

# Modelli e Linguaggi

Dipartimento di Elettronica e Informazione  
Politecnico di Milano

17 marzo 2017

# Modelli

## Progettazione basata su modelli

- La fase di progetto di un artefatto complesso si basa sull'uso di *modelli*
- Il modello consente di focalizzarsi sugli aspetti primari del problema e permette verifiche preliminari sul funzionamento

## Come modellare?

- Modelli fisici: riproduzioni in scala (e.g., dighe in miniatura)
- Modelli teorici/formali: oggetti matematici che consentono una visione astratta del fenomeno modellato

# Modelli formali

## Uso dei modelli formali

- 1 Formalizzazione del problema: ottenere una descrizione astratta dell'entità reale
- 2 Risoluzione del problema formalizzato
- 3 Interpretazione dei risultati alla luce del modello, nel contesto dell'entità reale → (ri-)valutazione delle scelte di progetto

## Quando un modello “funziona”?

- Un modello è *adeguato* se riflette correttamente il fenomeno modellato per gli aspetti che interessano
- Posso approssimare un cavallo a una sfera/punto materiale?
  - Per progettare una sella? Hmm ... no.
  - Per calcolare quanto è robusto il pavimento della stalla? Sì

# Modelli dell'informatica

## Una “realtà” molto vicina al modello ...

- L'informatica è una delle discipline con maggior vicinanza tra il modello e la realtà di cui si occupa (il calcolo)
- Nella sua evoluzione i primi calcolatori sono stati la materializzazione diretta del modello di calcolo (“inversione” di tendenza)

## ... e i relativi vantaggi

- I modelli formali sono di grande aiuto nella progettazione di HW e SW
  - Garanzie di correttezza, efficienza, robustezza, sicurezza
- Certificazione del livello di formalizzazione dell'artefatto HW/SW inclusa in standard (ISO/IEC 15408, livelli EAL)

# Applicazioni dell'informatica

## L'informatica a supporto

- L'informatica trova applicazioni in svariati contesti → un particolare calcolo effettuato è il modello di qualcosa di reale
- Il successo è determinato dalla flessibilità del calcolo astratto come modello di una sequenza di ragionamenti logici
- Spesso la difficoltà principale sta proprio nel *formulare* il modello (=definire il problema)!
- Cosa significa dare un modello (quantitativo) a:
  - estrazione di “informazione” da grandi moli di dati
  - garantire la sicurezza di un sistema rispetto a incidenti/dolo

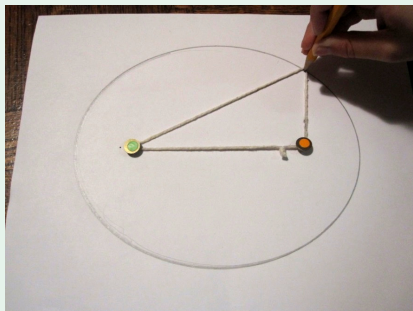
# Alcune provocazioni

## E. W. Dijkstra - premio Turing nel 1972

- *[...] in our area, perhaps more than everywhere else, mathematical elegance is not a dispensable luxury but decides between success and failure.*
- *It is nice to know the dictionary definition for the adjective "elegant" in the meaning "simple and surprisingly effective".*
- *Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians.*
- *If you want more effective programmers, you will discover that they should not waste their time debugging, they should not introduce the bugs to start with.*

# Una classificazione dei modelli - Ellisse

## Ellisse - Modello operativo



## Ellisse - Modello descrittivo

$$a, b, c, x, y \in \mathbb{R}$$

$$ax^2 + by^2 = c$$

# Una classificazione dei modelli - Ordinamento

## Modello operativo

Dato un insieme di  $n$  elementi disordinati,

- 1️⃣ ripeti  $n-1$  volte
  - 1️⃣ Trova l'elemento più piccolo tra quelli da ordinare
  - 2️⃣ Aggiungilo in coda agli elementi ordinati
- 2️⃣ aggiungi in coda l'ultimo elemento

## Modello descrittivo

La permutazione della sequenza data tale che

$$\forall i, a[i] \leq a[i + 1]$$



# Il linguaggio

## Un (il?) primo modello

- Il linguaggio è da sempre uno strumento per esprimere modelli
- Esempi di linguaggio
  - Lingue naturali: italiano, inglese, francese, giapponese
  - Linguaggi sintetici: C, Java, XML, JSON
  - Linguaggi grafici: cartelli stradali, simboli di lavaggio capi
  - Linguaggi acustici: musica, fischi degli arbitri sportivi

# Gli elementi del linguaggio

## Alfabeto

- Un linguaggio è definito su un *alfabeto*
- L'alfabeto è un insieme *finito* di simboli
- Esempi:
  - $\{a, b, c, \dots, z\}$
  - $\{0, 1\}$
  - Codice Morse, Codice Baudot, Codice ASCII
  - $\{\text{♪}, \text{♪}, \dots\}$
  - $\{\text{🐟}, \text{🏛️}, \text{🔪}, \dots\}$

# Gli elementi del linguaggio

## Stringa su un alfabeto $A$

- Stringa: sequenza *ordinata* e *finita* di elementi dell'alfabeto
- Esempio: a, stringa, vicesindaco, klatubaradanikto
- Lunghezza di una stringa  $\rightarrow$  numero di elementi:  
 $|a| = 1$ ,  $|stringa| = 7$
- Stringa nulla  $\varepsilon$ ,  $|\varepsilon| = 0$
- $A^*$  insieme di tutte le stringhe (inclusa  $\varepsilon$ ) su  $A$ 
  - Esempio:  $A = \{0, 1\}$ ,  $A^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 100, 101, \dots\}$

# Operazioni tra stringhe

## Concatenazione

- L'operatore di concatenazione è il  $.$  (può essere omissso)
  - Con  $x=\text{bana}$ ,  $y=\text{na}$ ,  $z=\text{ch}$ , si ha  $x.y=\text{banana}$ ,  $x.z=\text{banach}$
- L'operazione di concatenazione è binaria, associativa, non commutativa

## Monoide libero

- Dato un alfabeto  $A$ ,  $(A^*, .)$  è il *monoide libero* costruito su  $A$ 
  - L'elemento neutro del monoide è la stringa nulla  $\varepsilon$

# Linguaggio

## Linguaggio su un alfabeto $A$

- Un linguaggio  $L$  su un alfabeto  $A$  è un sottoinsieme di  $A^*$ 
  - ... ovvero un insieme di stringhe di simboli di  $A$  (parole)
- Esempi:
  - l'italiano è un linguaggio su  $A = \{a, b, c, \dots, z, -\}$
  - i files PDF sono un linguaggio su  $A = \{0, 1\}$
  - i numeri pari in base 4 sono un linguaggio su  $A = \{0, 1, 2, 3\}$
  - il DNA è un linguaggio codificabile su  $A = \{C, G, A, T\}$
- n.b. anche se  $|A| < \infty$ , non necessariamente  $|L| < \infty$
- A seconda dell'alfabeto scelto, un linguaggio può modellare moltissime cose (è, in un certo senso, *universale*)

# Operazioni tra linguaggi

## Operazioni insiemistiche

- $L$  è un insieme, valgono  $\cup$  (unione),  $\cap$  (inters.),  $\setminus$  (differenza)
- Il complemento  $\neg L = \bar{L}$  è definito come  $A^* \setminus L$

## Concatenazione tra linguaggi

- Dati  $L_1$  su  $A_1$ ,  $L_2$  su  $A_2$ ,  $L_1.L_2$ , definito su  $A_1 \cup A_2$  è:

$$L_1.L_2 = \{x.y \mid x \in L_1, y \in L_2\}$$

- Es.: Dati  $L_1 = \{0, 1\}^*$ ,  $L_2 = \{a, b\}^*$  abbiamo  
 $L_1.L_2 = \{\varepsilon, 0, 1, 0a, 011b, 0aba, \dots\}$ . n.b  $a1 \notin L_1.L_2$

# Operazioni tra linguaggi

## Alcune proprietà

- $L^n, n \in \mathbb{N}$ , la concatenazione di  $L$  con sè stesso  $n$  volte
  - in sintesi  $L^0 = \{\varepsilon\}, L^n = L^{n-1}.L$
- N.b : il linguaggio vuoto  $L_1 = \emptyset \neq \{\varepsilon\} = L_2$  il linguaggio fatto dalla sola stringa vuota.
  - effetti collaterali includono  $\{\varepsilon\}.L = L \neq L.\emptyset = \emptyset$

## Operatore +

- L'operatore + indica stringhe fatte concatenando *uno* o più elementi dell'insieme.
  - $A = \{0, 1\}, A^+ = \{0, 1, 00, 01, 10, \dots\}$
- Per estensione:  $L^* = \bigcup_{n=0}^{\infty} L^n, L^+ = \bigcup_{n=1}^{\infty} L^n$

# Operazioni tra linguaggi

## Nella pratica

- È possibile definire un formato di file come linguaggio (e.g., una pagina HTML)
- L'intersezione tra linguaggi ci indica se due formati sono "compatibili"
  - Se  $L_1$  sono i documenti HTML 4 e  $L_2$  quelli HTML 5,  $L_1 \cap L_2$  sono i documenti corretti secondo entrambi gli standard
- La concatenazione di linguaggi consente di descrivere più agevolmente formati complessi
  - Formato di un pacchetto di dati su rete  $L_{\text{header}} \cdot L_{\text{dati}} \cdot L_{\text{trailer}}$
  - Archivio di dati tar o zip:  $L_{\text{indice}} \cdot L_{\text{file}}^n$



# Usi del linguaggio

## Il linguaggio come formalismo espressivo

- Un linguaggio può essere usato per esprimere un problema:
  - “Trovare la miglior mossa successiva in una partita a scacchi”
  - “Trovare tre numeri interi positivi tali per cui  $x^3 + y^3 = z^3$ ”
- L'insieme delle soluzioni di un problema è un linguaggio:
  - Risolvere il problema  $\rightarrow$  calcolare (riconoscere) un  $x \in L$
  - Es: “Questo programma C è sintatticamente corretto?”
    - Data  $x$  dire se  $x \in L_{\text{linguaggio C}}$
  - “Questa immagine digitale contiene un volto?”
  - “Questo brano digitalizzato è in quattro quarti?”
  - “Questa funzione è la derivata di  $x^3 + x$ ?”

# Da un linguaggio all'altro

## Traduzioni

- Tradurre = mettere in corrispondenza parole di due linguaggi
- Formalmente, una traduzione è una mappa  $\tau(\cdot) : L_1 \rightarrow L_2$ 
  - $L_1 = \{a, b\}^*$ ,  $L_2 = \{c, d\}^*$ ,  $\tau$  mappa  $a \mapsto c$ ,  $b \mapsto d$ ,  $\tau(ba) = dc$
  - $L_1 = \{0, 1\}^*$ ,  $L_2 = \{0, 1\}^*$ ,  $\tau$  scambia 1 con 0:  $\tau(010) = 101$

## Esempi pratici

- L'insieme delle soluzioni di un problema è un linguaggio:
  - Compressione/cambio di formato di un file
  - Compilazione di un programma in codice oggetto
  - Correzione ortografica/sintattica meccanizzato

# Conclusione

## Linguaggi come metodo espressivo

- Il linguaggio costituisce un mezzo per descrivere la natura di un problema e quella delle sue soluzioni
- Useremo la formalizzazione del linguaggio come elemento fondamentale per formalizzare il concetto di calcolo
- Il passaggio alla pratica è “più diretto di quanto sembri”, dopotutto il calcolatore maneggia stringhe su un alfabeto binario...