

Cenni di computabilità

Stefano Mizzaro

Dipartimento di matematica e informatica
 Università di Udine
<http://www.dimi.uniud.it/mizzaro/>
 mizzaro@uniud.it
 Programmazione, lezione 24
 26 maggio 2015

Scaletta

- Problemi, domande, risposte
- Algoritmi, programmi, linguaggi di programmazione
- Algoritmi che calcolano funzioni
- Funzioni computabili e non

Stefano Mizzaro - Computabilità

2

Cosa fa un programma?

- Fornisce **risposte** a **domande**
 - Qual è la radice quadrata di 2345?
 - Qual è il massimo comun divisore di 234 e 36?
- Fornisce risposte a una **classe di domande**
 - Programma per calcolare la radice quadrata di un numero
- "Risolve **problemi**"

Stefano Mizzaro - Computabilità

3

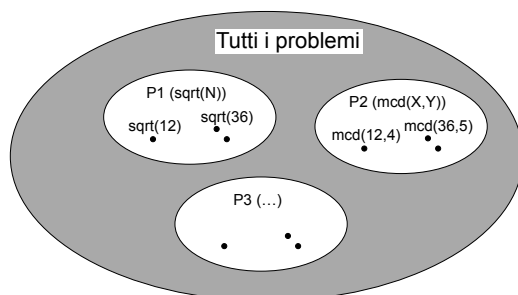
Definizioni (1/2)

- **Problema** = insieme di domande omogenee
 - Esempi:
 - P1 = Calcolare $\text{sqrt}(n)$
 - P2 = Calcolare $\text{mcd}(x,y)$
- **Domanda** ("istanza di un problema")
 - Esempi:
 - I1 = Quanto vale $\text{sqrt}(341)$?
 - I2 = Quanto vale $\text{mcd}(48,36)$?

Stefano Mizzaro - Computabilità

4

Problemi e domande



Stefano Mizzaro - Computabilità

5

Definizioni (2/2)

- **Risposta** (a una domanda) = "soluzione di un'istanza"
- **Soluzione di un problema** = metodo generale che permette di fornire in modo uniforme la risposta a tutte le istanze di un problema = **Algoritmo risolvete**

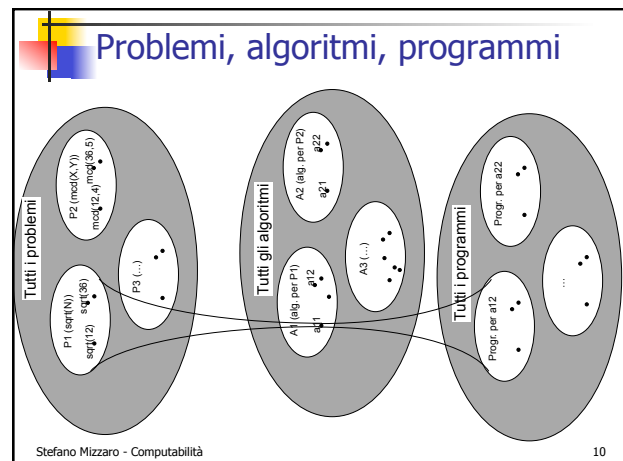
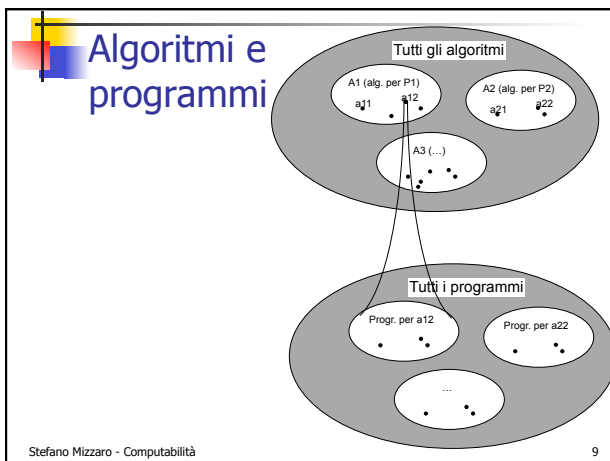
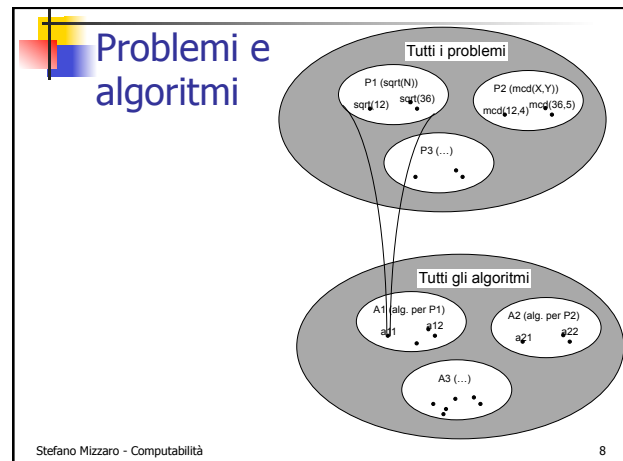
Stefano Mizzaro - Computabilità

6

Algoritmi e programmi

- **Algoritmo**
 - "Metodo", "procedimento", "sequenza di passi"
 - Astratto
 - Non si può toccare/vedere
- **Programma**
 - Rappresentazione precisa di un algoritmo in un certo **linguaggio di programmazione**
 - Per eseguire/comunicare un algoritmo bisogna rappresentarlo!!

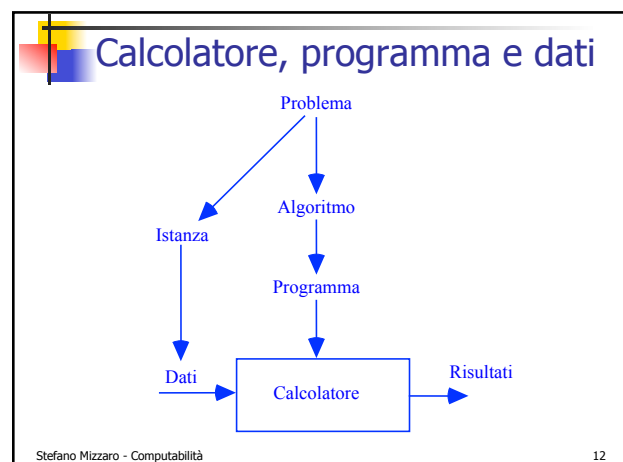
Stefano Mizzaro - Computabilità 7



Il calcolatore

- È un:
 - Esecutore di algoritmi
 - rappresentati da programmi
 - mediante un linguaggio di programmazione
- Esecutore **universale** di algoritmi
- Un algoritmo va eseguito su dati
 - Es.: calcola $\text{sqrt}(897)$

Stefano Mizzaro - Computabilità 11



Finitezza, univocità, effettività

- Un programma/algorithm deve essere:
 - **Finito** (# finito di istruzioni)


```
if (x == 0) return 1;
else if (x == 1) return 2;
else if (x == 2) return 3;
else if (x == 3) return 4;
else if (x == 4) return 5;
...
```
 - **Univoco**, non ambiguo ($x = x + \text{"un po'"})$
 - **Effettivo**: deve esistere un calcolatore in grado di eseguire le istruzioni
 - `prossimaVincitaAlLotto()`

Stefano Mizzaro - Computabilità

13

Curiosità legittime...

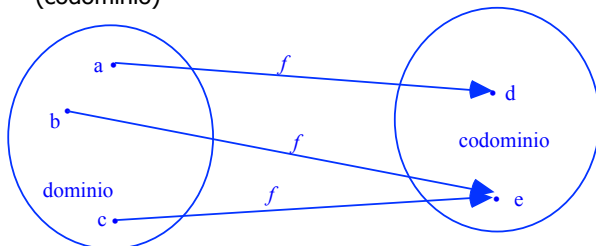
- Ma con gli algoritmi si può fare tutto?
- E se no, che cos'è che si può fare?
 - Per quali problemi esiste un algoritmo risolvibile?
- E quello che si può fare dipende dal linguaggio di programmazione scelto?
- A queste domande risponde la **teoria della computabilità (calcolabilità)**
- Approccio formale, matematico, preciso
 - Domande ancora poco precise, servono un paio di altre definizioni

Stefano Mizzaro - Computabilità

14

Funzioni

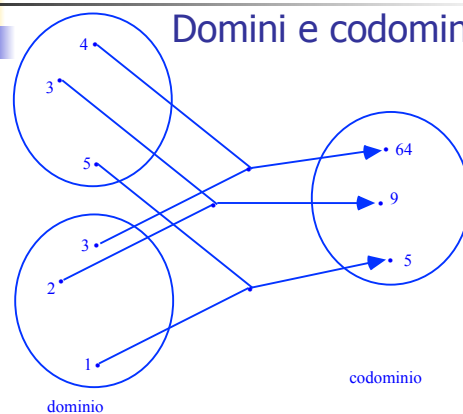
- Funzione = legge che associa a un elemento di un insieme (dominio) un elemento di un insieme (codominio)



Stefano Mizzaro - Computabilità

15

Domini e codomini



Stefano Mizzaro - Computabilità

16

Algoritmi calcolano funzioni

- Algoritmo per calcolare la radice quadrata calcola la funzione che associa a un numero la sua radice quadrata
- Algoritmo di Euclide calcola la funzione che dati 2 numeri restituisce il loro MCD
- Sembra riduttivo, ma
 - ... codifica...

Stefano Mizzaro - Computabilità

17

Quali funzioni?

- Funzioni $f: R \rightarrow R$: analisi matematica
- La teoria della computabilità si occupa delle funzioni $f: N \rightarrow N$
- Parziali
 - Ad esempio programma con un ciclo infinito che non dà nessun output
 - Funzione "non definita"

Stefano Mizzaro - Computabilità

18

Le funzioni $f: \mathbb{N} \rightarrow \mathbb{N}$

A large yellow oval labeled $f: \mathbb{N} \rightarrow \mathbb{N}$ contains four points labeled f_1 , f_2 , f_3 , and f_4 .

Stefano Mizzaro - Computabilità 19

Curiosità legittime...

- Def.: **Funzione calcolabile (computabile)** = funzione per cui esiste un algoritmo che la calcola
- Ora, finalmente, possiamo porci una domanda precisa:
Le funzioni $f: \mathbb{N} \rightarrow \mathbb{N}$ sono tutte calcolabili?
- Data una qualsiasi funzione, esiste sempre un algoritmo che la calcola?
 - MCD: ok. Somma: ok. Radice quadrata:...ok.
 - Ma in generale?

Stefano Mizzaro - Computabilità 20

No! Funzioni calcolabili e non

A large yellow oval labeled $f: \mathbb{N} \rightarrow \mathbb{N}$ contains a smaller grey oval labeled 'Funzioni calcolabili'. The grey oval contains points f_1 , f_2 , f_3 , and f_4 . A label 'Funzioni non calcolabili' with an arrow points to the yellow area outside the grey oval.

Stefano Mizzaro - Computabilità 21

No? Voglio un esempio...

- Il problema della terminazione (*Halting problem*)
 - Trovare un algoritmo A in grado di dirci se un qualsiasi algoritmo A_i su dati D termina.
- (un algoritmo potrebbe non terminare perché entra in un ciclo infinito)
 - CiaoATutti termina
 - while(true) non termina

Stefano Mizzaro - Computabilità 22

Il problema della terminazione

```

    graph LR
        A[Descrizione di un algoritmo A_i] --> C[Calcolatore]
        D[Dati di ingresso D] --> C
        C --> O[Termina / Non termina]
        A2[Algoritmo A?] --> C
    
```

Stefano Mizzaro - Computabilità 23

Una soluzione?

- Beh, potrei far eseguire A_i sui dati e attendere
- Se termina so che termina
- Se dopo 2 ore non ha terminato... so che non ha terminato in due ore... ma quanto devo aspettare?!
- Non va!
- Teorema: A non esiste

Stefano Mizzaro - Computabilità 24

1. La programmazione strutturata (22h)

- L'ABC della programmazione:
 - Espressioni
 - Strutture di controllo
 - Sviluppo incrementale
 - Array
 - Metodi
 - Ricorsione
- Presenti in ogni linguaggio di programmazione

Stefano Mizzaro - Computabilità

31

2. Tipi di dati astratti, occultamento delle informazioni (4h)

- Come fare programmi più grandi e complicati...
- ...e comunque modificabili, estendibili, comprensibili, ...
- Tipi di dati astratti, occultamento delle informazioni
- Concetti e realizzazione in Java
- Verso l'OO ("Object Oriented")

Stefano Mizzaro - Computabilità

32

3. I fondamenti della programmazione OO (10h)

- Evoluzione dai TDA: oggetti, classi, istanze, scambio messaggi
- Eredità
- Polimorfismo
- La programmazione OO in Java
 - Classi astratte
 - Interfacce
 - Package

Stefano Mizzaro - Computabilità

33

4. Le API del Java (10h)

- Application Programming Interface
- "Librerie"
- Il programmatore di oggi deve conoscere anche le API
- Introduzione, cenni:
 - File, stringhe, utilità, ...
 - Interfacce utente grafiche
 - Applet

Stefano Mizzaro - Computabilità

34

5. Teorie della computabilità e della complessità (cenni, 2h)

- Teoria matematiche (teoremi!)
- Computabilità
 - Cos'è un programma?
 - "Cosa può fare un programma"?
 - È possibile scrivere un programma che ...?
 - Ci sono linguaggi di programmazione più "potenti" di altri? Cosa vuol dire "più potente"?
- Complessità
 - Dato un problema, ci sono programmi più efficienti di altri per risolverlo? (più veloci, che usano meno risorse, ...)

Stefano Mizzaro - Computabilità

35