

# La programmazione

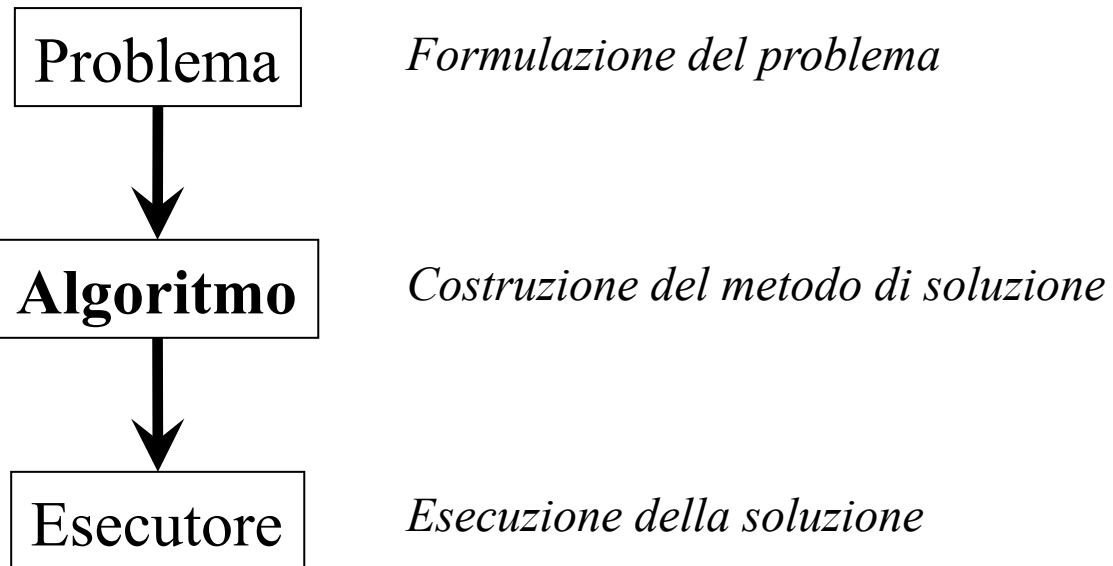
# Cos'è la programmazione?

- Termine usato per indicare le attività che trasformano l'esigenza di risolvere un ***Problema*** in un ***Programma***, inteso come sistema informatico capace di risolvere una classe di problema
- Problema molto ampio, nostro scopo essere in grado di:
  - Realizzare un ***programma***
  - per risolvere un dato ***problema***
  - mediante un ***computer***

# Il problema

- Quando ci troviamo di fronte ad un problema?
  - Ogni volta che ci troviamo di fronte alla necessità di ottenere qualcosa che non sia direttamente raggiungibile a partire da dei dati (*il risultato*)
- Chi risolve il problema?
  - Il “problem-solver” è colui che si occupa di trovare una soluzione (l'*algoritmo*) che ci consenta di raggiungere il risultato a partire dai dati di ingresso
- Chi mette in pratica l'algoritmo?
  - Un *esecutore* che può essere umano o meccanico

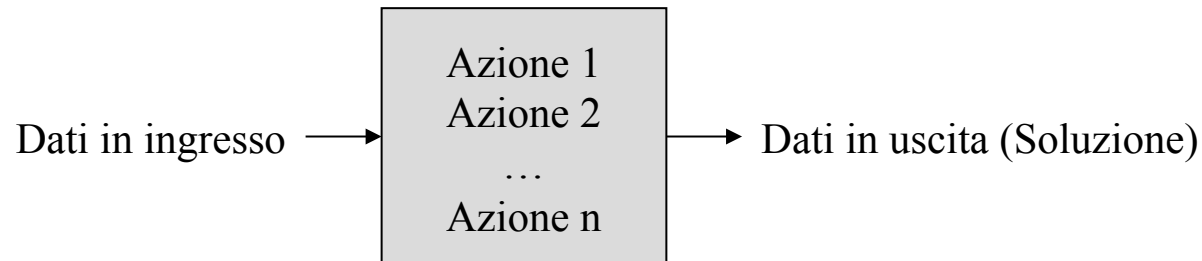
# Le tappe



# Gli Algoritmi

# Algoritmo

- Un algoritmo è una *sequenza di azioni* che permette di trasformare i dati di partenza del problema per ottenere la soluzione



# Proprietà dell'algoritmo

- Finito
  - ogni esecuzione deve terminare
- Generale
  - deve risolvere una classe di problemi
- Completo
  - deve contenere l'indicazione delle azioni da compiere in tutti i casi possibili
- Non ambiguo
  - interpretazione univoca
- Eseguitibile
  - ogni azione deve essere nota all'esecutore
- Se tutte le proprietà elencate sono soddisfatte, l'algoritmo potrà essere eseguito più volte producendo lo stesso risultato

# Descrizione dell'algoritmo

- Proprietà della descrizione di un algoritmo:
  - Indicazione dei **dati** su cui deve operare
  - **Sequenza** esatta delle azioni da compiere
  - Specifica dei **controlli** che determinano l'ordine con cui le azioni vanno eseguite
- ...
  - La **prima azione** che deve essere eseguita
  - La **sequenza esatta** delle azioni
  - L'**azione d'arresto**



# L'esecutore

- Per poter conseguire il risultato deve:
  - Disporre dell'algoritmo
  - *Essere in grado di capire la descrizione dell'algoritmo*
  - Essere in grado di eseguire effettivamente le azioni
  - Disporre dei dati iniziali
  - Comunicare il risultato

# Descrivere l'algoritmo

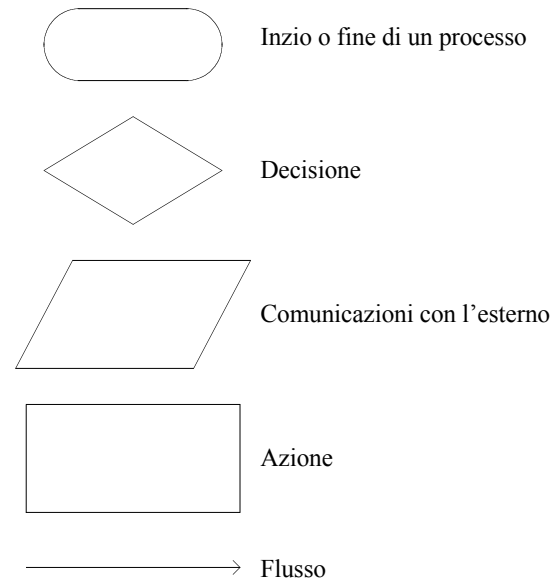
- Il linguaggio con cui si descrive un algoritmo non è un linguaggio di programmazione
- Esistono
  - linguaggi *grafici*
    - fanno uso di simboli grafici
  - linguaggi *lineari*
    - fanno uso di simboli alfanumerici

# Esempi di linguaggi

- Alcuni esempi di linguaggi di descrizione degli algoritmi:
  - Diagrammi di flusso (Flow chart)
  - Carte strutturate
  - Rappresentazioni ad albero

# Flow chart

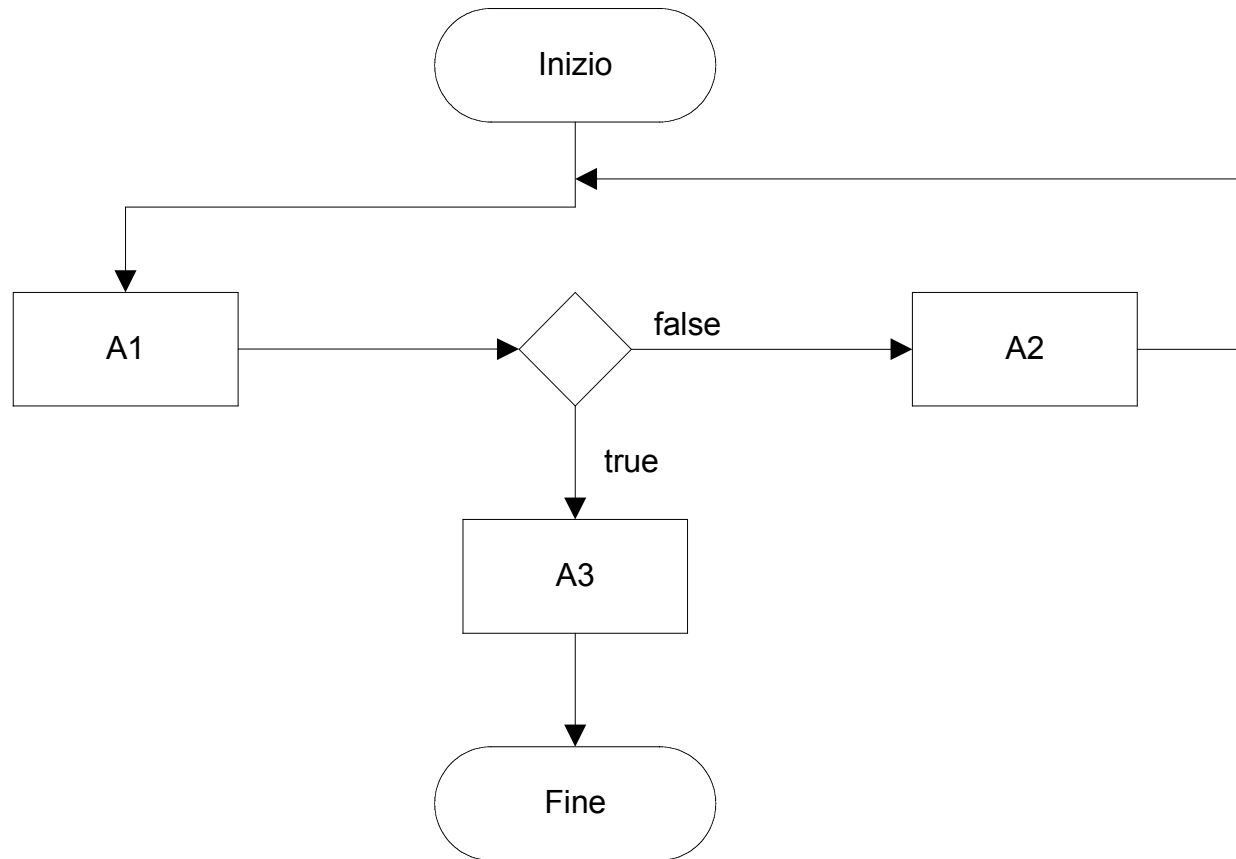
- Vanno bene nel momento in cui occorre descrivere un processo sequenziale
  - Le azioni non si sovrappongono
- I simboli usati sono:



# Flow chart: regole

- Ogni Flow chart contiene almeno:
  - un inizio
  - uno o più azioni
  - eventualmente dei blocchi decisionali
- Regole:
  - Ogni blocco di azione contiene una freccia entrante ed una uscente
  - Ciascun blocco decisionale contiene una freccia entrante e due frecce uscenti
  - Ciascuna freccia può entrare in un blocco, oppure si inserisce in un'altra freccia
  - Ciascun blocco è raggiungibile dal blocco iniziale
  - Il blocco finale è raggiungibile dal blocco iniziale

# Flow chart: esempio

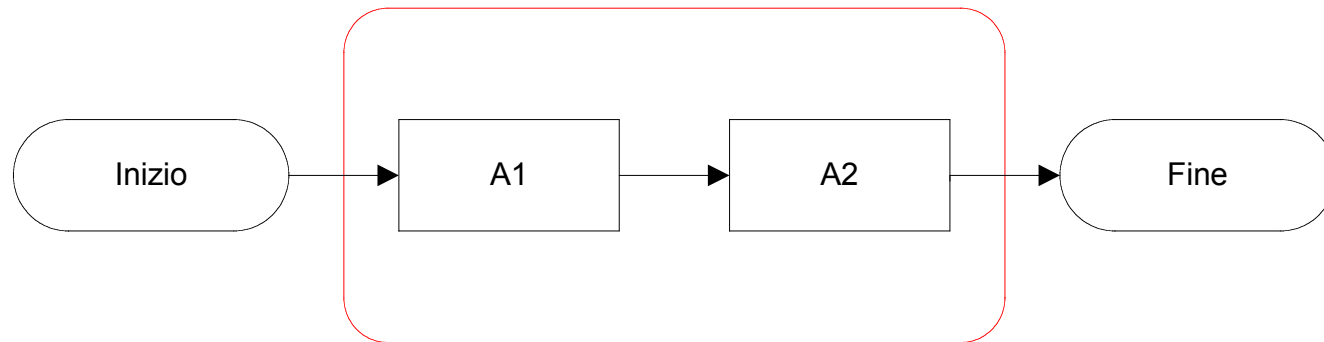


# Schemi di composizione fondamentali

- Sequenza
- Selezione
- Iterazione
  
- A partire da questi, si definiscono i *grafi di flusso strutturati*

# Sequenza

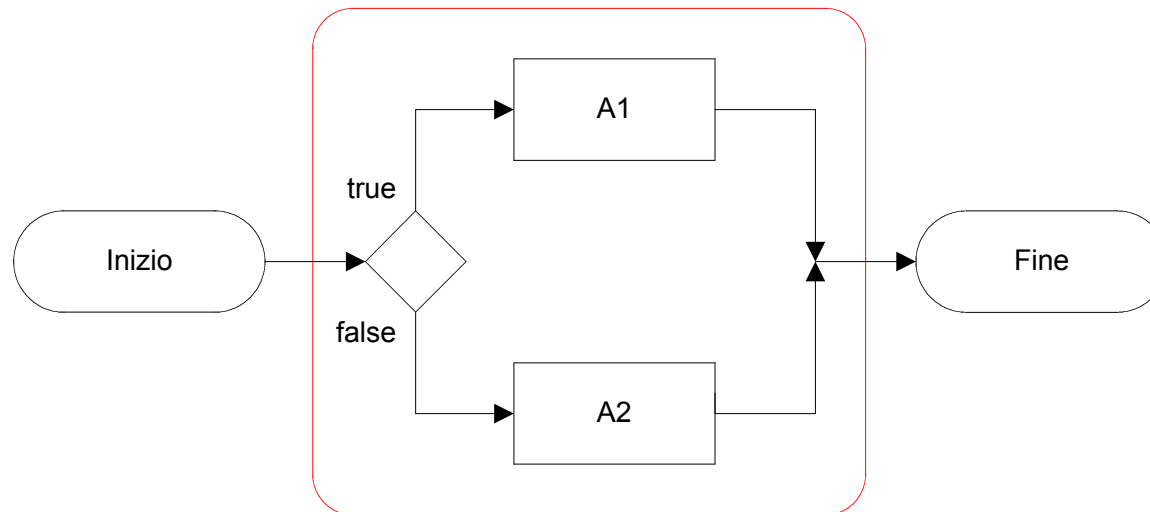
- Questo schema fondamentale di composizione descrive la sequenzialità di due azioni elementari





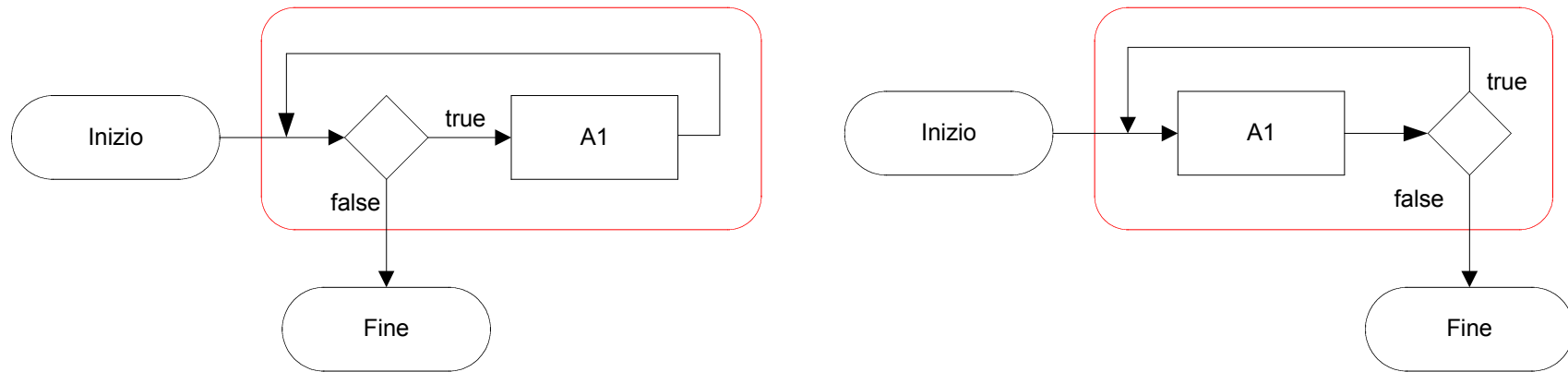
# Selezione

- Questo schema fondamentale di composizione descrive la selezione tra due alternative



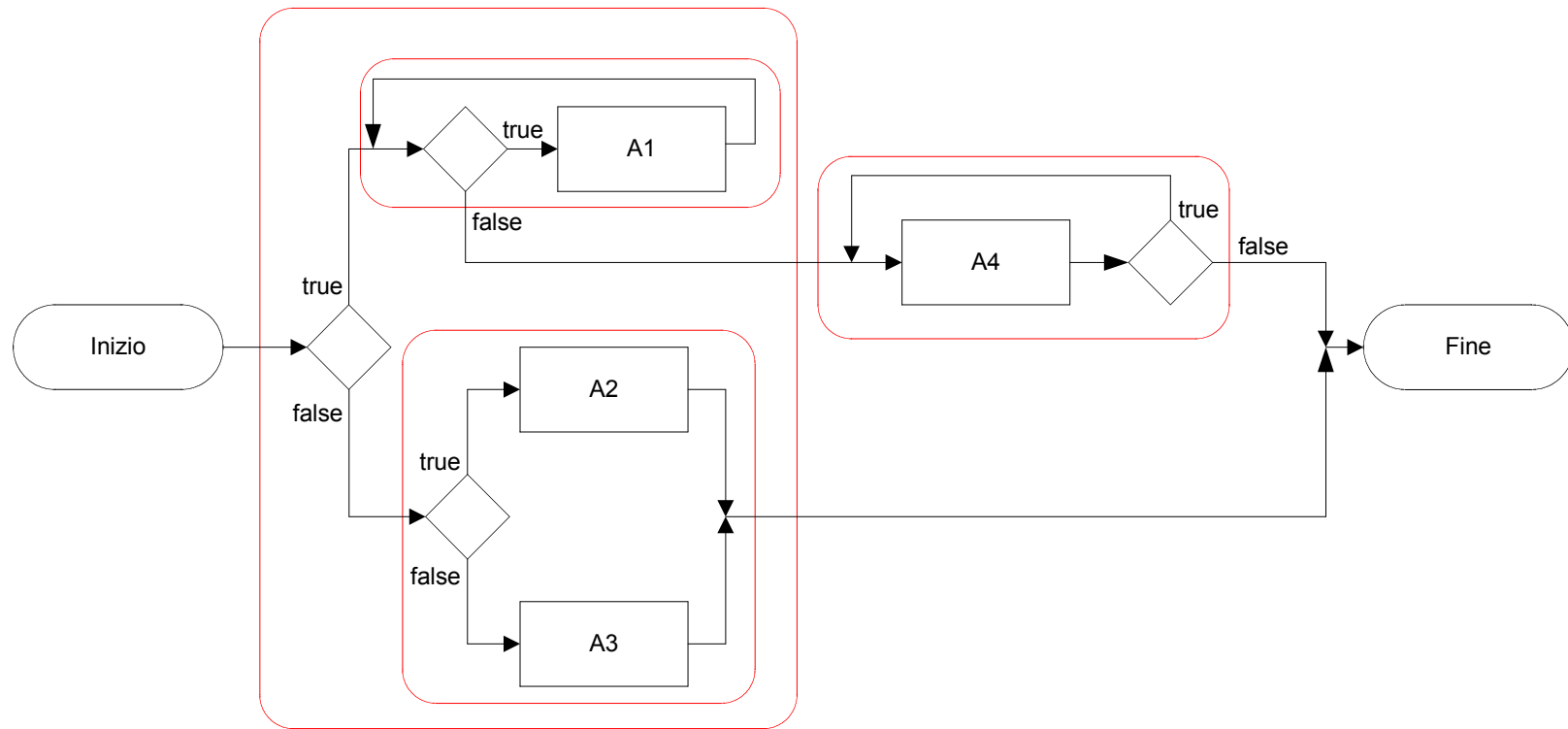
# Iterazione

- Questo schema fondamentale di composizione descrive l'iterazione (la ripetizione) di una azione quando una condizione è verificata. Esistono due modelli che si differenziano sul momento in cui viene valutata la condizione



# Grafi di flusso strutturati

- Descrivono un algoritmo componendo gli schemi fondamentali



# Esempio finale: la telefonata

- Si supponga che le azioni note siano:
  - Solleva il ricevitore
  - Attendi il segnale
  - Componi il numero
  - Attendi risposta
  - Chiacchiera
  - Deponi il ricevitore