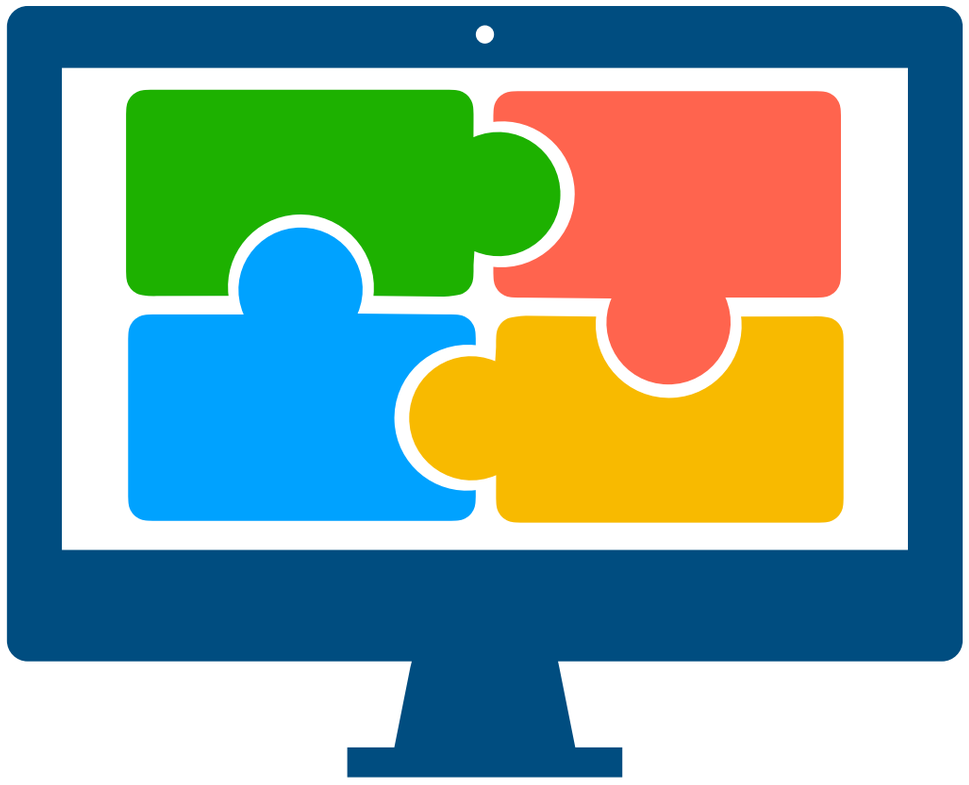


Linguaggi di Programmazione

Roberta Gori
Vincenzo Ciancia

Presentazione

Orario

Mercoledì: 11:00-12,45, H Lab

Mercoledi': 14-15,45 aula G

Giovedì: 9:00-10,45 aula N

Venerdì: 16:00-17:45 aula E1

I prof

Roberta Gori
Dipartimento di Informatica
roberta.gori@unipi.it

Vincenzo Ciancia
Istituto di Scienza e Tecnologie
dell'Informazione "Alessandro Faedo"
vincenzo.ciancia@isti.cnr.it

Obbiettivo del corso

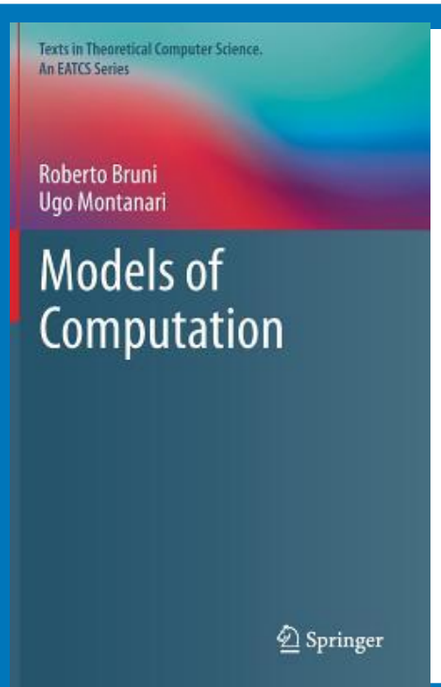
- presentarvi diverse modelli di calcolo:
  - linguaggi imperativi,
  - linguaggi funzionali di ordine superiore,
  - linguaggi concorrenti

- i loro paradigmi di programmazione,

- le loro descrizioni matematiche, sia concrete che astratte,

- alcuni strumenti/tecniche intellettuali per ragionare sui modelli.

Il nostro libro



Roberto Bruni and Ugo Montanari
Models of Computation
Texts in Theoretical Computer Science (an EATCS series)
https://www.springer.com/book/9783319428987

Informazioni utili

- Pagina web, mettero' li' tutte le slides del corso

http://didawiki.di.unipi.it/doku.php/matematica/lp/start#linguaggi_di_programmazione_e_laboratorio_aa_2021202guaggi-Compilatori2021

- Potete accederci piu' facilmente navigando attraverso le nostre pagine web

Cercate di partecipare!

cerchiamo di trovare insieme le risposte

```
% find the least (non-unitary) divisor p of n>0
p := 0;
x := 2;
while (.....) do {
    if ( n%x == 0 ) then {
        p := x;
    } else {
        x := x+1;
    }
}
```

Cercate di partecipare!

Correggetemi se sbaglio

```
% find the index of the last occurrence of n in a
i := length(a)-1;
while ( i>0 && n!=a[i] ) do {
    i := i-1;
}
```

Esame

- Prova scritta (obbligatoria)
- Prova orale (facoltativa)

Prerequisiti

Teoria degli insiemi

$\varnothing$ $A \cap B$ $A \cup B$ $A \setminus B$ $\overline{A}$

$a \in A$ $A \subset B$ $A \subseteq B$ $A \times B$

$a \notin A$ $A \not\subseteq B$ $A \cap B = \emptyset$

$\mathbb{N}$ $\mathbb{Z}$ $\mathbb{Q}$ $\mathbb{R}$ $\mathbb{B}$

$N \subseteq \mathbb{N}$ $N \in \wp(\mathbb{N})$ $S \subseteq \wp(\mathbb{N})$

Prerequisiti

teoria degli insiemi: funzioni, relazioni

$f : A \to B$

$R \subset A \times B$

funzioni come relazioni

$$R_f \triangleq \{(a, f(a)) \mid a \in A\}$$

insiemi come funzioni (funzioni come relazioni)

$$f_N : \mathbb{N} \to \mathbb{B}$$

$$f_N(n) \triangleq \begin{cases} 1 & n \in N \\ 0 & \text{otherwise} \end{cases}$$

$$N = \{n \mid f_N(n) = 1\}$$

Prerequisiti

Logica del primo ordine

ff false 0 F tt true 1 T $P \wedge Q$ $P \vee Q$ $\neg P$

$\exists x.\ P(x)$ $\forall x.\ P(x)$ $P \Rightarrow Q$ $P \Leftrightarrow Q$

significato dell'implicazione!

$$P \Rightarrow Q$$

$$Q \vee \neg P$$

$$\neg Q \Rightarrow \neg P$$

importanza dell'ordine dei quantificatori!

$$\forall n \in \mathbb{N}.\ \exists m \in \mathbb{N}.\ n < m$$

$$\exists m \in \mathbb{N}.\ \forall n \in \mathbb{N}.\ n < m$$

Prerequisiti

Stringhe e grammatiche libere

Alfabeto A

$$A^n \triangleq \underbrace{A \times \cdots \times A}_{n} \qquad A^* \triangleq \bigcup_{n \in \mathbb{N}} A^n$$

$$\begin{aligned}
\mathbb{B} &= \{0, 1\} \\
\mathbb{B}^0 &= \{\epsilon\} \\
\mathbb{B}^1 &= \{0, 1\} \\
\mathbb{B}^2 &= \{00, 01, 10, 11\} \\
\mathbb{B}^3 &= \{000, 001, 010, 011, 100, 101, 110, 111\} \\
&\ldots \\
\mathbb{B}^* &= \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}
\end{aligned}$$

Prerequisiti

Stringhe e grammatiche libere

Alfabeto A $\qquad A^n \triangleq \underbrace{A \times \cdots \times A}_{n} \qquad A^* \triangleq \bigcup_{n \in \mathbb{N}} A^n$

$$\mathbb{B}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$$

$$\mathsf{A} ::= \epsilon \mid 0\ \mathsf{A} \mid 1\ \mathsf{B}$$

$$\mathsf{B} ::= 0\ \mathsf{B} \mid 1\ \mathsf{A}$$

$$\underline{\mathsf{A}} \rightarrow \underline{0\ \mathsf{A}} \rightarrow 0\ 1\ \underline{\mathsf{B}} \rightarrow 0\ 1\ \underline{1\ \mathsf{A}} \rightarrow 0\ 1\ 1\ \epsilon = 0\ 1\ 1$$

$\mathcal{L}(\mathsf{A}) = ?$ $\qquad$ $\mathcal{L}(\mathsf{B}) = ?$

Prerequisiti

Definizioni ricorsive e induttive

$$0! \triangleq 1$$
$$(n+1)! \triangleq n! \cdot (n+1)$$

$$A^0 \triangleq \{\epsilon\}$$
$$A^{(n+1)} \triangleq A \times A^n$$

$$f(n) \triangleq \begin{cases} 1 & \text{if } n \leq 1 \\ f(n/2) & \text{if } n \neq 0 \wedge n\%2 = 0 \\ f(3n+1) & \text{otherwise} \end{cases}$$

Congettura di Collatz

$$f(12) = f(6) = f(3) = f(10) = f(5) = f(16) = f(8) = f(4) = f(2) = f(1) = 1$$

Prerequisiti

Congetture vs teoremi

un numero naturale p è primo
se non può essere scritto come il prodotto di due numeri più piccoli

n	Is n prime?	$2^n - 1$	Is $2^n - 1$ prime?
2	yes	3	yes
3	yes	7	yes
4	no: $4 = 2 \cdot 2$	15	no: $15 = 3 \cdot 5$
5	yes	31	yes
6	no: $6 = 2 \cdot 3$	63	no: $63 = 7 \cdot 9$
7	yes	127	yes
8	no: $8 = 2 \cdot 4$	255	no: $255 = 15 \cdot 17$
9	no: $9 = 3 \cdot 3$	511	no: $511 = 7 \cdot 73$
10	no: $10 = 2 \cdot 5$	1023	no: $1023 = 31 \cdot 33$

Prerequisiti

Congetture vs teoremi

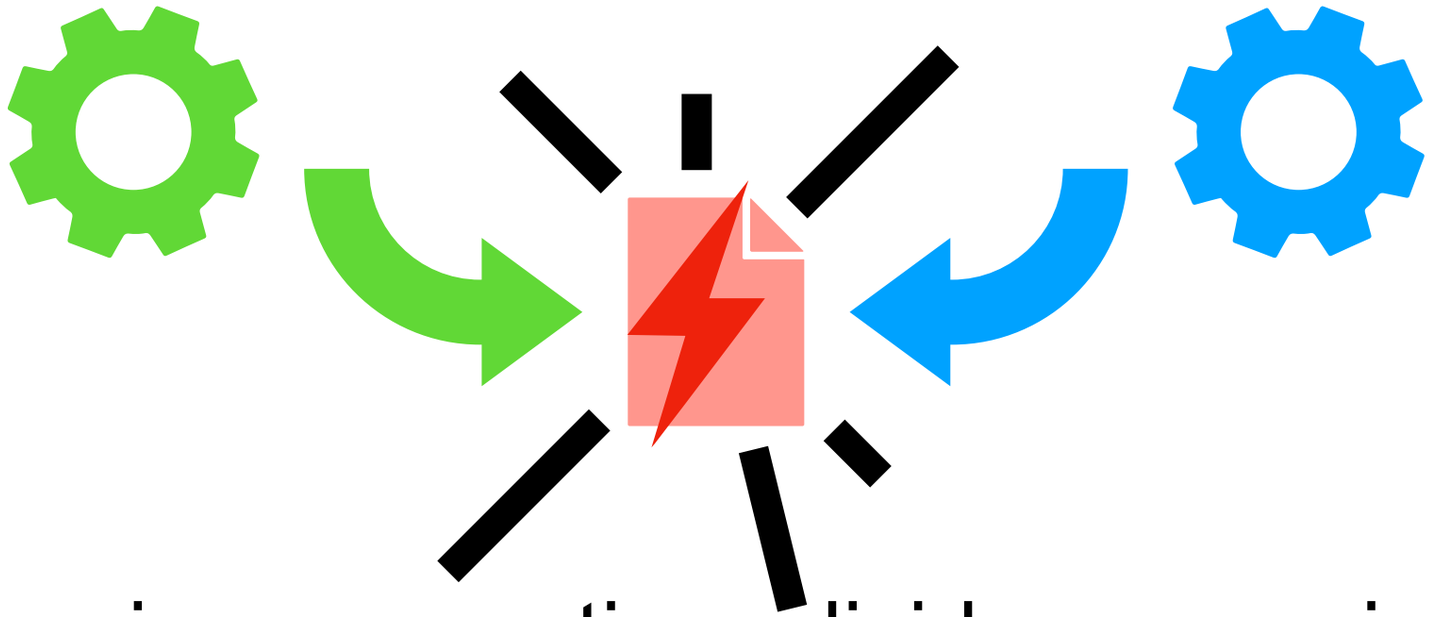
if p is prime
then $2^p - 1$ is prime

if $n > 1$ is not prime
then $2^n - 1$ is not prime

Usate qualsiasi mezzo per provare o confutare le congetture di cui sopra

Un antipasto

Il problema



Due processi concorrenti condividono una risorsa che possono usare solo uno alla volta

Possono comunicare usando la memoria condivisa

Vogliamo garantire che non ci siano conflitti quando i processi accedono alla risorsa

Non vogliamo imporre una rigida alternanza di turni senza motivo

Algoritmo di mutua esclusione di Peterson (1981)

```
% Two processes P1, P2
% Two boolean variables b1, b2 (both initially false)
% when Pi wants to enter the critical section, then it sets bi to true
% An integer variable k, taking values in {1,2}
% (initial value is arbitrary)
% the process Pk has priority over the other process
%
% Process P1 in pseudocode
```

```
while (true) {
    ...                          % non critical section
    b1 := true ;                 % P1 wants to enter the critical section
    k := 2 ;                     % P1 gives priority to the other process
    while (b2 && k==2) skip ;    % P1 waits its turn
    <critical section>           % P1 enters the critical section
    b1 := false                  % P1 leaves the critical section
}
```

```
% Process P2 is analogous to P1
```

Domande

L'algoritmo di Peterson funziona?

Cosa significa "funziona"? Cosa ci aspettiamo?

(Progresso)

Se la risorsa è disponibile, nessun processo è costretto ad aspettare

(Attesa limitata)

Nessun processo aspetterà per sempre la risorsa (*altrimenti la soluzione più semplice è che nessuno entri*)

(Mutua Esclusione)

P1 e P2 non sono mai nella sezione critica allo stesso tempo

Algoritmo di mutua esclusione di Hyman (1966)

```
% Two processes H1, H2
% Two boolean variables b1, b2 (both initially false)
% when Hi wants to enter the critical section, then it sets bi to true
% An integer variable k, taking values in {1,2}
% (initial value is arbitrary)
% the process Hk has priority over the other process
%
% Process H1 in pseudocode
```

```
while (true) {
    ...                          % non critical section
    b1 := true ;                 % H1 wants to enter the critical section
    while (k==2) {               % while H2 has priority
        while (b2) skip ;        % H1 waits
        k := 1;                  % H1 sets priority to itself
    }
    <critical section>           % H1 enters the critical section
    b1 := false                  % H1 leaves the critical section
}
```

```
% Process H2 is analogous to H1
```

The question

L'algoritmo di Peterson soddisfa la mutua esclusione?

L'algoritmo di Hyman soddisfa la mutua esclusione?

Per le risposte siate pazienti e aspettate le lezioni di fine corso