Linguistica Computazionale – Lezione 2

# Strumenti linguistico-formali

Mercoledì 7 Marzo 2007

Cristiano Chesi, chesi@media.unisi.it



## Strumenti linguistico-formali

- Indice
  - Grammatiche formali
    - Nozioni di base (grammatiche a struttura sintagmatica, PSG)
    - Gerarchia di Chomsky
    - Descrizioni Strutturali e derivazioni
  - Formalismi applicabili alla MT
    - Regole di Transfer
    - Interlingua e Ontologie
    - Grammatiche ad unificazione
    - Principi e Parametri



## Letture, approfondimenti

- **Bibliografia essenziale**
  - Hutchins & Somers (1992) Cap. 2
  - Jurafsky & Martin (2000) Cap. 1, 2
- **Approfondimenti**
  - Turcato D. (1993) *Grammatiche formali e linguaggio naturale*. Calderini Bologna
  - Allegranza, V., Mazzini G. (2000) *Linguistica generativa e grammatiche a unificazione*. Paravia scriptorium.
  - Baker M. (2001) *The Atoms of Language*. Basic Books



## Competence = grammatica



- **Competence** (data-structure, natura del problema)
  - di che tipo di struttura dati ha bisogno la conoscenza linguistica?
    - una parola può iniziare per *ma…* (*mare*) ma non per *mr…*
    - la e di case ha un valore diverso da quella di mare
    - "le case sono sulla collina" Vs. *"case le collina sono sulla"
    - il gatto morde il cane > sogg: gatto(agente); verbo: morde(azione); ogg: cane(oggetto)
    - ?il tostapane morde il gatto
    - l'espressione "le case" si riferisce ad un gruppo di case evidente dal contesto (Vs. "delle case")
  - ad ogni livello si devono specificare delle primitive elementari:
    - **fonemi** - tratti segmentali e soprasegmentali
    - **morfemi** - identificazione delle regole combinatorie
    - **parole** - gruppi di morfemi significativi
    - **sintagmi** - gruppi tipizzati di parole che esprimono relazioni
    - **elementi tematici** - paziente, agente…
    - **elementi discorsivi** - convenzioni, relazioni pragmatiche pertinenti…

## Livelli di adeguatezza di una grammatica



- **Adeguatezza**: una grammatica deve fornire una descrizione adeguata rispetto alla realtà empirica a cui si riferisce. In particolare si può parlare di adeguatezza a tre livelli:
  - **osservativa**: la lingua definita dalla grammatica coincide con quella che si intende descrivere
  - **descrittiva**: l'analisi grammaticale proposta è in linea con le intuizioni linguistiche dei parlanti fornendo descrizioni strutturali adeguate delle frasi accettabili
  - **esplicativa**: i dispositivi generativi utilizzati soddisfano criteri di plausibilità psicolinguistica e riproducono realmente i meccanismi operanti nell'attività linguistica del parlante. Una grammatica si dice esplicativa quando rende conto anche dell'apprendibilità della lingua.



## Come si formalizza una grammatica

(inizio ...)



- **A = Alfabeto**
  insieme finito di caratteri (A* = l'insieme di tutte le stringhe possibili costruite concatenando elementi di A; ε è l'elemento nullo)
- **V = Vocabolario**
  insieme (potenzialmente in)finito di parole, costruite concatenando elementi di A ($V \subseteq A^*$)
- **L = Linguaggio**
  insieme (potenzialmente in)finito di frasi, costruite concatenando elementi di V ($L \subseteq V^*$)



## Come si formalizza una grammatica

(... continua ...)



- Una **grammatica formale** per il linguaggio **L** è un insieme di regole che permettono di **generare/riconoscere** tutte e sole le frasi appartenenti a **L** e(d eventualmente) di assegnare a queste frasi un'adeguata descrizione strutturale.

  Una grammatica formale G deve essere:
  - **esplicita** (il giudizio di grammaticalità deve essere frutto solo dell'applicazione meccanica delle regole scelte)
  - **consistente** (una stessa frase non può risultare allo stesso tempo grammaticale e non grammaticale)



## Come si formalizza una grammatica

(... continua ...)



- Una grammatica formale **G** può essere formalizzata (**grammatica a struttura sintagmatica** o **Phrase Structure Grammar**, **PSG** Chomsky 1965), come una quadrupla ordinata **<V, $V_T$, →, {S}>** dove:
  - **V** è il **vocabolario** della lingua
  - **$V_T$** è un sottoinsieme di **V** che racchiude tutti e soli gli **elementi terminali** (il complemento di VT rispetto a V sarà l'insieme di tutti i vocaboli non terminali e sarà definito come **$V_N$**)
  - **→** è una relazione binaria, asimmetrica e transitiva definita su **V***, detta **relazione di riscrittura**. Ogni coppia ordinata appartenente alla relazione è chiamata **regola di riscrittura**. Per ogni simbolo $A \in V_N$ $\varphi A \psi \rightarrow \varphi \tau \psi$ per qualche $\varphi, \tau, \psi \in V^*$
  - **{S}** è un sottoinsieme di $V_N$ definito come l'insieme degli assiomi che convenzionalmente contiene il solo simbolo **S**.

## Come si formalizza una grammatica

(... fine)



- Date due stringhe φ e ψ ∈ V* si dice che esiste una **φ-derivazione di ψ** se φ →* ψ.
- Se esiste una φ-derivazione di ψ allora si può anche dire che **φ domina ψ**. Tale relazione è riflessiva e transitiva.
- Una φ-derivazione di ψ si dice **terminata** se:
  - ψ ∈ $V_T$*
  - per nessun χ esiste una ψ-derivazione di χ
- Data una grammatica G, una **lingua generata** da G, detta **L(G)**, è l'insieme di tutte le stringhe φ per cui esiste una S-derivazione terminata di φ.



## Descrizioni strutturali (cioè alberi sintattici)



- Una **descrizione strutturale** è una quintupla **<V, I, D, P, A>** tale che:
  - **V** è un insieme finito dei **vertici** (es. v1, v2, v3…)
  - **I** è l'insieme finito degli **identificatori** (es. S, DP, VP, la, casa…)
  - **D** è la relazione di **dominanza**. È un ordine debole (cioè una relazione binaria, riflessiva, antisimmetrica e transitiva) definita su V
  - **P** è la relazione di **precedenza**. È un ordine stretto (cioè una relazione binaria, irriflessiva, antisimmetrica e transitiva) definita su V
  - **A** è la **funzione di assegnazione**; una funzione non suriettiva da V a I



## Capacità generativa e relazioni di equivalenza



- La **capacità generativa** denota l'insieme di oggetti generati dalla grammatica; tale capacità è:
  - **debole** se riferita al solo semplice insieme di frasi generabili
  - **forte** se associa a tali frasi l'appropriata descrizione strutturale
- Due grammatiche si dicono **equivalenti** se sono in grado di generare lo stesso insieme di oggetti.
  Anche qua si può parlare di **equivalenza debole** o **equivalenza forte**



## Decidibilità



Un insieme **Σ** si dice

- **decidibile** (o **ricorsivo**) se per ogni elemento ***e*** appartenente all'insieme universo esiste un **procedimento meccanico *M*** che permette di stabilire in un **numero finito di passi** se **e** ∈ oppure ∉ a **Σ** (la non appartenenza a **Σ** determina l'appartenenza al complemento di **Σ** definito come $\overline{\Sigma}$)
- **ricorsivamente enumerabile** quando esiste un procedura che enumera tutti e soli gli elementi di Σ

## Inclusioni tra classi di grammatiche

(inizio ...)



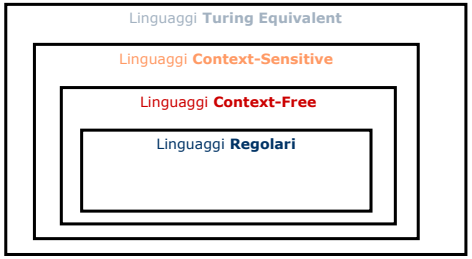
- La **gerarchia di Chomsky** (1956, 59) pone in relazione grammatiche di potenza diversa ponendo restrizioni sulla struttura delle regole:
- **Tipo 0**: grammatiche non ristrette (**Turing Equivalent**):
  $\alpha \rightarrow \beta$ $(\alpha \neq \varepsilon)$ [Augmented Transition Networks]
- **Tipo 1**: grammatiche contestuali (**Context Sensitive**):
  $\alpha A \beta \rightarrow \alpha \gamma \beta$ $(\gamma \neq \varepsilon)$ [Tree Adjoining Grammars]
- **Tipo 2**: grammatiche non-contestuali (**Context Free**):
  $A \rightarrow \gamma$ [Phrase Structure Grammars]
- **Tipo 3**: grammatiche **regolari**:
  $A \rightarrow xB$ [Finite State Automata]



## Inclusioni tra classi di grammatiche

(... fine)



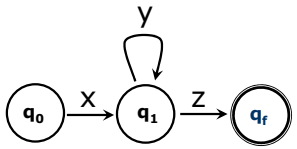





## Come si stabilisce l'appartenenza ad una grammatica



- **Pumping lemmas**
  Servono per verificare se una proprietà linguistica può essere catturata da una grammatica oppure no
- **Pumping lemma per le grammatiche regolari**
  $a^n b^n$ non è una stringa generabile da nessuna grammatica regolare (poiché nessuna sottostringa può essere "pompata" indefinitivamente garantendo lo stesso numero di *a* e di *b*)






## Dove stanno le lingue naturali?

(inizio ...)



- Le lingue naturali **non sono generabili da grammatiche regolari** (Chomsky 1956):

  If A then B (con A e B potenzialmente anch'esse nella forma "if X then Y"... quindi linguaggi di tipo $a^n b^n$)

- Le lingue naturali **non sono generabili da grammatiche context-free** (Shieber 1985):

  Jan säit das mer em Hans es huus hälfed aastriiche
  ("famoso" dialetto svizzero tedesco)
  J. dice che noi a H. la casa abbiamo aiutato a dipingere

  Gianni, Luisa e Mario sono rispettivamente sposato, divorziata e scapolo
  ( "ABC...ABC"... quindi linguaggi di tipo XX)

## Dove stanno le lingue naturali?

(... continua ...)



**Ricorsività** nelle lingue naturali, ovvero come fare un uso infinito di mezzi finiti:

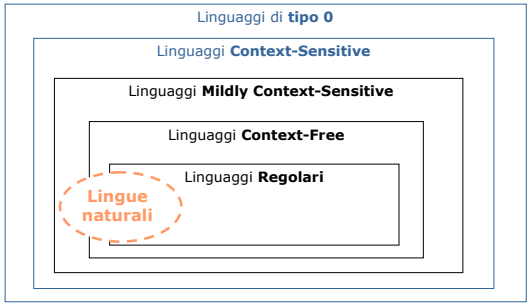
- **Incassamento a destra** ($ab^n$: iterazione):
  [il cane morse [il gatto [che rincorse [il topo [che scappò]]]]]
- **Incassamento centrale** ($a^nb^n$: counting recursion):
  [il cane [che il gatto [che il topo che scappò], rincorse], morse]
- **Dipendenze cross-seriali** (xx, identity recursion)
  Gianni, Maria e Marco sono rispettivamente sposato, nubile e divorziato



## Dove stanno le lingue naturali?

(... fine)








## Altri fenomeni linguistici interessanti "catturabili" con CFGs



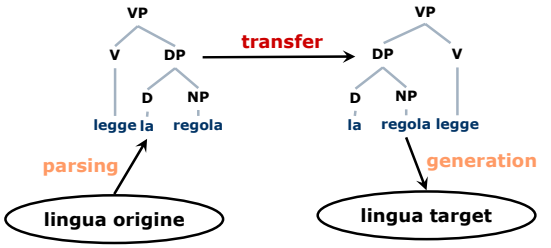
- **accordo**
  per cogliere fenomeni di accordo si deve ricorrere alla duplicazione delle regole di riscrittura. Ad esempio:
  $D_{pl} \rightarrow i$ , $N_{pl} \rightarrow cani$, $D_{sg} \rightarrow il$, $N_{sg} \rightarrow cane$, $DP \rightarrow (D_{sg}\ N_{sg} \mid D_{pl}\ N_{pl})$
- **sottocategorizzazione**
  ci si riferisce allo schema di sottocategorizzazione come alla possibilità di distinguere ulteriormente, all'interno di categorie maggiori (ad esempio la categoria verbale), categorie più precise che rendano conto in modo più adeguato del comportamento dei diversi elementi lessicali: transitivo, intransitivo, inaccusativo o ergativo sono solo tre delle sotto-classi verbali che servono a predire un determinato comportamento verbale (certi approcci, Levin 83, distinguono fino a 183 classi verbali distinte).
  esempi di regole:
  $VP \rightarrow (V_{transitivo}\ DP \mid V_{intransitivo} \mid V_{frasale}\ S \mid V_{modale}\ V_{inf}\ldots)$



## Modello del Transfer



- **Conoscenza contrastiva**
  esplicitare le differenze tra le due lingue è il primo passo verso la traduzione.
  Da questo punto di vista occorre una ristrutturazione linguistica per conformarsi alle regole della lingua target



## Esempi di Transfer con Context-Free Grammars

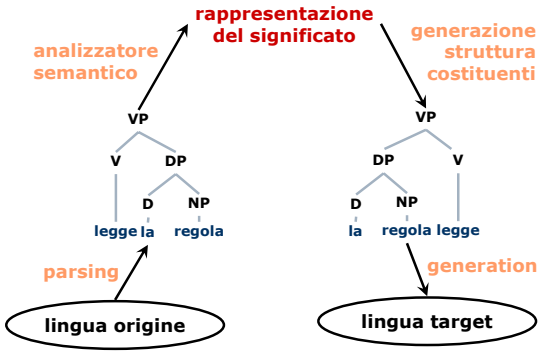


- Inglese ⇒ Italiano
  $DP \rightarrow D_1\ Agg_2\ Nome_3$ ⇒ $DP \rightarrow D_1\ Nome_3\ Agg_2$
- Giapponese ⇒ Inglese
  $DP \rightarrow relativa_1\ DP_2$ ⇒ $DP \rightarrow DP_2\ relativa_1$



## Modello dell'interlingua








## Ontologie

(inizio ...)



- **Concettualizzazione**
  astrazione (e semplificazione) delle relazioni tra oggetti e concetti in un dominio di conoscenza
- **Ontologia**
  specificazione dettagliata di queste entità e relazioni

  Ogni ontologia definisce un insieme di **classi**, **relazioni**, **funzioni** e **oggetti** costanti all'interno di un dominio discorsivo, esplicitando un'**assiomatizzazione** in modo da **vincolare interpretazioni** ed **inferenze**



## Ontologie

(... fine)



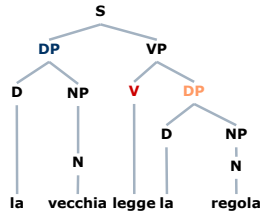
- **Knowledge Interchange Format**
  (**KIF**, Genesereth & Fikes, 1992)

```
(defrelation PHYSICAL-QUANTITY
    (<=> (PHYSICAL-QUANTITY ?q)
    (and (defined (quantity.magnitude ?q))
            (double-float (quantity.magnitude ?q))
            (defined (quantity.unit ?q))
                    (member (quantity.unit ?q)
                    (setof meter second kilogram ampere kelvin
                            mole candela)))
```

## Rappresentazione del significato



- **Aggiunta semantica** alle regole Context Free:
  $A \rightarrow \alpha_1\ \alpha_2 \ldots \alpha_n \qquad \{f(\alpha_j.sem \ldots \alpha_k.sem)\}$
  vecchia {*vecchia*}
  legge {$\exists e,x,y$ *Isa(e, leggere)* $\wedge$ *legge(e, x)* $\wedge$ *letta(e, y)*}




$\exists e$ Isa($e$, leggere) $\wedge$ legge($e$, la vecchia) $\wedge$ letta($e$, la regola)



## Grammatiche ad unificazione

(inizio ...)



- **Grammatiche basate su restrizioni**
  rappresentazione più efficiente e significativa dell'informazione linguistica
- **formalismi leggermente più "potenti" delle CFG**, con cui render conto in modo
  - **compatto** (quindi più elegante) ed
  - **efficiente**

  delle restrizioni linguistiche imposte da fenomeni produttivi quali quelli precedentemente elencati
- **gerarchie di tratti**
  proprietà aggiuntive rispetto alle regole di riscrittura

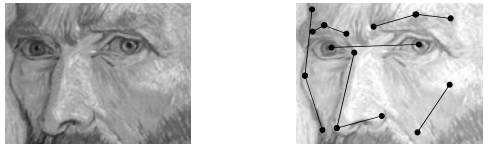


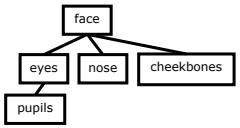
## Grammatiche ad unificazione

(... continua ...)



- **(ma... cosa sono i tratti?)**








## Grammatiche ad unificazione

(... continua ...)



- **(ma... cosa sono i tratti?)**

| **a.** | **b.** | **c.*** | **d.*** | **e.*** |
|---|---|---|---|---|
|  |  |  |  |  |
| la casa è bella | la casa è brutta | *casa la bella è | *lo casa è belli | *casa è |

## Grammatiche ad unificazione

(... continua ...)

Formalismi applicabili alla MT

- **Struttura di tratti** (**FS**, **Feature Structures**)
  è un insieme di coppie del tipo **tratto > valore**
  (es. numero > singolare)

- due tipologie di formalizzazione (equivalenti) delle coppie **tratto > valore**:

**Matrice di Attributi e Valori**
(**AVM**, **Attribute-Value Matrix**)

$$\begin{bmatrix} \text{Num} = \text{Sing} \\ \text{Gen} = \text{Femm} \\ \dots \\ \text{Tratto}_n = \text{Valore}_n \end{bmatrix}$$

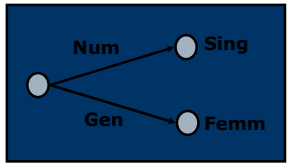



**diagramma ad archi orientati ed etichettati**
(**DAG**, **Direct Acyclic Graph**)



## Grammatiche ad unificazione

(... continua ...)

Formalismi applicabili alla MT

Alcune proprietà interessanti delle strutture di tratti:

- **parzialità**, **maggiore** o **minore specificità**, ovvero alcuni elementi posso restare non specificati, ad esempio il genere:

$$N \begin{bmatrix} \text{Num} = \text{Sing} \\ \text{Gen} = [\,] \end{bmatrix}$$

- la struttura delle AVM può essere **rientrante**, cioè un tratto che ha una qualche significatività dal punto di vista empirico, può essere definito da più sottotratti, come nel caso dell'accordo:

$$\begin{bmatrix} \text{Cat} = \text{N} \\ \text{Accordo} \begin{bmatrix} \text{Num} = \text{Sing} \\ \text{Gen} = \text{Femm} \end{bmatrix} \end{bmatrix}$$



## Grammatiche ad unificazione

(... continua ...)

Formalismi applicabili alla MT

Altre proprietà interessanti delle strutture di tratti:

- **percorsi**, il valore di un tratto è definito in base ad un percorso univoco che lo identifica, cioè una lista di tratti lungo la struttura del tipo: accordo>num>sing

- **condivisione di tipo** (**type sharing**), una struttura può essere condivisa tra più elementi anche se i valori non lo sono

- **condivisione di occorrenza** (**token sharing**), l'occorrenza di un determinato valore può essere condivisa in tal caso può essere indicato con l'uso di un indice, es [1]:

$$\begin{bmatrix} \text{Cat} = \text{S} \\ \text{testa} \begin{bmatrix} \text{Accordo [1]} \begin{bmatrix} \text{Num} = \text{Sing} \\ \text{Per} = 3 \end{bmatrix} \\ \text{Soggetto} \begin{bmatrix} \text{Accordo [1]} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$



## Grammatiche ad unificazione

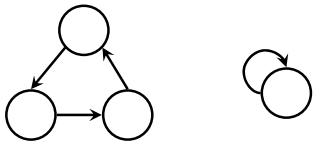
(... continua ...)

Formalismi applicabili alla MT

Alcune proprietà interessanti delle strutture di tratti:

- **significatività empirica**, i tratti sembrano catturare adeguatamente, almeno a livello descrittivo, fenomeni linguisticamente produttivi

- **aciclicità**, i grafi non possono essere ricorsivi



## Grammatiche ad unificazione

(... fine)



**Sussunzione**

stabilisce una relazione ordinata tra due strutture di tratti FS; la FS più **generica** sussume quella **specifica**. Si può quindi dire che:
$FS_a \sqsubseteq FS_b$

sse $FS_b$ ha tutti i tratti di $FS_a$ nella stessa configurazione strutturale e con uguali assegnazioni di valore

**Unificazione**

permette di combinare le informazioni per rappresentarle in formato più compatto e significativo:
$FS_a \sqcup FS_b = FS_x$ (se esiste) tale che $FS_x$ è la più generale delle FS sussunte da $FS_a$ e $FS_b$



## Da regole a principi e parametri

(inizio ...)



- **regole** — specifiche e valide per una sola lingua
- **principi & parametri** — universali linguistici + settaggio parametri di variazione
- Ricerca di una migliore **adeguatezza esplicativa** oltre che **descrittiva**
- **Obiettivo**: cogliere gli universali linguistici descrivendo precisamente la limitata variabilità sintattica
- I **principle-based parsers** (Barton 1984, Berwick e Fong 90, Stabler 92) si ispirano a queste idee:
  - i principi della grammatica sono **assiomi** per il parser
  - il parser è un **sistema deduttivo** che inferisce le espressioni grammaticali e la loro struttura partendo da tali assiomi



## Da regole a principi e parametri

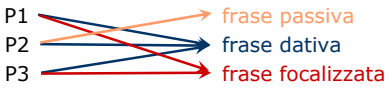
(... fine)



- **regole**

  regola passivo → frase passiva
  regola dativo → frase dativa
  regola di focalizzazione → frase focalizzata
  **...**

- **principi & parametri**




- potenzialmente una decina di principi + pochi parametri possono generare migliaia di regole

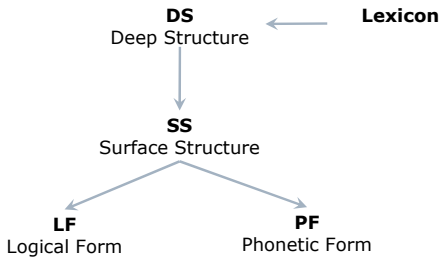


## Principi e parametri

(inizio ...)



- Modello a "T"

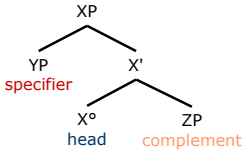


## Principi e parametri

(... continua ...)



- Alcuni principi
  - **X' theory**




  - **$\theta$ - criterion**

    ogni argomento deve ricevere uno ed un solo ruolo tematico (e ogni ruolo tematico è assegnato ad uno ed un solo argomento)

  - **Case filter**

    ogni NP lessicale deve ricevere un caso (P e $V_{finito}$ assegnano caso)



## Principi e parametri

(... continua ...)



- Altri principi
  - **Move α**

    una categoria può muoversi in qualsiasi momento, ovunque

  - **Free indexation**

    indici sono liberamente assegnati alle categorie in posizione A(rgomentale)

  - **Binding theory**

    ***condizione A*** – Un'anafora (es. *se stessa*) è legata nel suo dominio di legamento (binding domain)

    ***condizione B*** – Un pronome (es. *lei*) è libero nel suo dominio di legamento

    ***condizione C*** – Un'espressione referenziale (es. *Maria*) è sempre libera



## Principi e parametri

(... fine)



- **Generatori**

  principi che producono più strutture di quante non ne ricevano in input:
  - **Move α**
  - **Free indexation**
  - **...**

- **Filtri**

  principi che selezionano solo parte delle strutture che ricevono in input:
  - **X' theory**
  - **$\theta$ - criterion**
  - **Case filter**

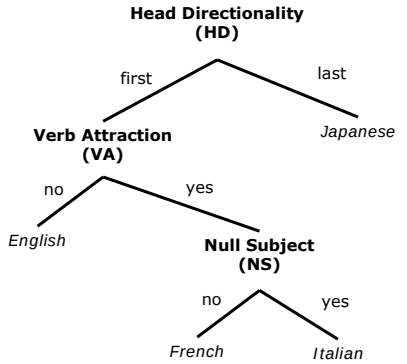


## Principi e parametri

(... fine)



- **Parametri (Baker 2002)**



## Prossima lezione

(Domani, Giovedì 8 Marzo, ore 16-18, Aula 456, Palazzo S. Niccolò)

- Indice
  - Fondamenti
    - Macchine di Turing (universali)
    - Concetto di computazione e computabilità
    - Dati, programmi, input e output
  - Basi Dati
    - Corpora
    - database
    - strumenti di interrogazione basi dati (esp. regolari, SQL)
  - Algoritmi
    - Cicli ed oggetti
    - Ideazione, descrizione, formalizzazione ed implementazione di un algoritmo