

**Recall: Obiettivi del corso**

In questo corso vediamo tre paradigmi diversi:

- Paradigma imperativo esemplificativo di linguaggi come C, C++, C#, Java, PHP, Python e Ruby
- Paradigma funzionale esemplificativo di linguaggi come ML, Miranda, OCAML, Haskell e F#
- Paradigma concorrente esemplificativo di linguaggi come Erlang, Scala, Go e Rust

Di ogni paradigma studiamo la/le semantiche

La semantica formale assegna un significato rigoroso e non ambiguo ai programmi: dice ai programmatori il significato del codice che scrivono (ad un certo livello di astrazione)

Vantaggi:

Ci da un modello di riferimento del linguaggio

Ci permette analisi formale delle proprietà del linguaggio, posso tentare di provare che termina che non ha errori....

Ci fornisce un'implementazione automatica del back-end del compilatore

# Quali problemi possiamo risolvere ?

Come dimostrare che il programma abbia un certo comportamento?

Come dimostrare che il codice semplificato sia corretto (per esempio dopo una compilazione)?

Come dimostrare l'assenza di problemi (per esempio nel mio programma non si verifica mai una divisione per 0)?

Come produrre codice affidabile?

Come provare la correttezza di un'implementazione?

Come definire i risultati corretti dei casi di test?

Come individuare precocemente ambiguità, anomalie, incoerenze?

Come esporre le debolezze?...

# Non e' facile capire cosa fa un programma solo guardando il codice

## Cosa calcolano questi programmi?

$k := 1;$

$r := x;$

while  $k > 0$  do

  if  $r > 100$  then

$r := r - 10;$

$k := k - 1$

  else

$r := r + 11;$

$k := k + 1$

in el [ ] prec = [el]

in el (x:xs) prec = if (prec el x)

  then (el:(x:xs))

  else (x:(in el xs prec))

ins [ ] prec = [ ]

ins (x:xs) prec = (in x (ins xs prec) prec)

# E se non abbiamo un modello di cosa fa un programma come possiamo evitare che produca errori?



SOFTWARE BUGS IN HISTORY

**Mars Climate  
Orbiter Disassembly**



SOFTWARE BUGS IN HISTORY

**The Ariane 5 Disaster**



SOFTWARE BUGS IN HISTORY

**Therac-25**



SOFTWARE BUGS IN HISTORY

**Losing \$460m in 45 minutes**