

# Paradigma logico e funzionale

## Paradigma logico

Il paradigma di programmazione logico utilizza un approccio dichiarativo per la soluzione dei problemi. Vengono fatte delle asserzioni logiche, cioè **dichiarazioni** che forniscono una definizione del problema stabilendo i fatti conosciuti. Le dichiarazioni sono costruite attraverso le **relazioni**.

Per esempio ci possono essere dichiarazioni come le seguenti:

Mario è il padre di Luca

$$(1) \text{ padre (Luca) = Mario}$$

Luca è il padre di Angelo

$$(2) \text{ padre (Angelo) = Luca}$$

Ci può essere anche la relazione:

$$(3) \text{ nonno}(x) = \text{padre (padre (x))}$$

Poi vengono preparate le **interrogazioni**, a partire dalle quali il sistema di risoluzione, mediante passaggi logici, restituisce i risultati in output.

Per esempio, usando le dichiarazioni precedenti possiamo chiedere: *chi è il nonno di Angelo?* Attraverso passaggi logici e cercando tra le dichiarazioni, il programma restituisce il nome *Mario*.

I passaggi che portano alla soluzione cominciano dall'interrogazione che può essere espressa in questa forma:

$$\text{nonno(Angelo) = ?}$$

Il sistema di risoluzione cerca tra le dichiarazioni e le relazioni una possibile associazione. Trova un'associazione tra il membro sinistro dell'interrogazione e quello della relazione (3). Con questa relazione si può effettuare la sostituzione  $x=Angelo$  e si ottiene:

$$\text{nonno(Angelo) = padre (padre (Angelo)) = ?}$$

A questo punto si può utilizzare la dichiarazione (2) ed effettuare la seguente sostituzione:

$$\text{nonno(Angelo) = padre (padre (Angelo)) = padre (Luca) = ?}$$

L'ultimo passaggio viene effettuato con la dichiarazione (1) che permette di inferire

$$\text{nonno(Angelo) = padre (padre (Angelo)) = padre (Luca) = Mario}$$

e quindi

$$\text{nonno(Angelo) = Mario.}$$

Il vantaggio di questa modalità di programmazione consiste nella possibilità di elencare le relazioni e definire le interrogazioni, lasciando poi al sistema il compito di risolvere il problema, senza dover programmare altre procedure.

## Paradigma funzionale

Il paradigma di programmazione funzionale ha come base l'uso delle **funzioni**, intese in senso matematico, per risolvere i problemi. Le funzioni ricevono in input certi valori e restituiscono un risultato. Si può sfruttare la composizione funzionale, usando cioè una funzione (il suo risultato) come input di un'altra funzione. Il sistema che implementa il paradigma funzionale ha il compito di risolvere la funzione associata al problema e restituire il risultato. Per risolvere questa funzione dovrà valutare le varie funzioni di cui è composta.

Come esempio di programmazione funzionale, si pensi al calcolo della funzione  $(x+y)^2$ , avendo come input la sequenza  $\langle x, y \rangle$ . Per risolvere questo problema si definiscono certe funzioni che poi verranno composte tra loro per ottenere il risultato voluto.

SOMMA:  $\langle a, b \rangle = a + b$

Abbiamo definito una funzione che calcola la somma di due valori di input.

QUADRATO:  $\langle c \rangle = c * c$

Abbiamo definito una funzione che calcola il quadrato di un numero dato come input.

Sfruttando le due funzioni definite precedentemente, possiamo costruire il programma che calcola  $(x + y)^2$ .

PROGRAMMA:  $\langle x, y \rangle = \text{QUADRATO}(\text{SOMMA}(x, y))$

Avendo come input la sequenza  $\langle 3, 4 \rangle$ , il risultato si trova eseguendo `PROGRAMMA (3, 4)`.

Seguendo passo a passo l'esecuzione ed effettuando le corrette sostituzioni si ottiene:

`PROGRAMMA(3, 4) = QUADRATO(SOMMA(3, 4)) = QUADRATO(7) = 49.`

I linguaggi basati sul paradigma funzionale non sono dipendenti dalle istruzioni di assegnamento e non si basano su una sequenza di passi con cui risolvere i problemi. Questa indipendenza dall'ordine di esecuzione fa sì che i linguaggi funzionali siano particolarmente adatti per programmare i calcolatori paralleli. Tra gli svantaggi di questi linguaggi c'è la bassa efficienza.