo equilatero.

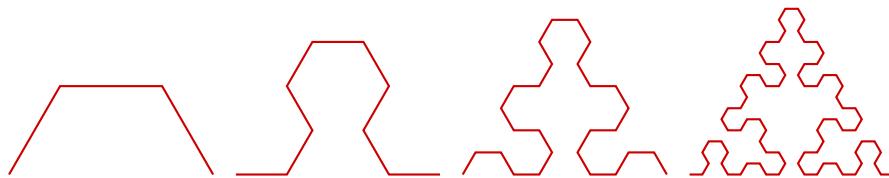


**10.26** Definire un L-sistema che generi il *triangolo di Sierpinski*, una figura costituita da triangoli equilateri: l'assioma è rappresentato da un triangolo equilatero; ad ogni passo, all'interno di ogni triangolo generato al precedente passo, viene generato un nuovo triangolo che genera altri 3 triangoli, come illustra la figura che segue che riporta alcuni passi della generazione di un triangolo di Sierpinski.



**10.27** Una variazione del triangolo di Sierpinski permette di ottenere una figura ricorsiva costituita da un'unica spezzata, costruita come segue: la figura base è

costituita da una spezzata composta da 3 lati di uguale lunghezza formanti fra loro un angolo di 120 gradi, come indicato nella figura che segue. Ad ogni passo ciascun segmento viene sostituito dalla figura base sottesa dal lato stesso. Descrivere la generazione della figura mediante un L-sistema. Determinare la lunghezza  $l(n)$  della figura al passo  $n$ -esimo. Dimostrare che tale lunghezza tende ad infinito al crescere di  $n$ .



**10.28** Analizzare le seguenti 4 istanze di ordine 0, 1, 2, 3 (per diverse lunghezze F) della *curva di Sierpinski*. Descrivere un L-sistema che generi la curva.

