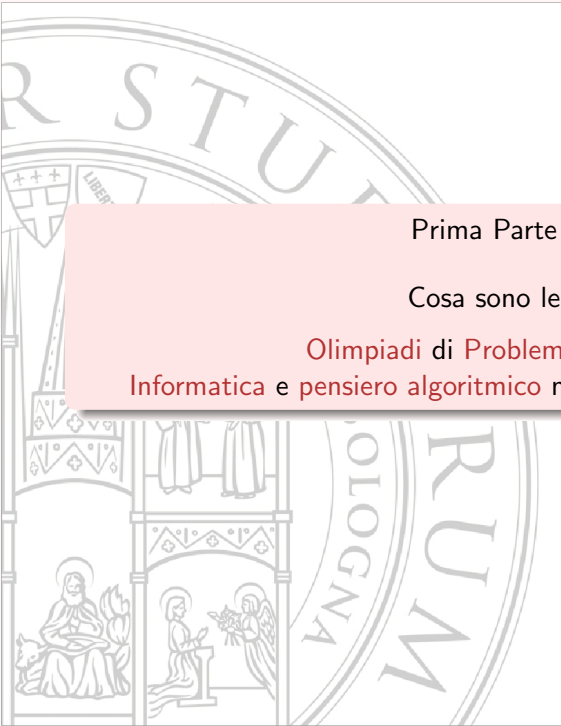


Problem Solving Una Strategia di Miglioramento dell'Apprendimento

Simone Martini

Dipartimento di Informatica – Scienza e Ingegneria
Alma Mater Studiorum • Università di Bologna


Firenze, 30 gennaio 2015



Prima Parte

Cosa sono le

Olimpiadi di **Problem Solving**
Informatica e **pensiero algoritmico** nella scuola dell'obbligo

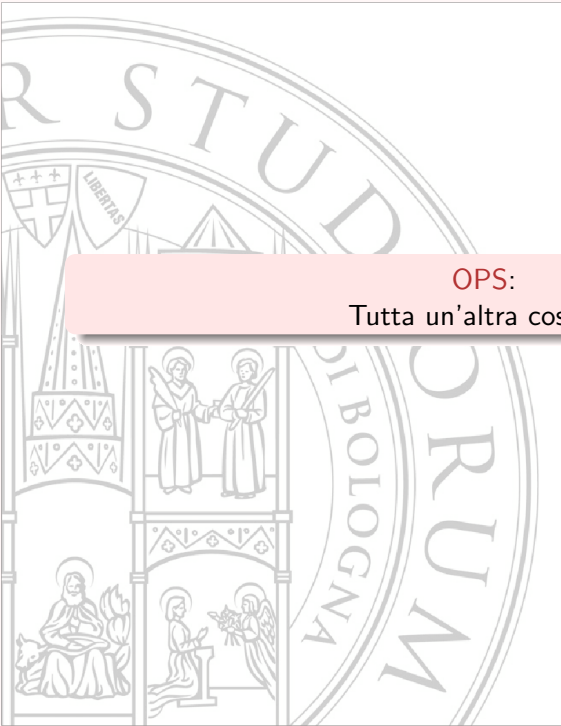


Olimpiadi di Problem Solving
Informatica e pensiero algoritmico nella scuola dell'obbligo

Olimpiadi

In genere s'intende:

- competizione
- rivolta a singoli
- per promuovere l'eccellenza
- tra i 16 e i 18 anni



OPS:

Tutta un'altra cosa...

Le “nostre” Olimpiadi:

OPS

- competizione: attività didattiche
- rivolta a singoli: a squadre
- per promuovere l'eccellenza: per tutti
- tra i 16 e i 18 anni: per tutta la scuola dell'obbligo (8-16), su tre livelli

Le “nostre” Olimpiadi:

OPS

- **competizione:** attività didattiche
- rivolta a singoli: a squadre
- per promuovere l'eccellenza: per tutti
- tra i 16 e i 18 anni: per tutta la scuola dell'obbligo (8-16), su tre livelli

Le “nostre” Olimpiadi:

OPS

- **competizione:** attività didattiche
- **rivolta a singoli:** a squadre
- **per promuovere l'eccellenza:** per tutti
- **tra i 16 e i 18 anni:** per tutta la scuola dell'obbligo (8-16),
su tre livelli

Le “nostre” Olimpiadi:

OPS

- ~~competizione~~: attività didattiche
- ~~rivolta a singoli~~: a squadre
- ~~per promuovere l'eccellenza~~: per tutti
- ~~tra i 16 e i 18 anni~~: per tutta la scuola dell'obbligo (8-16),
su **tre livelli**

Le “nostre” Olimpiadi:

OPS

- ~~competizione~~: attività didattiche
- ~~rivolta a singoli~~: a squadre
- ~~per promuovere l'eccellenza~~: per tutti
- ~~tra i 16 e i 18 anni~~: per tutta la scuola dell'obbligo (8-16),
su **tre** livelli

Attività didattiche

OPS

- attività (“giochi”, sfide) locali sui tre livelli
- prove erogate su web, da server centrale (tre/quattro volte l’anno)
- con attività didattiche di supporto
- seguite da una competizione
- una squadra per scuola, per ogni livello (“fase regionale”)
- la migliore di ogni regione (più recuperi), alla finalissima a Roma (presenza fisica)

Attività didattiche

OPS

- attività (“giochi”, sfide) locali sui tre livelli
- prove erogate su web, da server centrale (tre/quattro volte l’anno)
- con attività didattiche di supporto
- **seguite** da una competizione
- una squadra per scuola, per ogni livello (“fase regionale”)
- la migliore di ogni regione (più recuperi), alla finalissima a Roma (presenza fisica)

A squadre

OPS

- quattro persone
- possibilmente dei due sessi (**gender neutrality**)
- il tempo assegnato per una prova (90min)
- **non** è sufficiente ad un singolo
- **sono necessari** pianificazione, divisione dei compiti, organizzazione

Pianificazione, divisione dei compiti, organizzazione:
sono già problem solving!

A squadre

OPS

- quattro persone
- possibilmente dei due sessi (**gender neutrality**)
- il tempo assegnato per una prova (90min)
- **non** è sufficiente ad un singolo
- **sono necessari** pianificazione, divisione dei compiti, organizzazione

Pianificazione, divisione dei compiti, organizzazione:
sono già problem solving!

A squadre

OPS

- quattro persone
- possibilmente dei due sessi (**gender neutrality**)
- il tempo assegnato per una prova (90min)
- **non** è sufficiente ad un singolo
- **sono necessari** pianificazione, divisione dei compiti, organizzazione

Pianificazione, divisione dei compiti, organizzazione:
sono già problem solving!

Per tutti

OPS

- ogni scuola si iscrive con quattro squadre, per ogni livello (...)
- le squadre non sono blindate
- la selezione della squadra “regionale” è del coordinatore locale
- **Scopo:** stimolare abilità algoritmiche e di problem solving per tutti gli studenti, anche non partecipanti
- attività della classe, non della squadra

Per tutti

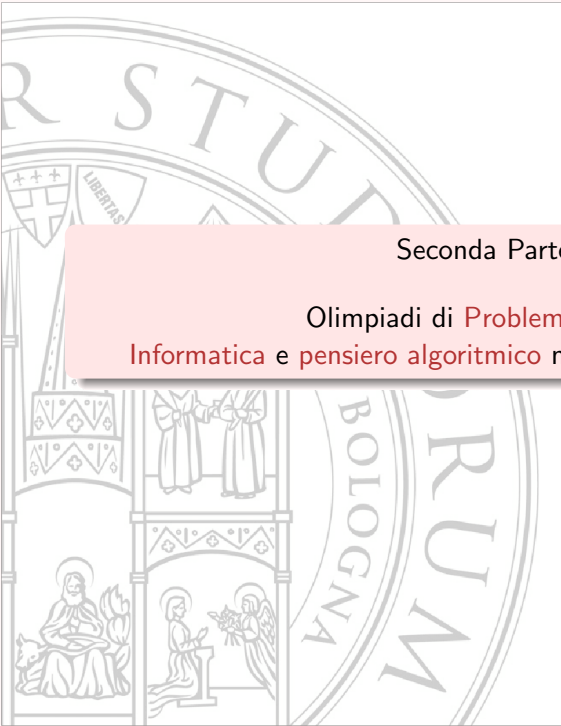
OPS

- ogni scuola si iscrive con quattro squadre, per ogni livello (...)
- le squadre non sono blindate
- la selezione della squadra “regionale” è del coordinatore locale
- **Scopo:** stimolare **abilità algoritmiche e di problem solving** per tutti gli studenti, anche **non partecipanti**
- attività della classe, non della squadra

Per la scuola dell'obbligo

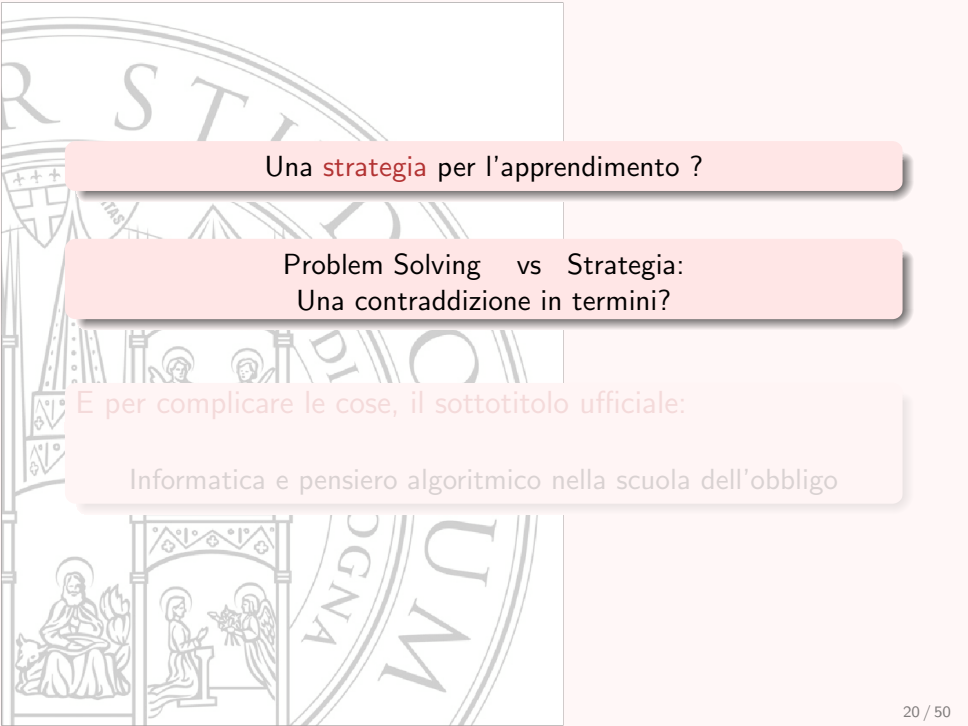
OPS

- tre livelli
- stesso **tipo** di prove
- differenze in **dimensione** o **astrattezza**



Seconda Parte

Olimpiadi di **Problem Solving**
Informatica e **pensiero algoritmico** nella scuola dell'obbligo

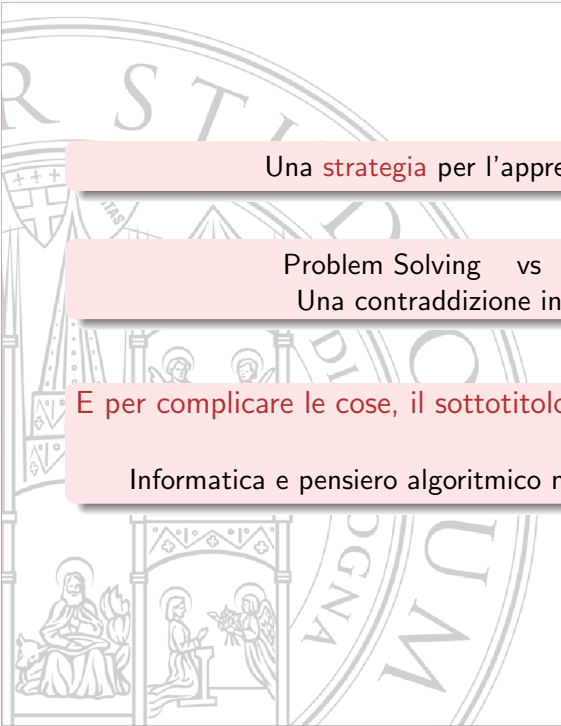


Una **strategia** per l'apprendimento ?

Problem Solving vs Strategia:
Una contraddizione in termini?

È per complicare le cose, il sottotitolo ufficiale:

Informatica e pensiero algoritmico nella scuola dell'obbligo



Una **strategia** per l'apprendimento ?

Problem Solving vs Strategia:
Una contraddizione in termini?

E per complicare le cose, il sottotitolo ufficiale:

Informatica e pensiero algoritmico nella scuola dell'obbligo

Prove


- ottimizzazione
- pianificazione
- ordinamento
- produzione di testi (e.g., testi “bucati”)
- comprensione di semplici procedure in pseudo-codice
- italiano e inglese

Prove, 2

- Zaini, Bin Packing
- Testi bucati
- Planning, GANT
- Cammini (su grafi, scacchiere ecc.)
- Comprensione di (semplici?) strutture di programmazione

Prove, 2

- Zaini, Bin Packing
- Testi bucati
- Planning, GANT
- Cammini (su grafi, scacchiere ecc.)
- Comprensione di (semplici?) strutture di programmazione

The background of the slide features a large, faint watermark of the official seal of the University of Bologna. The seal is circular and contains the Latin text 'R STUD' at the top and 'DI BOLOGNA' and 'ORUM' at the bottom. In the center, there is a shield with three crosses and the word 'LIBERTAS'. Below the shield, there are two figures in the upper section and two figures in the lower section, all within a decorative architectural frame.

Dov'è la strategia in tutto questo guazzabuglio?

Al più, regolarità

- Dall'esempio al caso generale
- Dal semplice al complesso
- Dal regolare all'irregolare

In primo luogo

- Tecniche trasversali
- Tecniche esportabili da un contesto ad un altro
- Tecniche generali

Esempio semplice

- Testo bucato:
 - competenza linguistica
 - saper enumerare *con metodo*
- Generare tutte le combinazioni
 - saper enumerare *con metodo*

Tecniche generali

- Cosa distingue una tecnica generale dalle sue specifiche applicazioni?
- Il suo livello di astrazione:
- descrive non una applicazione, ma una classe di esse
- si riferisce alle strutture generali del problema e non ai suoi “accidenti”

Tecniche generali

- Cosa distingue una tecnica generale dalle sue specifiche applicazioni?
- Il suo **livello di astrazione**:
- descrive non una applicazione, ma una classe di esse
- si riferisce alle strutture generali del problema e non ai suoi “accidenti”

Da tattiche a strategie


- apprendere una tecnica
- saperla descrivere
- saperla esportare ad altro contesto, per analogia
- saperla descrivere nella sua generalità



Ma bisogna saperla esprimere!

Individuare e “*nominare*”:

- la trama algoritmica
- la regolarità nascosta
- la possibilità di gestire la complessità
- risultati quantificabili

The background of the slide features a large, light-colored watermark of the official seal of the University of Bologna. The seal is circular and contains the Latin text 'R STUD' at the top and 'DI BOLOGNA' and 'ORUM' at the bottom. In the center, there are several figures: a seated figure on the left, two standing figures in the middle, and another seated figure on the right. A shield with a cross and three stars is visible at the top left of the seal's inner circle.

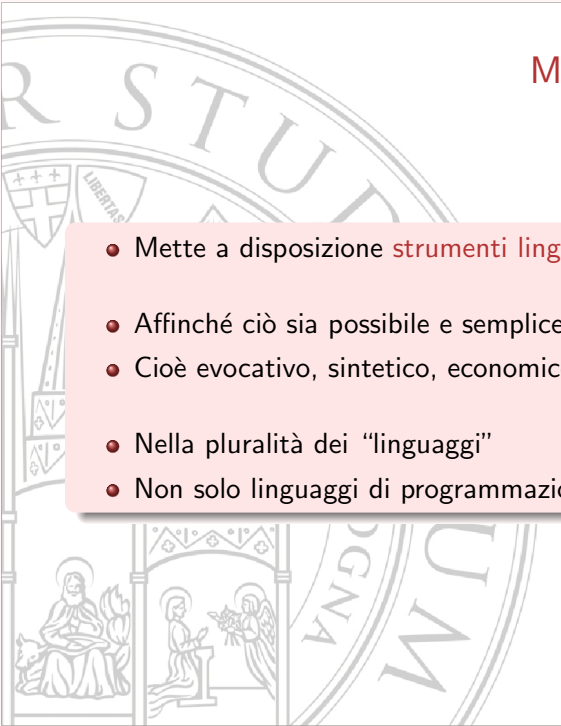
Occorre un linguaggio. . .

Informatica è...

- Un insieme di **applicazioni**
- Una **tecnologia** che rende possibili quelle applicazioni
- Una **scienza** che fonda quella tecnologia

Informatica:

- Studia i procedimenti **effettivi** di elaborazione dell'**informazione**.
- Contribuisce alle scienze con concetti propri, quali:
 - effettività
 - complessità computazionale
 - gerarchia di astrazione
 - informazione
- Si basa sulla capacità di **risolvere problemi**
- Inquadrandoli in teorie generali
- Ed esprimendo la soluzione in linguaggi specifici



Ma soprattutto...

- Mette a disposizione **strumenti linguistici**
- Affinché ciò sia possibile e semplice
- Cioè evocativo, sintetico, economico
- Nella pluralità dei “linguaggi”
- Non solo linguaggi di programmazione!



Janvier 1751.

ENCYCLOPÉDIE,
OU
DICTIONNAIRE RAISONNÉ
DES SCIENCES,
DES ARTS ET DES MÉTIERS,
RECUEILLI
DES MEILLEURS AUTEURS.
ET PARTICULIÈREMENT
DES DICTIONNAIRES ANGLAIS
DE CHAMBERS, D'HARRIS, DE DYCHE, &c.
PAR UNE SOCIÉTÉ DE GENS DE LETTRES.

Mis en ordre & publié par M. *DIDEROT*; & quant à la **PARTIE MATHÉMATIQUE,**
par M. *D'ALEMBERT*, de l'Académie Royale des Sciences de Paris
& de l'Académie Royale de Berlin.

Tantum series juncturaque polles,
Tantum de medio sumptis accedit honoris! HORAT.

DIX VOLUMES INFOLIO,
DONT DEUX DE PLANCHES EN TAILLE-DOUCE,

PROPOSÉS PAR SOUSCRIPTION.



A PARIS, Chez }
BRIASSON, rue Saint Jacques, à la Science.
DAVID l'aîné, rue Saint Jacques, à la Plume d'Or.
LE BRETON, Imprimeur ordinaire du Roy, rue de la Harpe.
DURAND, rue Saint Jacques, à Saint Landry, & au Griffon.

M. D. C. C. L. I.
AVEC APPROBATION, ET PRIVILEGE DU ROY.

On s'est adressé aux plus habiles de Paris et du royaume. On s'est donné la peine d'aller dans leurs ateliers [...]

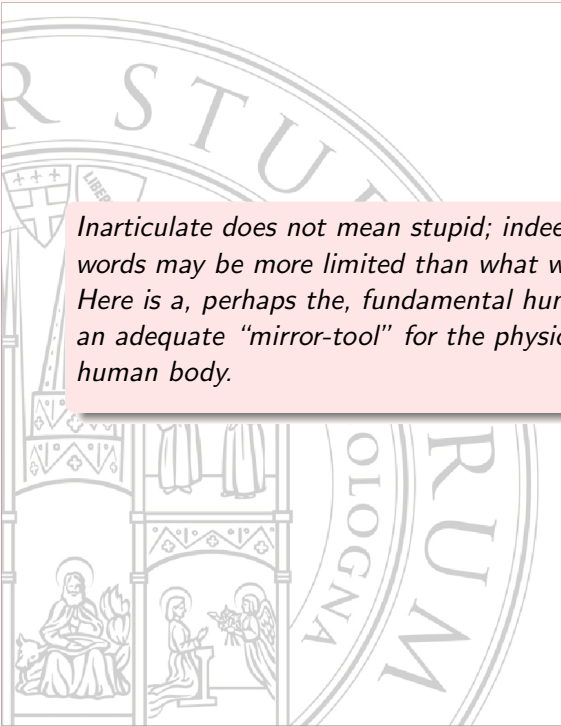
À peine, entre mille, en trouve-t-on une douzaine en état de s'exprimer avec quelque clarté sur les instruments qu'ils emploient et sur les ouvrages qu'ils fabriquent.

[D. Diderot, Prospectus à l'Encyclopédie, 141; 1751.]

Ci siamo rivolti ai più abili di Francia. Siamo andati nelle loro botteghe. [...]

A malapena su mille se ne trovano una dozzina capaci di esprimersi con qualche chiarezza sugli strumenti che impiegano e sui lavori che fabbricano.

[D. Diderot, Prospectus à l'Encyclopédie, 141; 1751.]



Inarticulate does not mean stupid; indeed, what we can say in words may be more limited than what we can do with things. [...] Here is a, perhaps the, fundamental human limit: language is not an adequate “mirror-tool” for the physical movements of the human body.

[R. Sennett, The Craftsman. 2009]



Ma anche:

Finchè ci sarà uno che conosce 2000 parole e uno che ne conosce 200, questi sarà oppresso dal primo.

La parola ci fa uguali.

[Scuola di Barbiana, Lettera ad una professoressa. 1967]

Meccanismi di astrazione

- Il contributo **specifico** dell'informatica al problem solving
- Linguaggi (artificiali) in cui esprimere le soluzioni
- Meccanismi linguistici di astrazione:
 - sul controllo
 - sui dati
- che permettono di *scalare* tra livelli diversi

Un esempio

Cammino minimo tra due città

- Astrazione sui dati

 - Mappa modellata con grafo

 - Grafo realizzato con liste concatenate

 - Liste concatenate realizzate con...

- Astrazione sul controllo

 - Algoritmo su grafo astratto

 - Algoritmo su specifica rappresentazione

 - Algoritmo modulare:

 - funzioni separate risolvono sottoproblemi separati

 - ricomposizione della soluzione

Linguaggi “di programmazione”

No scientific discipline exists without first inventing a visual and written language which allows it to break with its confusing past.

[B. Latour, Visualisation and Cognition: Thinking with Eyes and Hands; 1986]

*Referring to Dagognet, F.: Tableaux et Langages de la Chimie. Paris : Le Seuil 1969;
and to: Ecriture et Iconographie. Paris : Vrin 1973.*

*Seguiamo Latour,
immaginando una carta geografica rispetto al territorio che
descrive:*

La “mappa” è:

- mobile
- immutabile quando sono spostata
- piana
- “la sua scala può essere cambiata a piacimento”:
phenomena can be dominated with the eyes and held by hands
- riprodotta e comunicata a basso costo
- può essere spostata e ricombinata
- può essere inserita in un testo scritto
- è compatibile con la geometria ed è un modello fedele della realtà

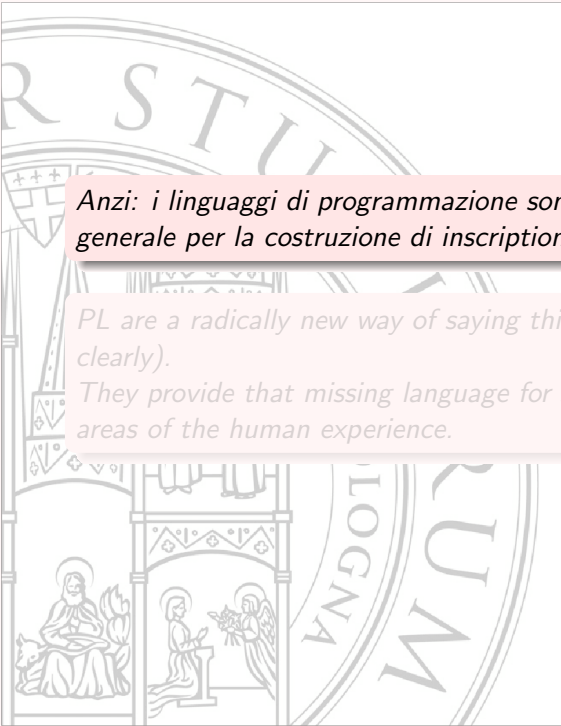
Sono *inscriptions*: come i diagrammi, o come... i programmi!

*Seguiamo Latour,
immaginando una carta geografica rispetto al territorio che
descrive:*

La “mappa” è:

- mobile
- immutabile quando sono spostata
- piana
- “la sua scala può essere cambiata a piacimento”:
phenomena can be dominated with the eyes and held by hands
- riprodotta e comunicata a basso costo
- può essere spostata e ricombinata
- può essere inserita in un testo scritto
- è compatibile con la geometria ed è un modello fedele della realtà

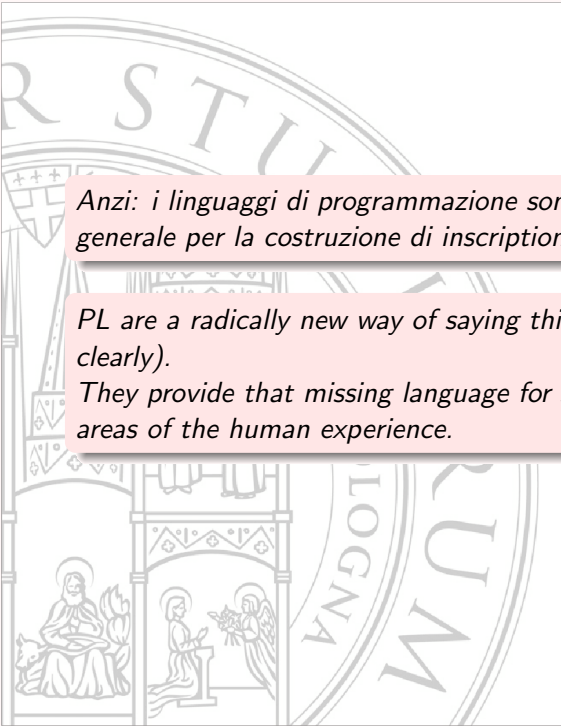
Sono **inscriptions**: come i diagrammi, o come... i programmi!



Anzi: i linguaggi di programmazione sono un linguaggio formale e generale per la costruzione di inscriptions.

PL are a radically new way of saying things (and saying them clearly).

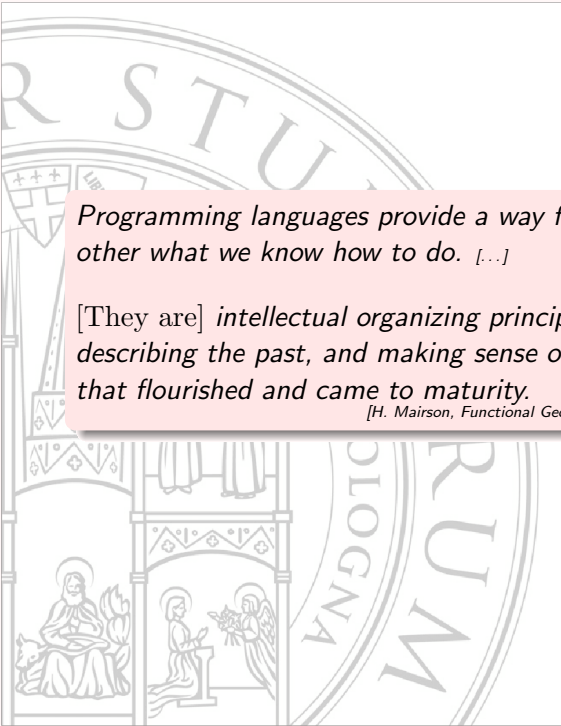
They provide that missing language for saying things in several areas of the human experience.



Anzi: i linguaggi di programmazione sono un linguaggio formale e generale per la costruzione di inscriptions.

PL are a radically new way of saying things (and saying them clearly).

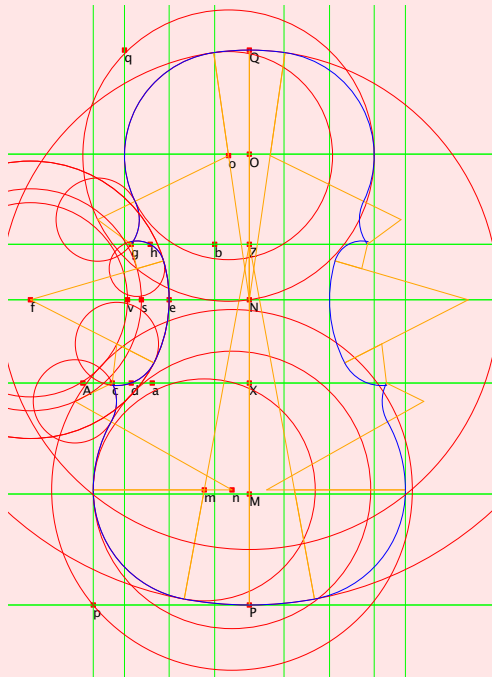
They provide that missing language for saying things in several areas of the human experience.

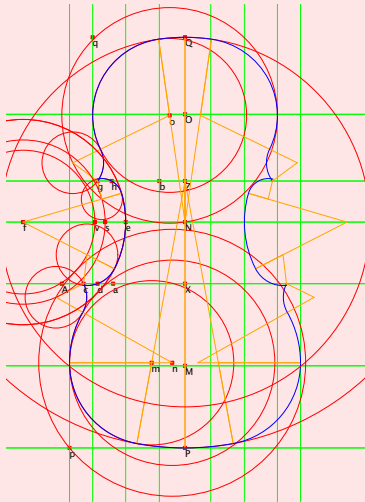


Programming languages provide a way for us to describe to each other what we know how to do. [...]

[They are] intellectual organizing principle[s] for understanding and describing the past, and making sense of the kinds of expertise that flourished and came to maturity.

[H. Mairson, Functional Geometry and the Traité de Lutherie. ICFP 2013]





Appendix: Violin by Andrea Amati

```
(define Amati
  (let ((sq 400)) ;; should be 200mm in the Amati---this is just a screen fit...

    ; LAYOUT OF THE AREA on which the curves are drawn...
    (let ((I (label "I" (point 0 0))) ; this could be anywhere---just to center it on the output screen
          (C (label "C" (xshift I (- (/ sq 2))))
              (label "D" (yshift I sq))))
      (let ((H (label "H" (pointfrom I (/ 4 4)))
            (G (label "G" (xshift (intersect (horizontal G) (vertical A))
                                      (/ (distance X H) 2))))
            (V (xshift A (/ (distance X H) 4))
              (D (label "D" (xshift G (- (/ (distance X H) (/ 2 4))))
                (E (label "E" (yshift B (- (/ (distance X H) (/ 2 2))))
                  (F (label "F" (yshift I (- (/ (distance X H) (/ 2 2))))
                    (G (label "G" (distance (horizontal P) (vertical v))))
                    (H (label "H" (pointfrom I (/ 4 2)))
                      (I (label "I" (distance X H) 2))))
                    (let ((b (label "b" (xshift I (- (/ (distance X H) 2))))
                          (let ((a (label "a" (xshift (intersect (vertical B) (horizontal H)) (- (/ (distance b p) (/ 2 H))))
                                (c (label "c" (xshift (intersect (vertical C) (horizontal E)) (/ (distance a p) 4)))
                                (d (label "d" (xshift (intersect (vertical g) (horizontal Z)) (/ (distance a p) 2)))
                                (e (label "e" (xshift (intersect (vertical ee) (horizontal Z)) (- (/ (distance a p) 4))))
                                (f (label "f" (xshift (intersect (vertical ee) (horizontal Z)) (- (/ (distance a p) 2))))
                                (list 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100)
                                  (horizontal H) (horizontal D) (horizontal P) (horizontal E) (horizontal Z) (horizontal g) (vertical q) (vertical l) (vertical ee)
                                  (vertical p) (vertical q) (vertical l) (vertical ee)

                                ; THE LOWER ROUTE.
                                (let ((C2(circircle (circle 2 (distance 2 H)))
                                      (D2(D(circle 2 (distance 2 P))))
                                      (let ((h (label "h" (bottom (intersect 2Bcircle
                                                                                   (make-line 1 p) ; line w/slope 1 through p
                                                                                   ))))
                                          (let ((C1(circircle (circle a (distance H P))))
                                                (let ((h (label "h" (xshift M (- (distance X Z) (distance H P))))
                                                    (let ((C1(circle (circle a (distance H P) (distance X Z) (- (distance H P))))
                                                          (reverse-lower-left
                                                           (lower-circle (reverse-curve (circle a (distance X Z)) (+ (distance X H) 20) 4)))
                                                           (list m (circle a (distance a (center reverse-lower-left)))
                                                           (D2(circircle (circle (reverse-lower-left)
                                                                                   (make-curve p (- (list 2P(circle acircle acircle reverse-lower-left))
                                                                                   ))))))

                                ; THE UPPER ROUTE...
                                (let ((C3(circle (circle M (distance H Q)))
                                      (e (label "e" (top (intersect
                                                                                   (circle H (distance H D)
                                                                                   (make-line -1 q) ; line w/slope -1 through q
                                                                                   ))))
                                      (let ((circle (circle c (distance Q D)))
                                          (let ((reverse-upper-left
                                                  (upper-circle (reverse-curve ocircle
                                                              (distance H D)
                                                              (p))
                                                              (list a (circle a (distance a (center reverse-upper-left)))
                                                              (circle acircle reverse-upper-left)
                                                              (make-curve q (list 2P(circle ocircle reverse-upper-left))
                                                              ))))

                                ; THE MIDDLE ROUTE...
                                (let ((F (label "F" (xshift ee (- (distance X E) 2)))
                                      (v (label "v" (xshift ee (- (/ (distance X H) 2))))
                                      (e (label "e" (xshift ee (- (/ (distance X E) 2))))
                                      (let ((acircle (circle f (distance f ee)))
                                          (circle (circle f (distance f v)))
                                          (circle (circle f (distance f 4)))
                                          (let ((reverse-lower-middle
                                                  (upper-circle (reverse-curve acircle (distance f v) 4))
                                                  (reverse-upper-middle
                                                   (lower-circle (reverse-curve acircle (distance a) h)))
                                                  (list f v a (circle f (distance f ee)) reverse-upper-middle reverse-lower-middle
                                                  (make-curve c (list reverse-upper-middle (circle f (distance f ee)) reverse-lower-middle)
                                                  ))))))))))))
```