Informatica

Informatica significa Informazione automatica. Per fornire informazione è necessario:

- comprendere la richiesta. Es. Tra quanto passa l'espresso per Torino?
- avere i dati necessari a disposizione. L'espresso per Torino passa alle 12. Sono le 11.30
- essere in grado di elaborarli. Saper calcolare $12.00 - 11,30 = 0.30$
- comunicare la risposta. Tra 30 minuti

Informazione automatica

L'informazione automatica viene fornita da una macchina che:

- per comprendere la richiesta. deve avere un dispositivo che permetta di specificare la richiesta, ad es. attraverso dei tasti
- avere i dati necessari a disposizione. scritti su un dispositivo di memorizzazione
- essere in grado di elaborarli. Saper calcolare la funzione che in base alla richiesta e ai dati memorizzati calcola la risposta
- comunicare la risposta. Avere un dispositivo su cui comunicare il risultato del calcolo.

Un esempio: la calcolatrice

- La calcolatrice
  - ha 10 tasti per specificare le cifre
  - ha 4 tasti per specificare le operazioni (funzioni) da calcolare
  - una memoria su cui mantenere i risultati parziali
  - un foglio o un display sui cui stampare (o visualizzare) il risultato
- Analogamente posso definire altre macchine che calcolano altre funzioni su altri dati ma...
- ogni macchina è limitata ad un numero finito di operazioni mentre per i dati..

Funzioni calcolabili

Sorgono spontanee alcune domande:

- Quali sono le funzioni che si possono calcolare?
- Quante sono le funzioni calcolabili?
- Quali sono le funzioni che una macchina può calcolare?
- Posso costruire una macchina per ogni funzione calcolabile?
- È possibile costruire una macchina che calcoli tutte le funzioni calcolabili?

Funzioni calcolabili

Molto prima che venissero costruito il primo calcolatore i matematici avevano studiato il dominio delle funzioni calcolabili Ω

1. Le funzioni calcolabili sono quelle per le quali esiste un modo effettivo di calcolarle (algoritmo)
2. Le funzioni calcolabili sono infinite (numerabili)
3. Si! posso costruire una macchina per ogni funzione calcolabile
4. Ma la vera cosa interessante è che posso costruire una macchina che calcoli tutte le (infinite) funzioni calcolabili: questa macchina è il moderno calcolatore.

Importanti proprietà di Ω

- contiene tutte le funzioni i cui valori possono essere calcolati in modo effettivo: non solo concepiti (Es. razionali vs. reali).
- Ogni funzione calcolabile ha una (?) descrizione finita D. Es. la funzione fattorale ha la seguente descrizione:
- D(fatt)

  fatt (0) = 1

  fatt (n+1) = (n+1) * fatt (n)

Funzione Universale $\mathcal{U}_\Omega$

- Esiste una funzione universale $\mathcal{U}_\Omega$ che calcola tutte le funzioni calcolabili: $\forall f \in \Omega$, $\forall$ descrizione $D(f)$, $\forall$ valore n nel dominio di f $\mathcal{U}_\Omega(D(f), n) = f(n)$
- $\mathcal{U}_\Omega \in \Omega$ ovvero è calcolabile
- Il calcolatore è una realizzazione di $\mathcal{U}_\Omega$ cioè calcola prendendo come dati la descrizione della funzione da calcolare D(f) e i valori (n) su cui calcolare f.

Linguaggi di programmazione

Aspetto fondamentale è la descrizione della funzione e il linguaggio in cui tale descrizione è data.

- Linguaggi per descrivere funzioni sono chiamati linguaggi di programmazione
- Sono formalismi, con sintassi e semantica formalmente definite.
- Esempi: Caml, Haskel, Prolog, C, JAVA, Pascal, linguaggi macchina
- Le descrizioni delle funzioni da calcolare sono frasi in un linguaggio di programmazione e sono detti programmi.

Linguaggi di programmazione

- I linguaggi di programmazione sono tutti equivalenti: Se ho $D_{L1}(f)$ programma nel linguaggio L1 e $D_{L2}(f)$ programma nel linguaggio L2 che calcolano la stessa funzione abbiamo:

$$D_{L1}(f) \rightarrow D_{L2}(f)$$

- La traduzione è una funzione calcolabile e quindi descrivibile con un programma, che quindi può essere eseguito da un calcolatore.

Il linguaggio macchina

Il linguaggio direttamente eseguibile dal calcolatore (detto linguaggio macchina) è un linguaggio di programmazione poco comprensibile agli umani (programmatori) che devono scrivere il programma perchè:

- le operazioni sono molto semplici,
- sono specificate in binario: sequenze di 0 e 1 quindi
- specificare un'operazione complessa richiede la scrittura di un lungo programma, incomprensibile.

Linguaggi ad alto livello

- Tutti i programmi vengono scritti in un linguaggio ad alto livello (per voi il C) e sono molto più:
  - compatti,
  - comprensibili
  - modificabili
  - estendibili
  - correggibili
- i programmi di tali linguaggi devono pertanto essere tradotti in linguaggio macchina per poter essere eseguiti.
- Fortunatamente la traduzione (come abbiamo già detto) può essere fatta dalla macchina stessa.
- I programmi che effettuano la traduzione si chiamano compilatori

Le fasi della programmazione

Ad un primo livello di astrazione l'attività della programmazione può essere suddivisa in quattro (macro) fasi principali.

1. Definizione del problema (specifica): quale funzione si vuole calcolare e quali sono i dati di interesse
2. Individuazione di un procedimento risolutivo (algoritmo)
3. Codifica dell'algoritmo in un linguaggio di programmazione (codifica)
4. Esecuzione e messa a punto (esecuzione)

Specifica

- La prima fase della programmazione consiste nel comprendere e definire (specificare) il problema che si vuole risolvere.
- La specifica del problema può essere fatta in maniera più o meno rigorosa, a seconda del formalismo descrittivo utilizzato.
- La specifica di un problema prevede la descrizione dello stato iniziale (dati iniziali, input) e dello stato finale atteso (i risultati, output).
- La caratterizzazione degli stati iniziale e finale dipende dal particolare problema in esame e dagli oggetti di interesse.

Esempi di specifica informale

1. Dati due numeri, trovare il maggiore.
2. Dato un elenco telefonico e un nome, trovare il numero di telefono corrispondente.
3. Data la struttura di una rete stradale e le informazioni sui flussi dei veicoli, determinare il percorso più veloce da A a B.
4. Il testo di un esercizio: Si scriva un programma C che stampa "Ciao Nome Cognome", dove nome e cognome sono il proprio nome e cognome.

- la descrizione (es 3) non fornisce un metodo risolutivo
- la descrizione (es 2) del problema può essere ambigua o imprecisa (se Mario Rossi compare più volte)

Algoritmi

- Una volta specificato il problema, si determina un procedimento risolutivo dello stesso (algoritmo), ovvero un insieme di azioni da intraprendere per ottenere i risultati attesi.
- Il concetto di algoritmo ha origini molto lontane: l'uomo ha utilizzato spesso algoritmi per risolvere problemi di varia natura. Solo in era moderna, tuttavia, ci si è posti il problema di caratterizzare problemi e classi di problemi per i quali è possibile individuare una soluzione algoritmica e solo nel secolo scorso è stato dimostrato che esistono problemi per i quali non è possibile individuare una soluzione algoritmica.

Algoritmi (cont.)

- Utilizziamo algoritmi nella vita quotidiana tutte le volte che, ad esempio, seguiamo le istruzioni per il montaggio di una apparecchiatura, per la sostituzione della cartuccia di una stampante, per impostare il ciclo di lavaggio di una lavastoviglie, per prelevare contante da uno sportello Bancomat, ecc.

Un algoritmo è una sequenza di passi che, se intrapresa da un esecutore, permette di ottenere i risultati attesi a partire dai dati forniti.

Proprietà di un algoritmo

La descrizione di un procedimento risolutivo può considerarsi un algoritmo se rispetta alcuni requisiti essenziali, tra i quali:

Finitezza: un algoritmo deve essere composto da una sequenza finita di passi elementari.

Eseguibilità: il potenziale esecutore deve essere in grado di eseguire ogni singola azione in tempo finito con le risorse a disposizione

Non-ambiguità: l'esecutore deve poter interpretare in modo univoco ogni singola azione.

Tipici procedimenti che non rispettano alcuni dei requisiti precedenti sono:

- le ricette di cucina: *aggiungere sale q.b.* - non rispetta 3)
- le istruzioni per la compilazione della dichiarazione dei redditi (!)

Codifica

- Questa fase consiste nell'individuare una rappresentazione degli oggetti di interesse del problema ed una descrizione dell'algoritmo in un opportuno linguaggio noto all'esecutore.
- Nel caso in cui si intenda far uso di un elaboratore per l'esecuzione dell'algoritmo, quest'ultimo deve essere tradotto (codificato) in un opportuno linguaggio di programmazione. Il risultato in questo caso è un programma eseguibile al calcolatore.
- Quanto più il linguaggio di descrizione dell'algoritmo è vicino al linguaggio di programmazione scelto, tanto più semplice è la fase di traduzione e codifica. Se addirittura il linguaggio di descrizione coincide con il linguaggio di programmazione, la fase di traduzione è superflua.

Codifica (cont.)

I linguaggi di programmazione forniscono strumenti linguistici per rappresentare gli algoritmi sottoforma di programmi che possano essere compresi da un calcolatore.

In particolare dobbiamo rappresentare nel linguaggio di programmazione

- l'algoritmo $\Longrightarrow$ programma
- le informazioni iniziali $\Longrightarrow$ dati in ingresso
- le informazioni utilizzate dall'algoritmo $\Longrightarrow$ dati ausiliari
- le informazioni finali $\Longrightarrow$ dati in uscita

In questo corso impareremo a codificare algoritmi utilizzando il linguaggio di programmazione denominato C.

Esecuzione

- La fase conclusiva consiste nella:
  - compilazione del programma: cioè la traduzione in linguaggio macchina. Questa fase porta (quasi sempre) alla luce errori di vario genere (sintattici, o di tipo ma sempre statici).
  - esecuzione vera e propria del programma. Anche questa fase può portare alla luce errori, in questo caso di semantica (errori dinamici)
- In entrambi i casi la correzione degli errori può richiedere la revisione di una o piú fasi (dalla specifica, alla definizione dell'algoritmo, alla codifica di quest'ultimo).
- Nel caso dei linguaggi di programmazione moderni, vengono forniti strumenti (denominati di solito debugger) che aiutano nella individuazione degli errori e nella messa a punto dei programmi.

Esempi

Negli esempi che seguono, utilizziamo un linguaggio pseudo-naturale per la descrizione degli algoritmi.

Tale linguaggio utilizza, tra l'altro, le comuni rappresentazioni simboliche dei numeri e delle operazioni aritmetiche.

Problema 1: Calcolo del prodotto di due interi positivi

Specifica

Input: due valori interi positivi A e B

Output: il valore di $A \times B$

Algoritmo

Se l'esecutore che scegliamo è in grado di effettuare tutte le operazioni di base sui numeri, un semplice algoritmo è il seguente:

Passo 1.	Acquisisci il primo valore, sia esso A
Passo 2.	Acquisisci il secondo valore, sia esso B
Passo 3.	Ottieni il risultato dell'operazione $A \times B$

Problema 1 (cont.)

Codifica

Un esecutore che rispetta le ipotesi precedenti è una persona dotata di una calcolatrice tascabile. La codifica dell'algoritmo in questo caso può essere allora la seguente:

Passo 1.	Digita in sequenza le cifre decimali del primo valore
Passo 2.	Digita il tasto $*$
Passo 3.	Digita in sequenza le cifre decimali del secondo valore
Passo 4.	Digita il tasto $=$

Esecuzione

Problema 1 (cont.)

- Supponiamo ora che l'esecutore scelto sia in grado di effettuare solo le operazioni elementari di somma, sottrazione, confronto tra numeri.
- È necessario individuare un nuovo procedimento risolutivo che tenga conto delle limitate capacità dell'esecutore

Algoritmo 2

Passo 1.	Acquisisci il primo valore, sia esso A
Passo 2.	Acquisisci il secondo valore, sia esso B
Passo 3.	Associa 0 ad un terzo valore, sia esso C
Passo 4.	**Finché** B>0 **ripeti**
Passo 4.1.	Somma a C il valore A
Passo 4.2.	Sottrai a B il valore 1
Passo 5.	Il risultato è il valore C

Problema 1: (cont.)

Un esecutore che rispetta le ipotesi precedenti è un bimbo in grado di effettuare le operazioni richieste (somma, sottrazione e confronto) e di riportare i risultati di semplici calcoli su un quaderno.

Codifica

Passo 1. Scrivi il primo numero nel riquadro A

Passo 2. Scrivi il secondo numero nel riquadro B

Passo 3. Scrivi il valore 0 nel riquadro C

Passo 4. Ripeti i seguenti passi finché il valore nel riquadro B è maggiore di 0:

1. calcola la somma tra il valore in A e il valore in C e e scrivi il risultato ottenuto in C
2. calcola la differenza tra il valore in B ed il numero 1 e scrivi il risultato ottenuto in B

Passo 5. Il risultato è quanto contenuto nel riquadro C.

Il concetto di stato

Gli esempi visti finora consentono di fare le seguenti considerazioni di carattere generale:

- La specifica (astratta) di un problema consiste nella descrizione di uno stato iniziale (che descrive i dati del problema) e di uno stato finale (che descrive i risultati attesi).
- È necessario individuare una rappresentazione degli oggetti coinvolti (e dunque dello stato) che sia direttamente manipolabile dall'esecutore prescelto.
- Un algoritmo è una sequenza di passi elementari che, se intrapresi da un esecutore, comportano ripetute modifiche dello stato fino al raggiungimento dello stato finale desiderato.
- Le azioni di base che l'esecutore è in grado di effettuare devono dunque prevedere, tra le altre, azioni che hanno come effetto cambiamenti dello stato.

Un'astrazione dello stato

Introduciamo una possibile astrazione del concetto di stato, di cui faremo spesso uso in seguito.

Stato

Uno stato è un insieme di associazioni tra nomi simbolici e valori.

In uno stato, ad ogni nome simbolico è associato al più un valore.

Rappresentiamo una associazione tra il nome simbolico x ed il valore val con la notazione

$$x \rightsquigarrow val$$

Lo stato: esempi

Stati corretti

- $\{$nome $\rightsquigarrow$ Antonio, cognome $\rightsquigarrow$ Rossi, eta' $\rightsquigarrow$ 25$\}$
- $\{$importo $\rightsquigarrow$ \$1650, tasso $\rightsquigarrow$ 10%, interesse $\rightsquigarrow$ \$165 $\}$
- $\{$a $\rightsquigarrow$ 25, b $\rightsquigarrow$ 3, c $\rightsquigarrow$ 50$\}$

Stati non corretti

- $\{$nome $\rightsquigarrow$ Antonio, nome $\rightsquigarrow$ Paolo, eta' $\rightsquigarrow$ 25$\}$
- $\{$b $\rightsquigarrow$ 45, a $\rightsquigarrow$ 150, b $\rightsquigarrow$ 10$\}$

La soluzione facendo riferimento allo stato

- Scriviamo la soluzione precedente facendo riferimento alle modifiche dello stato:
-

Passo 1.	Associa il primo numero ad A (nello stato)
Passo 2.	Associa il secondo numero a B (nello stato)
Passo 3.	Associa il valore 0 a C (nello stato)
Passo 4.	Ripeti finché $B > 0$:
	1. calcola $A + C$ e associa il risultato a C (nello stato)
	2. calcola $B - 1$ e associa il risultato a B (nello stato)
Passo 5.	Il risultato è il valore associato a C (nello stato).

Gli stati per il calcolo di 3×2

Stati calcolati

- Passo 1 Scrivi il primo numero in A. $\{A \rightsquigarrow 3\}$
- Passo 2 Scrivi il secondo numero in B $\{A \rightsquigarrow 3, B \rightsquigarrow 2\}$
- Passo 3 Scrivi 0 in C. $\{A \rightsquigarrow 3, B \rightsquigarrow 2, C \rightsquigarrow 0\}$
- Passo 4.1 Calcola la somma di A e C e scrivila in C (B>0). $\{A \rightsquigarrow 3, B \rightsquigarrow 2, C \rightsquigarrow 3\}$
- Passo 4.2 Calcola la differenza tra B e 1 e scrivila in B (B>0) $\{A \rightsquigarrow 3, B \rightsquigarrow 1, C \rightsquigarrow 3\}$
- Passo 4.1 Calcola la somma di A e C e scrivila in C (B>0) $\{A \rightsquigarrow 3, B \rightsquigarrow 1, C \rightsquigarrow 6\}$
- Passo 4.2 Calcola la differenza tra B e 1 e scrivila in B (B>0) $\{A \rightsquigarrow 3, B \rightsquigarrow 0, C \rightsquigarrow 6\}$
- Passo 5 $\{A \rightsquigarrow 3, B \rightsquigarrow 0, C \rightsquigarrow 6\}$

Problema del prodotto

Sintetizziamo ulteriormente:

Passo 1. leggi (A);
Passo 2. leggi (B);
Passo 3. $C = 0$;
Passo 4. Ripeti finché $(B > 0)$:
1. $\{C = A + C$;
2. $B = B - 1;\}$

Passo 5. stampa (C);

- È quasi un programma in C!

Soluzione induttiva

La soluzione precedente utilizza, per calcolare, il paradigma *procedurale*, ovvero un algoritmo iterativo. Un modo diverso di calcolare è quello che utilizza definizioni induttive delle funzioni. Ad esempio per il prodotto abbiamo:

$$\text{prodotto}\ (A, 0) \quad = 0 \qquad (1)$$
$$\text{prodotto}\ (A, B) \quad = A + \text{prodotto}(A, B-1) \text{ se } B \neq 0 \qquad (2)$$

- In questo caso non c'e' lo stato, ma la computazione avviene per *riduzione*

  $\text{prodotto}(3, 2)$

  $\text{prodotto}(3, 2) = 3 + \text{prodotto}(3, 1)$ utilizzo (2) con $A = 3$ e $B = 2$

  $3 + \text{prodotto}(3, 1) = 3 + 3 + \text{prodotto}(3, 0)$ utilizzo (2) con $A = 3$ e $B = 1$

  $3 + 3 + \text{prodotto}(3, 0) = 3 + 3 + 0$ utilizzo (1) con $A = 3$ e $B = 0$

  $3 + 3 + 0 = 6$ calcolo il risultato il modo di esprimere il calcolo è diverso.