

Capitolo 3

Elaborazione e Strutturazione dell'Informazione

(Introduzione ai Sistemi Informatici - Sciuto et alii)

Il calcolatore come strumento per gestire informazione



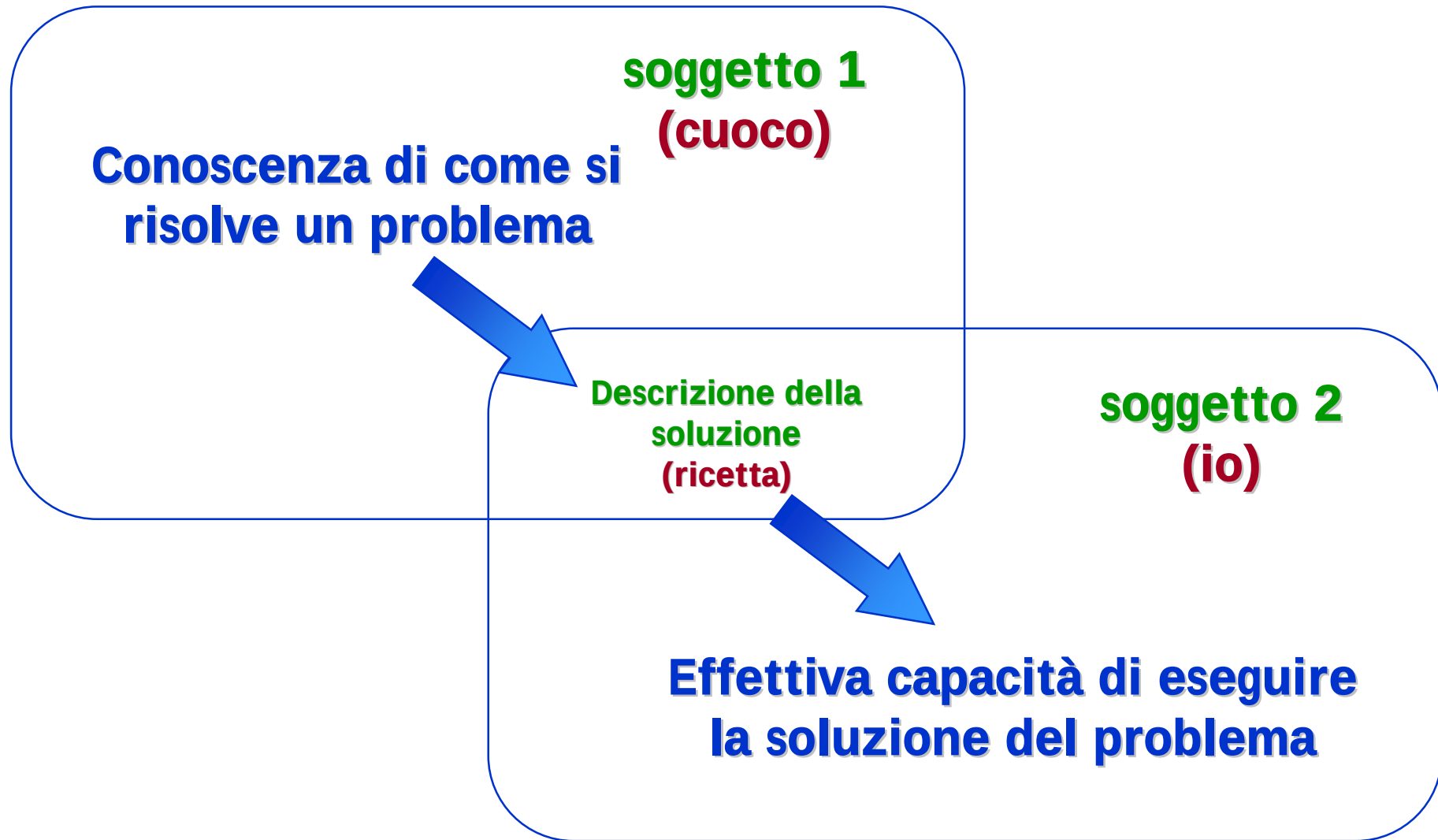
tradizionalmente
l'enfasi è sui

sistemi di elaborazione
delle informazioni ...

Soluzione di un problema



La soluzione di un problema

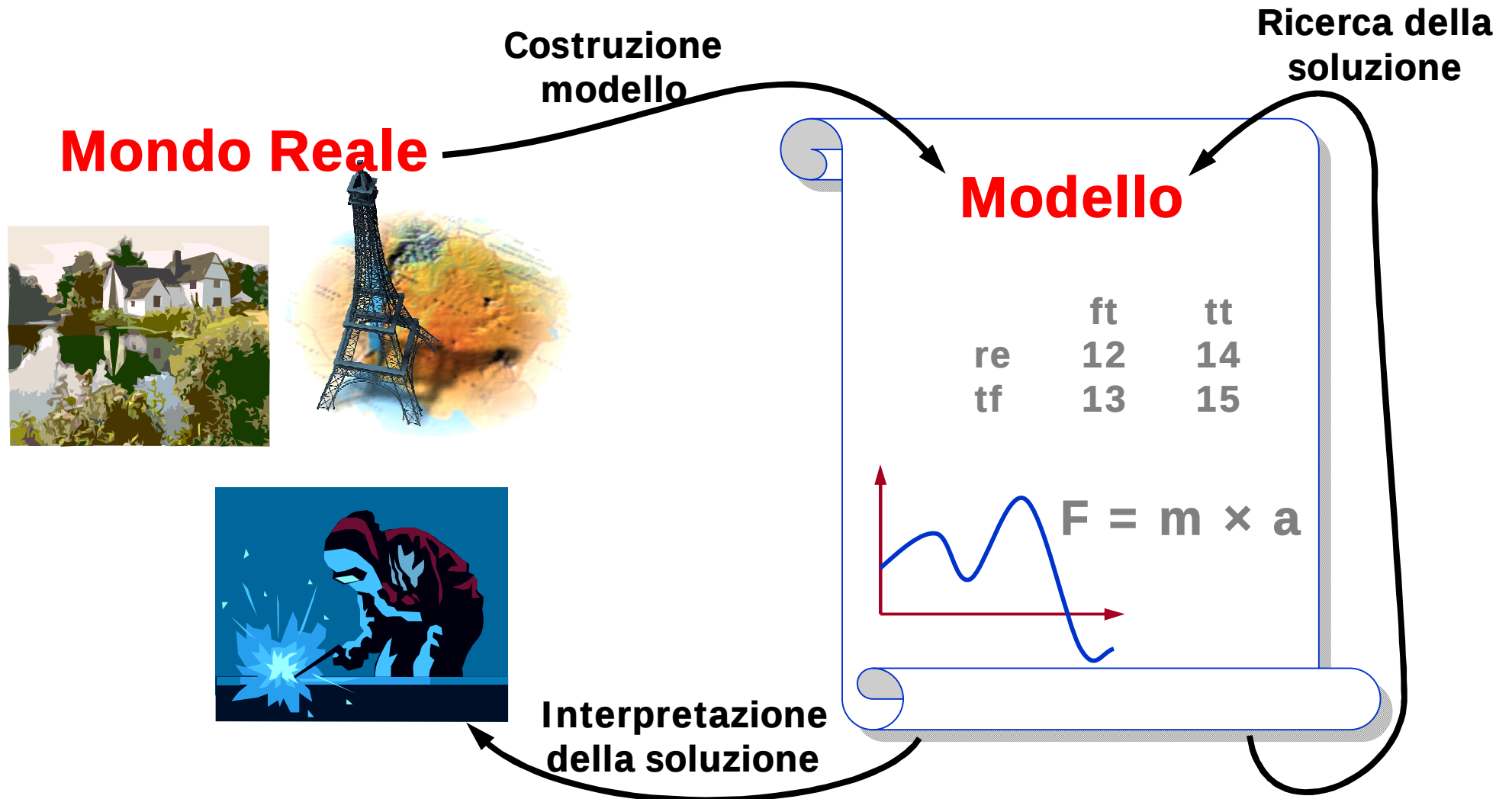


Problemi ed Algoritmi

Definire il problema

- Eliminare le **ambiguità** nella formulazione del problema
- Individuare il **risultato** che si vuole ottenere, gli obiettivi da raggiungere
- Evidenziare
 - le **regole** da rispettare
 - i **vincoli** interni ed esterni
 - i **dati** espliciti ed impliciti
- Eliminare i dettagli inutili ed ambigui

Relazione tra Realtà e Modello



Soluzioni ed esecutori

➤ La descrizione della soluzione di un problema dipende dall'esecutore

- esecutore con un livello medio di scolarità $\Rightarrow$
“**determina la superficie s di un cerchio di cui è noto il raggio r** ”;
- esecutore che non conosce come calcolare l'area del cerchio $\Rightarrow$
“**la superficie di un cerchio è $s = \pi r^2$** ”;
- esecutore che non conosce π $\Rightarrow$
“**la superficie di un cerchio è $s = \pi r^2$ e π indica *pi greco* ed è **3.1415****”;
- ... $\Rightarrow$ “**eleva al quadrato il raggio e quindi moltiplica il risultato per π** ”;
- ... $\Rightarrow$ “**moltiplica il raggio per se stesso e poi il risultato per **3.14****”;
- ...

➤ Descrizione della soluzione di un problema sia accettabile per un esecutore

- si scompone il problema originario in sottoproblemi;
- si scompongono i sottoproblemi in sotto-sottoproblemi;
- si prosegue nella scomposizione fino a giungere a **problemi elementari** (o **primitivi**), risolvibili direttamente dall'esecutore

Istruzioni e azioni elementari

- Ogni **istruzione elementare** è associata a un'**azione elementare** (oppure a una **successione di azioni elementari**) che può essere direttamente compiuta dall'esecutore.
- **Processo che porta alla soluzione di un problema:**
 - per il soggetto **descrittore** richiede un'attività di scomposizione progressiva del problema, fino a giungere a una successione di **istruzioni elementari** (ciascuna associata al corrispondente **problema elementare**);
 - per il soggetto **esecutore** richiede l'esecuzione delle **azioni elementari** associate alle istruzioni elementari identificate. La condizione che le azioni elementari siano eseguite in successione, cioè secondo un ordine definito, è necessaria perché, in generale, ognuna di esse opera sui dati che riceve dalle azioni eseguite precedentemente.
- Le **azioni elementari** vengono interpretate in **termini funzionali**, come delle entità che trasformano i dati che ricevono in ingresso (*input*) in risultati (*output*), con ciò prescindendo dalle modalità con cui tale trasformazione viene effettuata, cioè assumendo un modello "a scatola nera" (*black box*).

Procedura effettiva

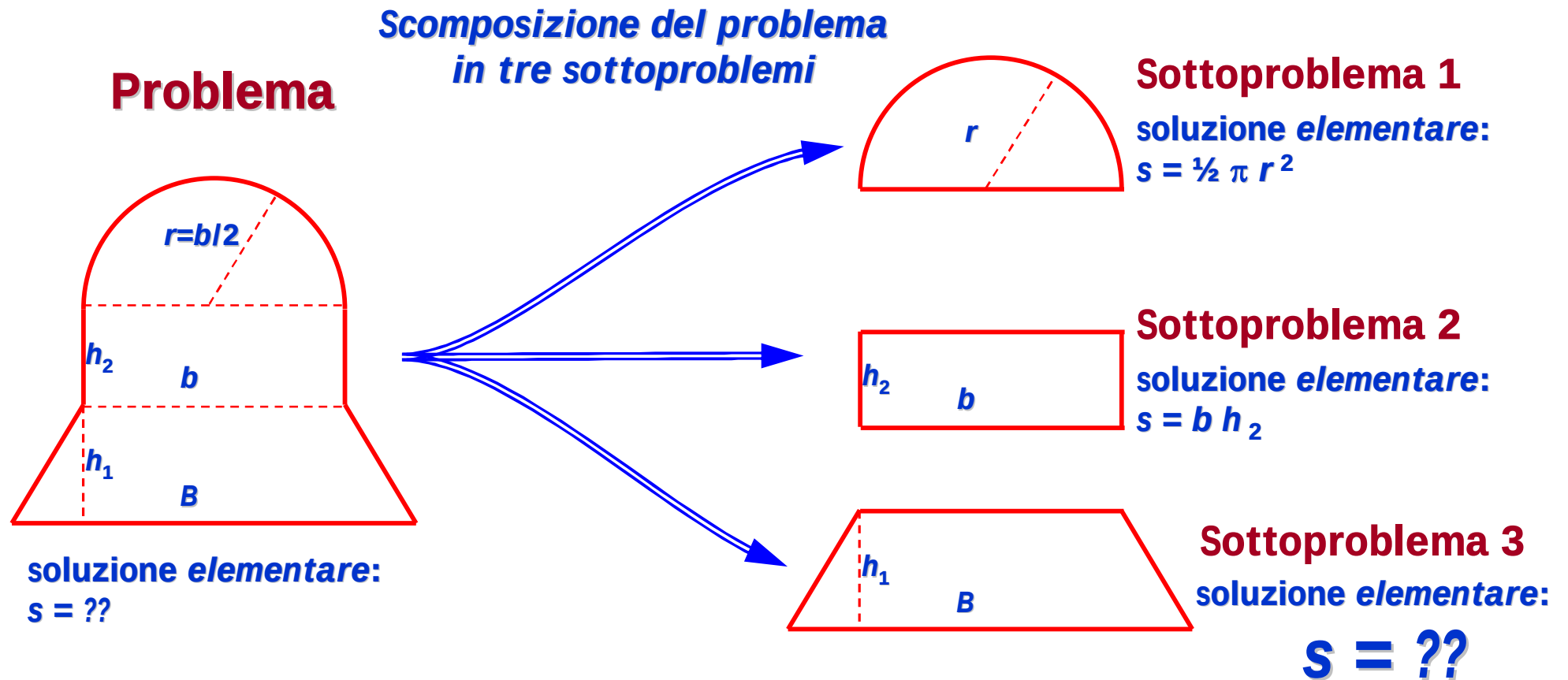
➤ Si dice ***procedura effettiva per un esecutore*** una successione di azioni tale che:

- tutte le azioni della successione sono *elementari* per l'esecutore, che è in grado di eseguire ciascuna di esse in un **tempo finito** e in modo **deterministico**, cioè ottenendo sempre gli stessi risultati a parità di input;
- è fissato l'ordine di esecuzione delle azioni;
- è esplicitamente specificato il modo in cui un'azione utilizza i risultati delle azioni che la precedono.

Soluzione effettiva

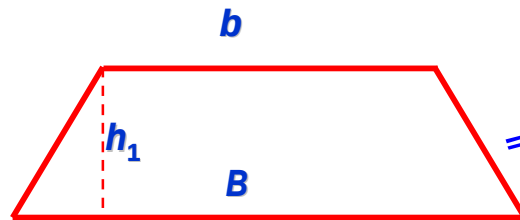
- **Dati un problema P e un esecutore E , si definisce ***soluzione effettiva del problema P per l'esecutore E*** **una successione di istruzioni elementari tale che:**
 - l'esecutore è in grado di interpretare le istruzioni nella successione e quindi di associare a ciascuna di esse l'azione (o la successione di azioni) che deve compiere per eseguirla;
 - la successione di azioni risultante dall'interpretazione delle istruzioni costituisca una procedura effettiva per l'esecutore stesso.**
- **In generale, possono esistere diverse soluzioni effettive dello stesso problema per lo stesso esecutore**

Area di una campana (1)



Area di una campana (2)

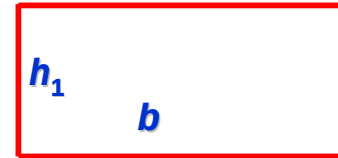
*Scomposizione del sottoproblema 3 in
tre ulteriori sottoproblemi*



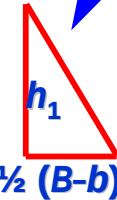
Sottoproblema 3
soluzione effettiva:
 $s = \frac{1}{2} (\frac{1}{2}(B-b) h_1) +$
 $b h_1 +$
 $\frac{1}{2} (\frac{1}{2}(B-b) h_1)$



Sottoproblema 3.1
soluzione elementare:
 $s = \frac{1}{2} (\frac{1}{2}(B-b) h_1)$



Sottoproblema 3.2
soluzione elementare:
 $s = b h_1$

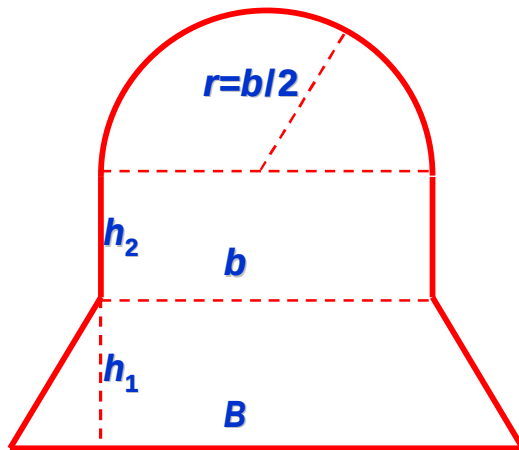


Sottoproblema 3.3
soluzione elementare:
 $s = \frac{1}{2} (\frac{1}{2}(B-b) h_1)$

*Composizione delle soluzioni dei tre
sottoproblemi 3.1, 3.2 e 3.3 per
risolvere il sottoproblema 3*

Area di una campana (3)

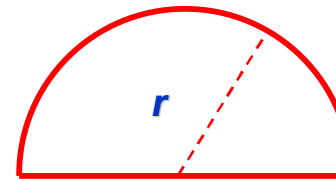
Problema



soluzione effettiva:

$$s = \frac{1}{2} \pi r^2 + b h_2 + \frac{1}{2} (\frac{1}{2}(B-b) h_1) + b h_1 + \frac{1}{2} (\frac{1}{2}(B-b) h_1)$$

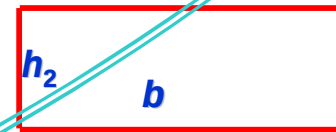
Composizione delle soluzioni dei tre sottoproblemi 1, 2 e 3 per risolvere il problema originario



Sottoproblema 1

soluzione elementare:

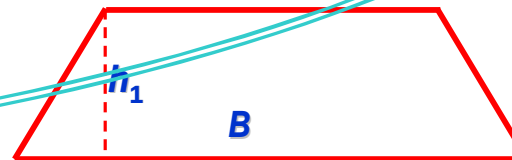
$$s = \frac{1}{2} \pi r^2$$



Sottoproblema 2

soluzione elementare:

$$s = b h_2$$



Sottoproblema 3

soluzione effettiva:

$$s = \frac{1}{2} (\frac{1}{2}(B-b) h_1) + b h_1 + \frac{1}{2} (\frac{1}{2}(B-b) h_1)$$

Algoritmo (definizione informale)

- Sequenza **finita** di istruzioni,
- **comprensibili** da un esecutore (automatico),
- che descrive come **realizzare un compito** (come risolvere un “problema”).

➤ Alcuni esempi

- Istruzioni di montaggio di un elettrodomestico
- Uso di un terminale Bancomat
- Calcolo del massimo comune divisore di numeri naturali

Esecutori e linguaggi

➤ Un esecutore è definito in base a tre elementi:

- l'insieme delle **operazioni** che è capace di compiere;
- l'insieme delle **istruzioni** che capisce (**sintassi**);
- quali **operazioni** associa ad ogni **istruzione** che riconosce (**semantica**).

➤ Il calcolatore “capisce” le istruzioni che fanno parte del **linguaggio macchina**

- istruzioni primitive semplici (e.g. max 2 operandi)
- attenzione all'efficienza (costi, complessità, velocità)
- difficile e noioso da utilizzare per un programmatore

➤ La soluzione si dice **effettiva** se l'esecutore è in grado di:

- Interpretarla (associa alle istruzioni le corrispondenti operazioni)
- eseguire le azioni (in un tempo finito!)

Proprietà di un'azione elementare

➤ **Finitezza**

- l'azione deve concludersi in un tempo finito

➤ **Osservabilità**

- l'azione deve avere un effetto osservabile, cioè deve produrre qualcosa

➤ **Riproducibilità**

- a partire dallo stesso stato iniziale, la stessa azione deve produrre sempre lo stesso risultato

Dal problema alla soluzione automatica

- **Specifiche dei requisiti:**

descrizione precisa e corretta dei requisiti

---> **cosa?**

- **Progetto:**

procedimento con cui si individua la soluzione

---> **come?**

- **Soluzione: algoritmo**

Proprietà degli algoritmi

➤ Correttezza

L'algoritmo perviene alla soluzione del compito cui è preposto, **senza difettare di alcun passo fondamentale**

➤ Completezza

descrivono la **soluzione** non di un singolo problema, ma **di una intera classe di problemi strutturalmente equivalenti**.

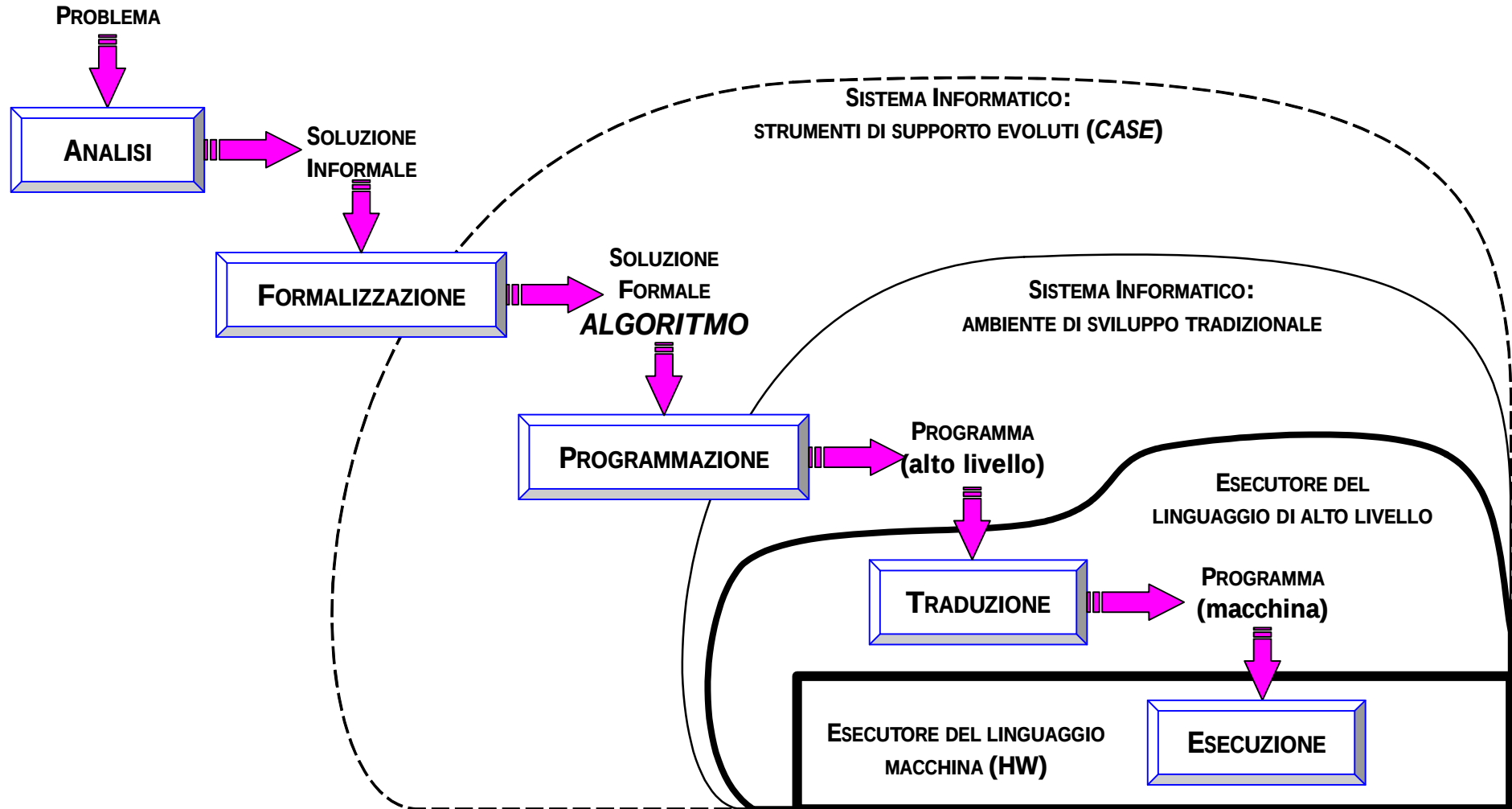
➤ Efficienza

- L'algoritmo perviene alla soluzione del problema usando la **minima quantità di risorse fisiche**
 - tempo di esecuzione, memoria, ...

Alcuni concetti

- **Algoritmo** = descrizione di come si risolve un problema
- **Programma** = algoritmo scritto in modo che possa essere eseguito da un calcolatore (linguaggio di programmazione)
- **Linguaggio macchina** = linguaggio effettivamente “compreso” da un calcolatore, caratterizzato da
 - istruzioni primitive semplici (e.g. max 2 operandi)
 - attenzione all'efficienza (costi, complessità, velocità)
 - difficile e noioso da utilizzare per un programmatore
- **Due aspetti rilevanti:**
 - **produrre algoritmi** (capire la sequenza di passi che portano alla soluzione di un problema)
 - **codificarli in programmi** (renderli comprensibili al calcolatore)

Sviluppo di un programma



Algoritmi

Formalizzazione

Codifica degli algoritmi

- **Algoritmo formulato per essere comunicato tra esseri umani (*il nostro punto di vista*)**
 - sintetico e intuitivo
 - codificato in linguaggi informali o semi-formali (linguaggio naturale, diagrammi di flusso, ...)
- **Algoritmo formulato per essere eseguito automaticamente**
 - preciso ed eseguibile
 - codificato in linguaggi comprensibili dagli esecutori automatici (linguaggio macchina o linguaggio di programmazione di alto livello)

Algoritmi e variabili

➤ **Gli algoritmi sono parametrici:**

- producono un risultato che dipende da un insieme di dati (parametri) di partenza;
- descrivono la **soluzione non di un singolo problema, ma di una intera classe di problemi** strutturalmente equivalenti.
- **Esempi:**
 - l'algoritmo per la moltiplicazione di due numeri specifica come effettuare il prodotto di *tutte* le possibili coppie di numeri;
 - l'algoritmo per la ricerca di un libro nello schedario della biblioteca vale per tutti i possibili libri;
 - ...

➤ **Le istruzioni dell'algoritmo fanno riferimento a *variabili (o parametri)*, il cui valore non è fissato a priori ma cambia a seconda della situazione elaborativa in cui l'esecutore si trova.**

Uso delle variabili (1/3)

➤ All'interno di **espressioni**,

- l'esecutore usa il **valore contenuto nelle variabili** per calcolare il risultato dell'espressione,
- per esempio

(op1 + op2) × op3 oppure

op1 / (op2 - op3),

sono variabili che contengono valori

Uso delle variabili (2/3)

➤ in istruzioni di **assegnamento**

- introdurre nel contenitore identificato dal nome della variabile il valore specificato a destra dell'assegnamento ($\leftarrow$);

- **Esempio:**

$r \leftarrow 35$

(assegna 35 alla variabile il cui nome è **r**)

$pi \leftarrow 3,14$

...

Uso delle variabili (3/3)

➤ in istruzioni di **assegnamento** combinate con **espressioni**

- assegna a una variabile il risultato ottenuto dalla valutazione di un'espressione, per esempio in

"circ ← 2 × r × pi"

il risultato dell'espressione **2 × r × pi** viene calcolato utilizzando i valori contenuti nelle variabili **r** e **pi** e il risultato viene poi assegnato alla variabile **circ**;

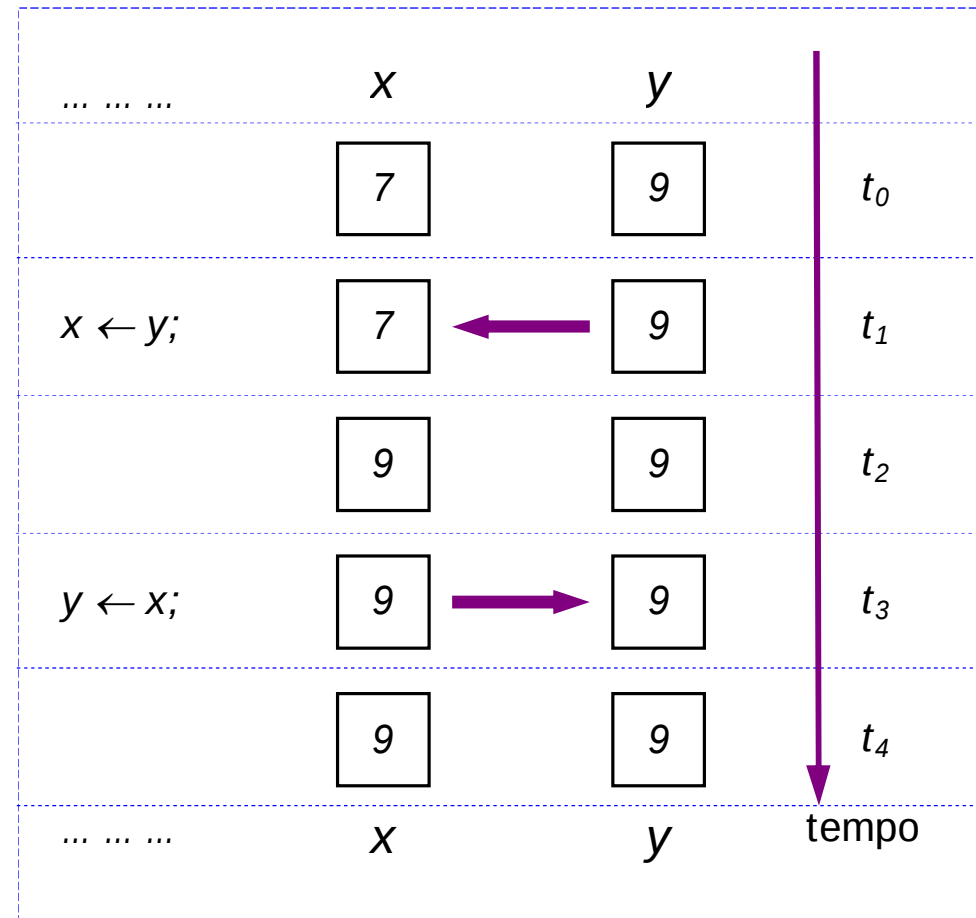
- la stessa variabile può comparire in entrambi i lati dell'istruzione di assegnamento, per esempio in

"k ← k + 1"

il valore contenuto in **k** viene utilizzato per trovare il valore dell'espressione **k + 1** che viene memorizzato come nuovo valore di **k**.

Assegnamento di valori a variabili

- **Esempio:** si ipotizzi di voler **scambiare** i valori contenuti in due variabili x e y .
- **Pseudo-soluzione proposta:** doppio assegnamento del tipo
$$x \leftarrow y$$
$$y \leftarrow x$$
per indicare che il valore di y deve essere copiato in x e che, nello stesso tempo, il valore di x sia trasferito in y .
- **Le istruzioni però vengono eseguite in sequenza!** Quindi l'assegnamento $x \leftarrow y$ viene completato prima di iniziare $y \leftarrow x$.



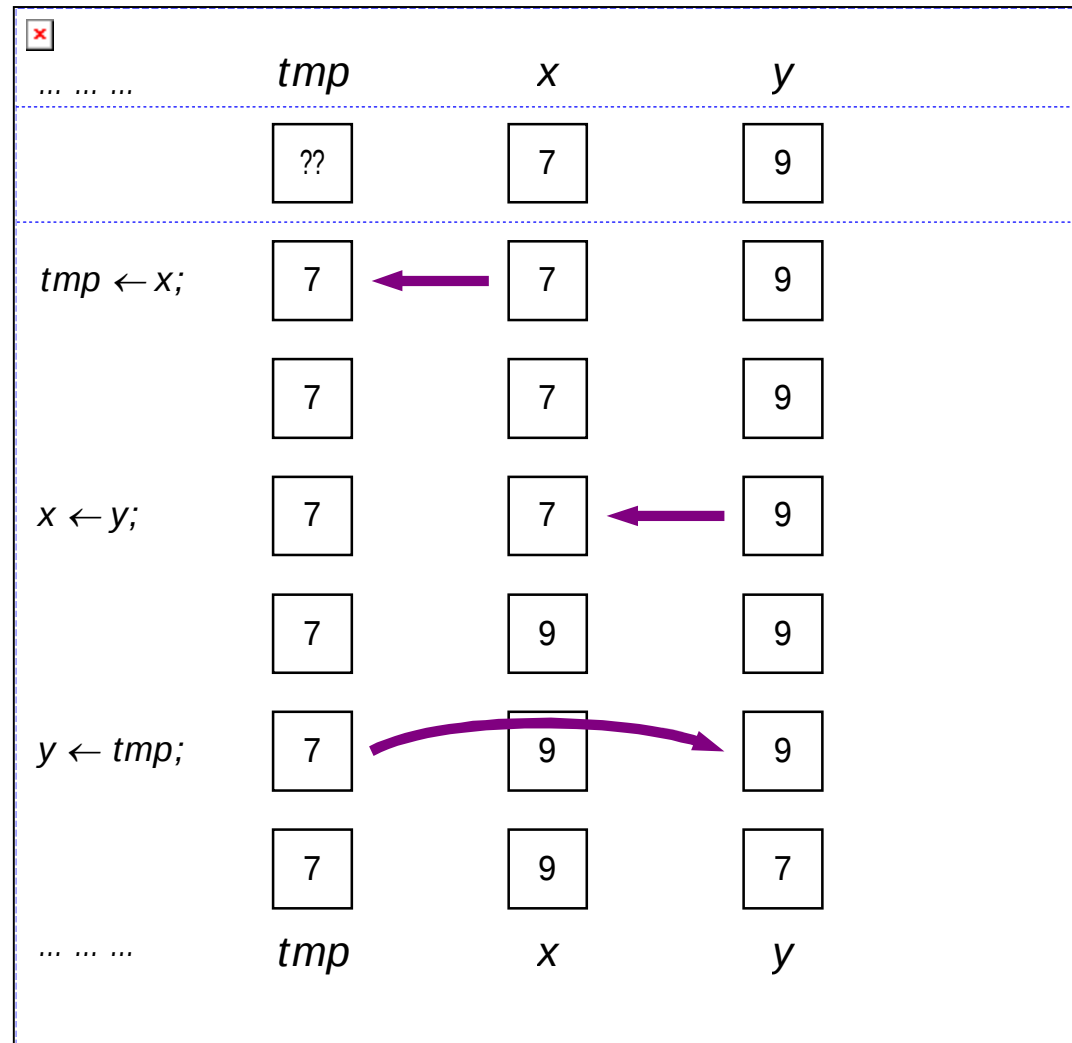
Assegnamento di valori a variabili

Soluzione corretta:

uso di una variabile aggiuntiva (*tmp*), come strumento di memorizzazione temporanea (“buffer”) del valore originariamente contenuto in *x*

$tmp \leftarrow x$
 $x \leftarrow y$
 $y \leftarrow tmp$

In questo modo lo scambio avviene senza perdere i valori originali



Dati e istruzioni

➤ Tipi di dati

- Numeri naturali, interi, reali (1, -2, 0.34)
- Caratteri alfanumerici (A, B, ..)
- Dati logici o booleani (Vero, Falso)
- Array o vettore di n elementi ({1,2,3})

➤ Istruzioni

- Operazioni di input/output (es. *leggi, scrivi*)
- Operazioni aritmetico-logiche (es. $max = A + B$)
- Strutture di controllo (es. *SE...ALLORA, RIPETI*)

Criteri di classificazione dei dati

➤ **Visibilità da parte dell'utente**

- visibile (di ingresso o uscita)
- trasparente (dati temporanei di supporto)

➤ **Variabilità nel tempo**

- costanti
- variabili (acquisizione dall'esterno o assegnazione)

➤ **Struttura**

- elementari (interi, alfanumerici, booleani, ...)
- strutturati (array, matrici, ...)

Operazioni elementari

➤ Operazioni aritmetiche e assegnamenti di valori a singole variabili

- $C \leftarrow (A + B)$

➤ Condizioni sul valore di singole variabili

- se A allora C, altrimenti D

➤ Lettura e scrittura di variabili

- Leggi A oppure Stampa B

Rappresentazione degli algoritmi

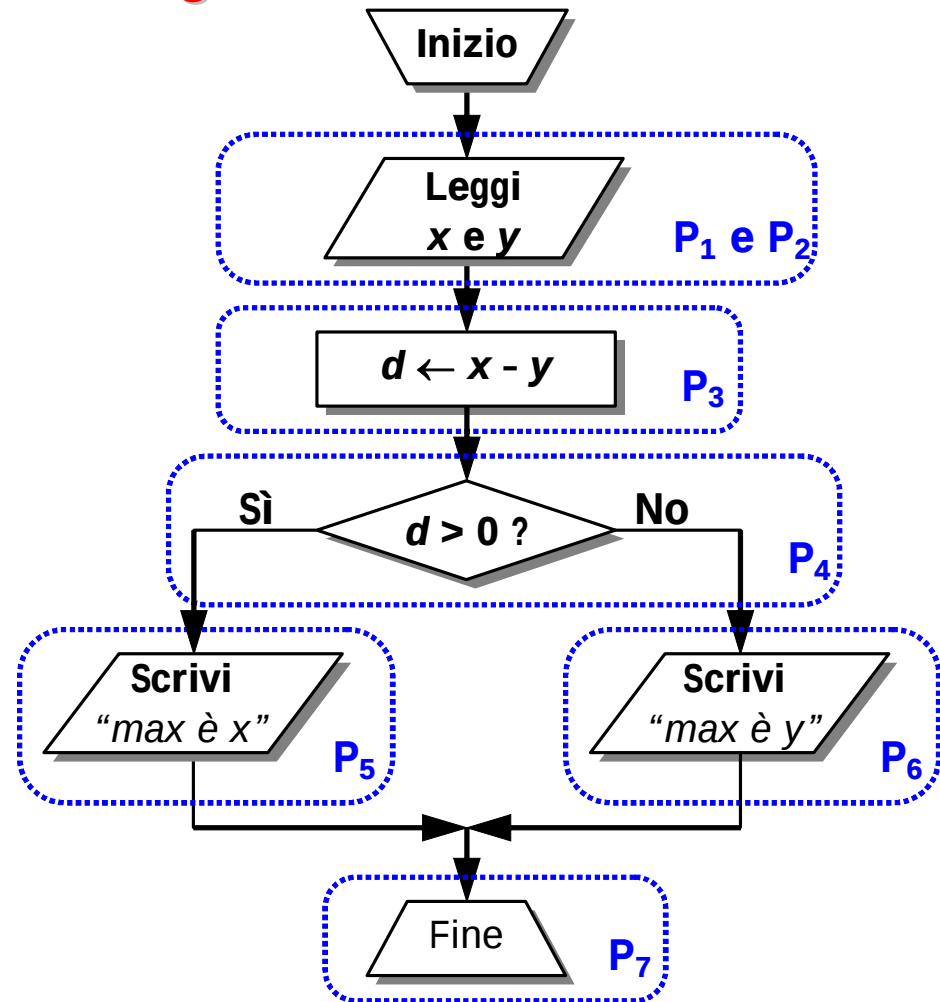
- **Linguaggio naturale**
- **Diagramma a blocchi**
- **Macchine di Turing**
- **Pseudo codice**
- **Linguaggio di programmazione**

Rappresentazione degli algoritmi

➤ Linguaggio Naturale

- Sollevare il ricevitore
- Attendere il segnale di linea libera
- Comporre il numero
- ...

➤ Diagramma di flusso



Rappresentazione degli algoritmi

➤ Pseudo Codice

```
leggi alfa, beta  
prod ← 0  
finché alfa ≠ 0 ripeti  
    prod ← prod + beta;  
    alfa ← alfa - 1;  
stampa prod;
```

➤ Linguaggio di programmazione

```
#include <stdio.h>  
  
int main (void) {  
    puts ("ciao mondo!");  
    return Exit_success;  
}
```