
25 Maggio2006

Introduzione
Automati & Morfologia

Marco Pennacchiotti

pennacchiotti@info.uniroma2.it

Tel. 06 7259 7717

Ing.dell'Informazione, *stanza P1B-03* (nuova ala Ing.Inf, primo piano)

Qualche info

- **Scopo delle lezioni seminariali :**
 - Introdurre *modelli e tecnologie* per l'*Elaborazione del Linguaggio Naturale* (NLP – *Natural Language Processing*)
 - Utilizzo pratico di *tool* per l'NLP su testi di *Wikipedia* (*it.wikipedia.org*)
- **Cosa faremo?**
 - Morfologia
 - Sintassi
 - Semantica

Programma

- **Breve introduzione all’NLP**
 - Linguaggi Naturali e Linguaggi Formali
 - Complessità
- **Morfologia**
 - *Teoria*: Morfologia del Linguaggio Naturale
 - *Strumenti*: Automi e Trasduttori
 - *Analisi Morfologica*: con automi e trasduttori
- **Part of Speech Tagging**
 - *Teoria*: Le classi morfologiche
 - *Strumenti a Analisi*: modelli a regole e statistici
- **Sintassi**
 - *Teoria*: Sintassi del Linguaggio Naturale
 - *Strumenti*: CFG
 - *Analisi Sintattica*: parsing top-down, bottom-up, Early
- **Semantica**
 - Lexical Semantics
 - Sentence Semantics

Qualche info

Testi :

- *Testo di riferimento:*
“*Speech and Language Processing*”, Jurafski & Martin, Prentice Hall.
- *Approfondimenti:*
 - “*Semantics*”, John I. Saeed , Blackwell Publishing
 - dispense

Sito Progetto e Lezioni:

<http://ai-nlp.info.uniroma2.it/pennacchiotti/teaching/>

Ricevimento

- Giovedì, dalle 14 alle 16
Tel. 06 7259 7717
Ing.dell'Informazione, stanza P1B-03 (nuova ala Ing.Inf, primo piano)

Qualche info

Esercizi

- Nelle slide di colore blu: esercizi di base e di preparazione al test

Parte delle slides sono un adattamento di :

- Corso IA di *Fabio Massimo Zanzotto*
- Corso di Lexical Semantics di *Patrick Pantel (ISI, Univ. Southern California)*
- Introduction to Computer Speech and Language Processing, *Dan Jurafsky*

Sogni realizzabili ?

- **Hal :** - *Buonasera, David.*
- **David:** - Come va, Hal?
- **Hal :** - *Va tutto benissimo, e tu?*
- **David:** - Beh, non c'è male.
- **Hal :** - *Hai fatto dell'altro lavoro?*
- **David:** - Sì, qualche disegno.
- **Hal :** - *Posso vederlo?*
- **David:** - Certo.
- **Hal :** - *Un'ottima esecuzione, David. Mi pare che tu abbia migliorato parecchio.*

(da 2001: Odissea nello Spazio)

Chi (o cosa) è Hal ?

Una intelligenza "Artificiale" o "Naturale" ?

Cosa vuol dire "Intelligenza Artificiale" ?

HAL

- Ascolta
- Analizza
- Comprende
- Ragiona
- E' pragmatico
- Risponde

Il test di Turing

<http://cogsci.ucsd.edu/~asaygin/tt/ttest.html>
http://it.wikipedia.org/wiki/Test_di_Turing

■ Test ideato da **Alan Turing** (1919-1954) per determinare se una macchina è in grado di pensare. Vi sono tre partecipanti:

- **A**: un esaminatore
- **B**: un uomo
- **C**: una macchina

■ A è tenuto separato da B e C, e deve stabilire chi è l'uomo e chi è la macchina ponendo delle domande

■ Se A non distingue la macchina, allora la macchina ha passato il test, ed è considerata *“in grado di pensare”*



E un bel problema !

-**Filosofico**: cosa vuol dire “pensare”? Cosa vuol dire intelligente?

-**Tecnologico**: esiste Hal? Dove è l'IA oggi?

Loebner Prize

<http://members.aol.com/NeoNoetics/MindsBrainsPrograms.html>
<http://www.loebner.net/Prizef/loebner-prize.html>

■ *Due punti di vista sull' Intelligenza Artificiale:*

■ **IA forte**

“Ragionare è calcolare”, Thomas Hobbes

- il calcolatore può essere dotato di intelligenza pura, non distinguibile da quella umana, può essere cosciente di se

■ **IA debole :**

“Nessun programma, da solo, è sufficiente a ragionare intenzionalmente”, John R. Searle

- il calcolatore ha un'intelligenza parziale, non comparabile a quella umana, può solo emulare i processi cognitivi umani

■ **Loebner Prize:** “Istanziamento” del Turing Test

- Differenti *ChatBot* vengono testati su più Turing Test (3 esaminatori eseguono più *round* di Turing Test)

■ Vincitori:

- 2004 : A.L.I.C.E. (<http://www.alicebot.org>)
- 2005 : Jabber Wacky (<http://www.jabberwacky.com>)

Intro

■ Come funziona ALICE?

- Non è poi così “intelligente”!
- Semplice insieme di 41.000 coppie domanda-risposta (*pattern-template*) espressi tramite *espressioni regolari*
- Implementato in **AIML** (*Artificial Intelligence Markup Language*), linguaggio di markup XML-compliant
- Nessun tipo di conoscenza linguistica, tranne l'utilizzo di sinonimi
- ALICE può quindi essere messa in difficoltà con *trucchi* che richiedono conoscenza linguistica!
 - *Sintassi*: “*you are a robot*” VS “*a robot, this is what you are*”



AIML knowledge base

```
<aiml>
<category>
<pattern>WE MET YESTERDAY</pattern>
<template>I remember, <get name="name"/>. We were talking about <get name="topic"/>.
</template>
</category>
<category>
<pattern>WE MUST *</pattern>
<template>
<srai>WE SHOULD <star/>
</srai>
</template>
</category>
<category>
<pattern>WE SHOULD *</pattern>
<template>I'm not sure if I am ready for that yet.
</template>
</category>
<category>
<pattern>BYE</pattern>
<template>
<random>
<li>Bye.</li>
<li>Adios.</li>
<li>Goodbye.</li>
<li>Bye bye.</li>
<li>Goodbye.</li>
<li>Sayonara.</li>
<li>Bye for now.</li>
<li>See you later!</li>
<li>See you later.</li>
<li>Until next time.</li>
<li>See you later <get name="name"/>.</li>
<li>Thanks for chatting, <get name="name"/>.</li>
</random>
</template>
</category>
</aiml>
```

→ *Category*: unità di conoscenza

→ *Pattern*: testo utente

→ *Template*: risposta di ALICE

Si può fare di meglio ??
...ci stiamo lavorando!

Obiettivi dell'NLP

L'Elaborazione del Linguaggio Naturale (*Natural Language Processing, NLP*) si prefigge come obiettivi principali:

- costruzione di modelli e strumenti informatici in grado di eseguire specifici task riguardanti il Linguaggio Naturale:
 - Permettere la comunicazione *uomo – macchina*
 - Migliorare la comunicazione *uomo – uomo*
 - Elaborare e manipolare oggetti linguistici

PERCHE' E' IMPORTANTE L' NLP ?

- Sempre maggiore quantità di conoscenza condivisa in testi in Linguaggio Naturale *machine readable (ES: il Web)*
- Necessità di un'interazione più diretta uomo-macchina (*ES: agenti intelligenti*)

How many calories should i consume daily to loose weight?

Ask

Type in your question in English, French, Spanish, German, Italian or Portuguese.

QUESTION ANSWERING: rispondere a domande in Linguaggio Naturale

(www.answerbus.com)

Who is the guitarist in Led Zeppelin?

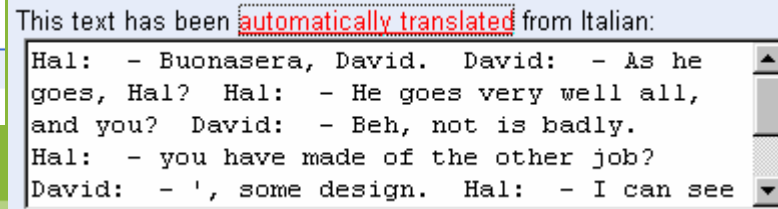
Led Zeppelin, English pop music group formed in 1968 by guitarist Jimmy Page (1944)

Who first circumnavigated the globe?

It was del Cano, NOT Magellan who first circumnavigated the GLOBE

How many calories should i consume daily to loose weight?

You must consume 3500 calories less than you eat or expend to cause a weight difference of one pound.



MACHINE TRANSLATION: tradurre automaticamente testi in lingue diverse

(translate.google.com/translate_t)

- **Hal :** - *Buonasera, David.* -*Buonasera, David.*
- **David:** - *Come va, Hal?* -*As he goes, Hal?*
- **Hal :** - *Va tutto benissimo, e tu?* - *He goes very well all, and you?*
- **David:** - *Beh, non c'è male.* - *Beh, not is badly.*
- **Hal :** - *Hai fatto de* **My name is Marco** *the other job?* → اسمي ماركو
- **David:** - *Si', qualche disegno.* - *I can see it?*
- **Hal :** - *Posso vederlo?* - *Sure.*
- **David:** - *Certo.* - *an optimal execution, David. It seems to me that you have improved much.*
- **Hal :** - *Un'ottima esecuzione, David. Mi pare che tu abbia migliorato parecchio.*

IT to EN



Intro

ANALISI NARRATIVA: aiutare un critico ad analizzare un testo letterario
(ai-nlp.info.uniroma2.it/cyberletteratura/)

RICERCA NARRATIVA

Tag
Narrativo descrizione

Sub-Tag
Narrativo azione punto di vista

Cerca

Da “*Gli Indifferenti*” di Alberto Moravia

1. *[Capitolo VII]* ma ad una certa distanza tutto si confondeva, gli alberi si torcevano come serpenti, tutto si annebbiava;
2. *[Capitolo VIII]* Alzò gli occhi, e vide davanti a sé la macchia nera del cancello, i due pilastri bianchi, il fogliame scuro di un grande albero curvo sotto la pioggia; aprì la porticina di servizio, uscì nella strada volgendo gli occhi alla parte opposta a quella dove Leo aspettava.
3. *[Capitolo XVI]* Carla seguiva attentamente la corsa e con quella stessa velocità i pensieri turbinavano nella sua mente eccitata e stanca; l'automobile era la sua vita, lanciata ciecamente nell'oscurità.

Cosa serve ?

- **CONOSCENZA LINGUISTICA:** tutta la conoscenza che ha a che vedere con il linguaggio: cos'è una parola? Quali sono le regole per costruire una frase? Qual è il significato di un sintagma?
- **MODELLI (teorie):** i modelli linguistici hanno lo scopo di catturare la conoscenza linguistica e rappresentarla in una forma comprensibile per il computer
- **ALGORITMI:** strumenti per manipolare i modelli e le strutture linguistiche necessarie per l'analisi e la comprensione del linguaggio

Cosa serve? Modelli

■ MODELLI PROCEDURALI:

- Automi a Stati Finiti
- Trasduttori a Stati Finiti
- Markov Models

■ MODELLI DICHIARATIVI:

- Grammatiche regolari
- Context Free Grammar

■ MODELLI LOGICI:

- Calcolo dei Predicati
- Logica del Primo Ordine

➤ Solitamente un modello procedurale ha una sua controparte in un modello dichiarativo (ad es. *automi – grammatiche regolari*)

➤ Un modello può essere più o meno complesso da un punto di vista computazionale (ad es. le Context Free Grammar sono più complesse di quelle Regolari)

➤ Nei diversi modelli possono generalmente essere integrati elementi di **probabilità** (**modelli probabilistici**)

Cosa devono fare i modelli ? Che analisi devono portare a termine ?

Livelli di analisi del Linguaggio Naturale

I **sistemi di NLP** possono operare a diversi livelli di analisi, ognuno dei quali richiede una specifica **conoscenza linguistica** .

- **FONETICA**: studio dei suoni linguistici
- **MORFOLOGIA**: studio delle componenti significative di una parola
- **SINTASSI**: studio delle strutture relazionali tra le parole
- **SEMANTICA**: studio del significato
- **PRAGMATICA**: studio di come il linguaggio è usato per raggiungere obiettivi
- **ANALISI DEL DISCORSO**: studio di unità linguistiche complesse

Una architettura per L’NLP può portare a termine uno o più livelli di analisi, generalmente in cascata

Livelli di analisi: *un esempio*

- **David** : - Apri la saracinesca esterna, Hal.
- **Hal** : - *Mi dispiace David, purtroppo non posso farlo.*
 - **FONETICA**: Hal deve essere in grado di analizzare il segnale audio e ricostruire la giusta sequenza delle parole
 - **MORFOLOGIA**: Hal deve saper rispondere con la giusta flessione: ad esempio *posso* e non *puoi*
 - **SINTASSI**: Hal deve sapere che *la saracinesca esterna* è un sintagma nominale complemento oggetto di *apri*, e che la frase di David è corretta
 - **SEMANTICA**: Hal deve sapere cos'è una saracinesca, e cose vuol dire *aprire qualcosa*
 - **PRAGMATICA**: Hal deve rispondere cortesemente a David
 - **ANALISI DEL DISCORSO**: Hal risponde *farlo* riferendosi a una frase del discorso precedente

Chaos: *un architettura di NLP*

- **Chaos** (SP) è un parser sintattico multilingue misto (*costituenti + dipendenze*)
- Dato un frammento di testo S in input:
 - effettua *analisi morfologica*
 - effettua *analisi sintattica*
 - produce una *interpretazione strutturata e sintattica* del testo in input, utilizzando una base di conoscenza linguistica K :

$$SP(S,K)=I$$

- Il formalismo di rappresentazione utilizzato è chiamato **XDG** (eXtended Dependency Graph) composto da:
 - **Constituents**: gruppo di parole consecutive che si comportano come una singola unità sintattica
 - **ICD** (Inter-constituent dependencies): dipendenze sintattiche tra costituenti (*es: dipendenza verbo-soggetto*)

Chaos: *un architettura di NLP*

■ Approccio modulare:

- Chaos è formato da una cascata di moduli

$$SP = P_n \bullet \dots \bullet P_2 \bullet P_1$$

- Ogni modulo P_i effettua un particolare tipo di analisi linguistica:

$$P_i(S_i, K_i) = S_{i+1} \equiv P_i(S_i) = S_{i+1}$$

- Ogni modulo produce una versione arricchita dell'XDG, utilizzando una specifica conoscenza linguistica e uno specifico modello

■ Moduli principali:

- Morpho-Analizer

- PoS tagger

- Chunker

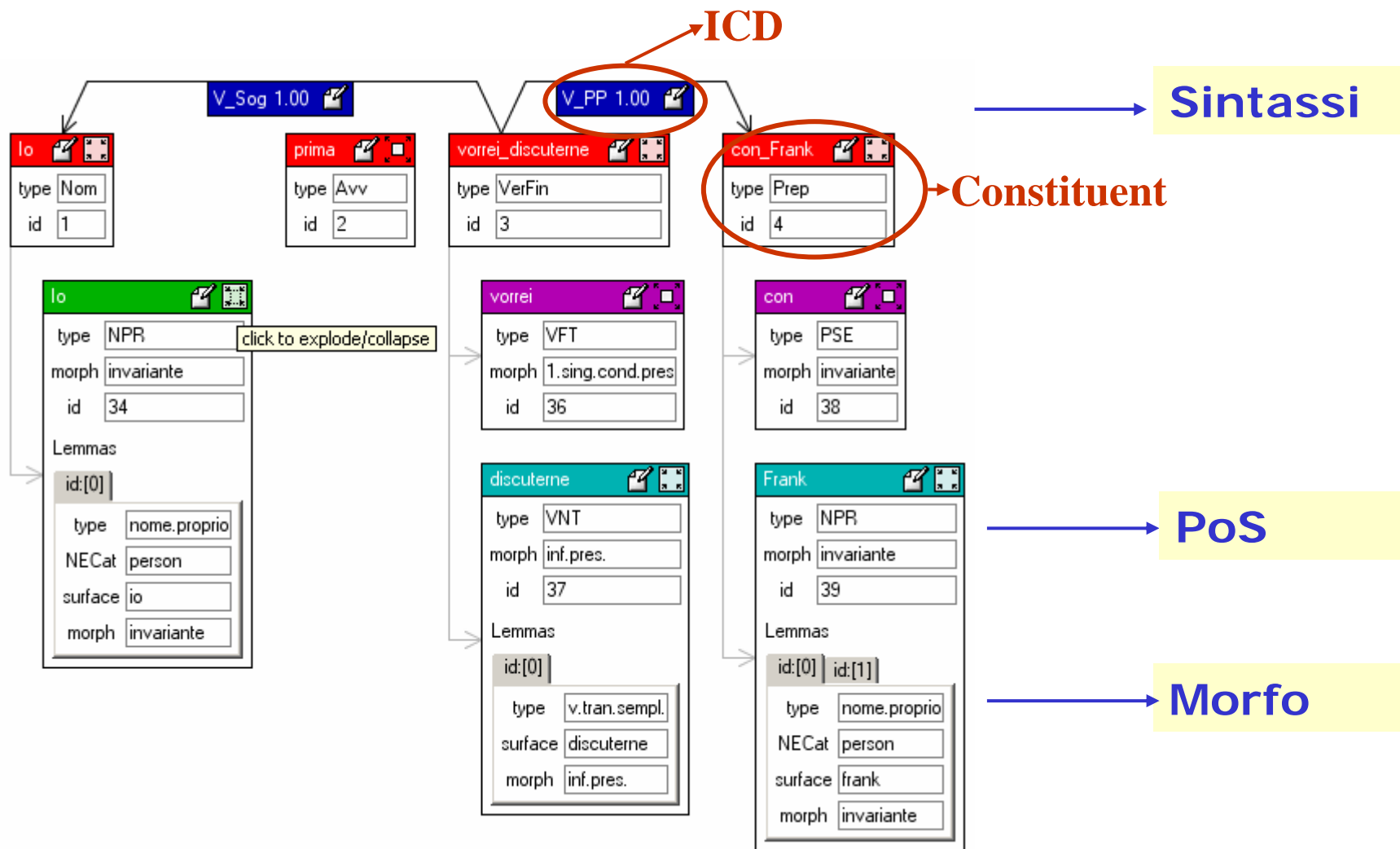
- Verb Subcategorization Analyzer

- Shallow Syntactic Analyzer

MORFOLOGIA

SINTASSI

Chaos: un architettura di NLP



Linguaggio Naturale e Linguaggi Formali

■ Cos'è il Linguaggio Naturale ?

- Strumento di comunicazione tra persone;
 - Fatti, idee e conoscenze sul mondo esterno ed interiore
 - Emozioni
 - Ordini
- E' **ambiguo!** (*“La vecchia porta la sbarra”*)

■ Cos'è un Linguaggio Formale ?

Dato un insieme di simboli Σ detto alfabeto, un linguaggio formale è un sottoinsieme di tutte le possibili concatenazioni dei simboli:

$$L \subseteq \Sigma^*$$

Un linguaggio formale **non è ambiguo** (una concatenazione di simboli ha una interpretazione univoca) ed esprime le sue regole in maniera canonica

Un elaboratore può riconoscere e generare solo Linguaggi Formali, attraverso l'utilizzo di modelli e algoritmi

Linguaggi Formali

ESEMPIO

$\Sigma = \{a, b\}$

$\Sigma^* = \{a, b, aa, ab, ba, bb, aaa, baba, baaab, \dots\}$

$L = \{ba, baa, baaa, baaaa, \dots\}$

Come definire il linguaggio L senza enumerare tutte le stringhe?

- **Modello procedurale:** automi, regole formali ...
- **Modello dichiarativo:** grammatiche

Linguaggi Formali e grammatiche

- Una **grammatica** può essere informalmente intesa come un insieme di regole per interpretare/generare un linguaggio formale
 - iniziando da un *simbolo iniziale*
 - applicando regole che indichino come rimpiazzare alcune sequenze di simboli con altre combinazioni di simboli (*derivazioni*)

ESEMPIO

$$L = \{ba, baa, baaa, baaaa, \dots\}$$

$$S \rightarrow Aa$$

$$A \rightarrow b$$

$$A \rightarrow Aa$$

Linguaggi Formali e grammatiche

- Una **grammatica** può essere informalmente intesa come un insieme di regole per interpretare/generare un linguaggio formale
 - iniziando da un *simbolo iniziale*
 - applicando regole che indichino come rimpiazzare alcune sequenze di simboli con altre combinazioni di simboli (*derivazioni*)

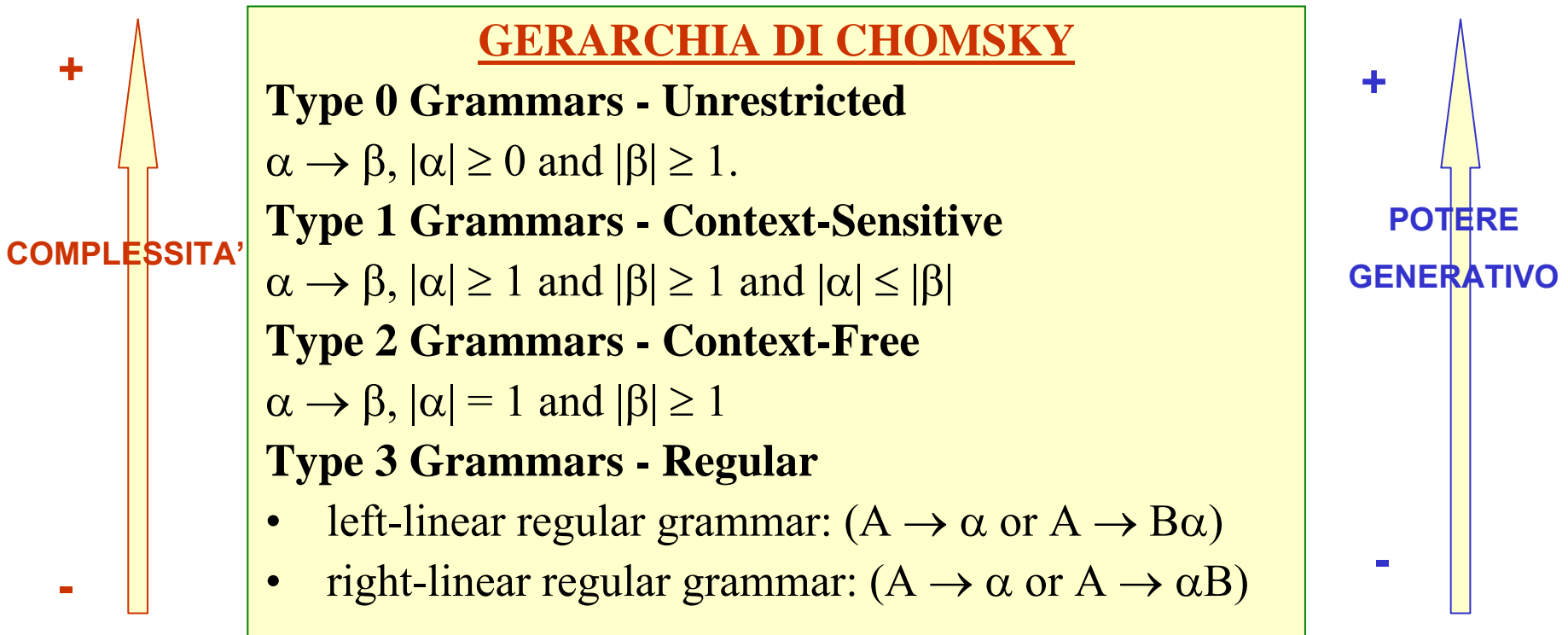
Formalmente:

Una grammatica è una quadrupla $(\mathbf{N}, \Sigma, \mathbf{S}, \mathbf{P})$ dove:

- \mathbf{N} è l'alfabeto dei simboli non-terminali
- Σ è l'alfabeto dei simboli terminali
- \mathbf{S} è elemento di \mathbf{N} detto *simbolo iniziale*
- \mathbf{P} è un insieme finito di produzioni, ovvero:
 - se \mathbf{V} è definito come $\mathbf{N} \cup \Sigma$, allora le produzioni di \mathbf{P} hanno la forma $\alpha \rightarrow \beta$, dove $\alpha \in \mathbf{V}^+$ $\beta \in \mathbf{V}^*$

Linguaggi Formali: complessità

- Un linguaggio formale può essere più o meno complesso, ed essere quindi computazionalmente più o meno esigente.
- La **gerarchia di Chomsky** è un tentativo di ordinare le grammatiche che generano i linguaggi in base alla loro complessità



Linguaggi Formali: complessità

- Le grammatiche sono **modelli dichiarativi**
- I corrispondenti **modelli procedurali** sono:

- **Type 0 Grammars - Unrestricted**
Turing Machine
- **Type 1 Grammars - Context-Sensitive**
Turing Machine
- **Type 2 Grammars - Context-Free**
Push-down automaton
- **Type 3 Grammars - Regular**
Finite State Automaton (FSA)

Linguaggi Formali: complessità

ESEMPIO

$$L = \{a^n b^n \mid n > 0\}$$

è un linguaggio **context-free**

$$L = \{a^n b^n c^n \mid n > 0\}$$

è un linguaggio **context-sensitive**

(dimostrabile con il Pumping Lemma)

$$L = \{ab, aabb, aaabbb, \dots\}$$

$$L = \{abc, aabbcc, aaabbbccc, \dots\}$$

DOMANDONE:

Il Linguaggio Naturale può essere rappresentato attraverso un Linguaggio Formale ?

Se sì, un Linguaggio Formale di quale complessità ?

Quanto è complesso il Linguaggio Naturale ?

Linguaggi Formali e Linguaggio Naturale

- ... dipende da *quale* linguaggio naturale
- un livello alto nella gerarchia vuol dire che il linguaggio naturale è strutturalmente complesso (*Tipo 0*)

▪ ITALIANO

- In generale, sembrerebbe “catturabile” da una Grammatica Regolare (*Tipo 3*)
 - **ECCEZIONE**: costrutti “**center-embedded**”. Ad esempio:

*“Moggi, Giraudò e Bettega erano
rispettivamente DG, amministratore
delegato e vicepresidente della Juventus”*

ha struttura $a^n b^n$

- Sembrerebbe quindi un linguaggio più complesso, ovvero **Context-Free** (*Tipo 2*)
- E' più complesso? No, perché sembra non avere costrutti del tipo $a^n b^n c^n$

Linguaggi Formali e Linguaggio Naturale

- L'italiano è quindi un linguaggio mediamente complesso (*Tipo 2*)
- E gli altri linguaggi naturali ?

▪ **Inglese:** Context-Free *Tipo 2*

▪ **Olandese:** Context-Sensitive *Tipo 1* (*Huybregt, 1976*)

presenta costrutti “**cross-serial**”. Ad esempio:

“dat Jan Marie Pieter Arabisch laat zien schrijven”
(**THAT JAN MARIE PIETER ARABIC LET SEE WRITE*)
‘that Jan let Marie see Pieter write Arabic’

Linguaggi Formali e Linguaggio Naturale

- La **sintassi** italiana e inglese e inglese sembrano quindi essere *Context-Free*
- La **morfologia**, invece, sembra essere ancora più semplice: può essere infatti rappresentata da *grammatiche Regolari*

QUINDI, NEL CORSO VEDREMO:

MORFOLOGIA	→	Automati a Stati Finiti (FSA)	<i>Tipo 3</i>
SINTASSI	→	Grammatiche Context-Free (CFG)	<i>Tipo 2</i>

Sommario

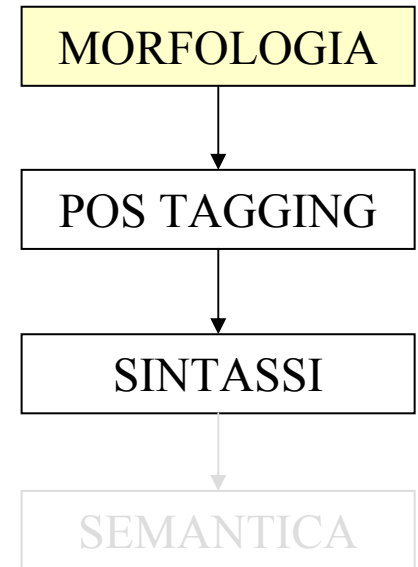
Morfologia

- **Automi a stati finiti (FSA)**

- FSA deterministici
- FSA non-deterministici (NFSA)
- Introduzione alla Morfologia
- FSA e Morfologia: *riconoscimento*

- **Trasduttori a stati finiti (FST)**

- Cosa sono
- FST e Morfologia: *parsing*



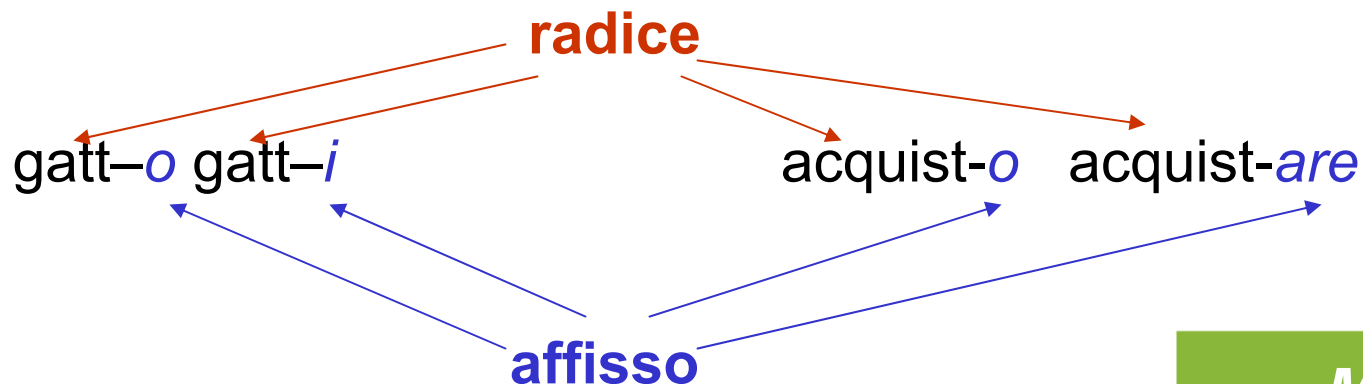
Morfologia ?

La **morfologia** è lo studio di come le parole sono costruite a partire da unità atomiche dette *morfemi*.

I **morfemi** sono le più piccole unità linguistiche che possiedono un significato. Ne esistono due classi:

- **Radice** → il morfema che dà il significato principale alla parola
- **Affisso** → il morfema che dà significato aggiuntivo alla parola

ESEMPIO



Analisi Morfologica: Automi a Stati Finiti

Strumenti per l'analisi morfologica :

- **Automi a Stati Finiti (FSA)** → Riconoscimento
- **Finite State Transducers (FST)** → Parsing

RICONOSCIMENTO : indica se una data parola in input è morfologicamente corretta o no (ad esempio *gatti* è corretta, *gattare* è scorretta)

PARSING : produce un'analisi morfologica della parola in input (ad esempio *gatti* → *gatto N PL*)

Sia FSA che FST sono di *tipo 3* nella gerarchia di Chomsky: l'analisi morfologica può essere quindi portata a termine con strumenti relativamente poco complessi

Analisi Morfologica: qualche esempio

Un analizzatore morfologico completo dovrebbe essere in grado riconoscere la classe (nomi, verbi, ecc.) delle parole e la loro morfologia:

house

house+N+SG

houses

house+N+PL

went

go+V+PastTense+123SP

play

play+V+Pres+Non3sg

played

played+A+VPap

miaow

miaow+Onom

Sommario

Morfologia

- **Automi a stati finiti (FSA)**
 - **FSA deterministici**
 - FSA non-deterministici (NFSA)
 - Introduzione alla Morfologia
 - FSA e Morfologia: *riconoscimento*
- **Trasduttori a stati finiti (FST)**
 - Cosa sono
 - FST e Morfologia: *parsing*

Automati a Stati Finiti (FSA)

Un automa a stati finti è un automa in grado di *riconoscere* o di *generare* una sequenza di simboli (*stringa*).

Formalmente: Un FSA è un grafo orientato i cui nodi sono detti **stati** e i cui archi sono detti **transizioni**

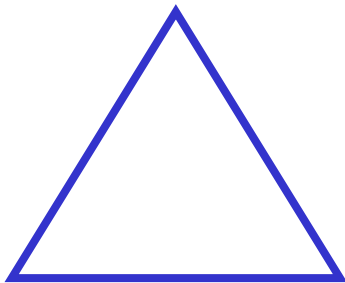
Caratteristiche principali:

- molto efficienti (tipo 3 nella ger. di Chomsky)
- facili da implementare
- Ogni FSA implementa una *espressione regolare*
- Ogni espressione regolare descrive un FSA
- Ogni FSA descrive un *linguaggio regolare*

Utilizzi principali in linguistica:

- Riconoscimento morfologico
- Fonetica
- Text-to-Speech

ESPRESSIONI REGOLARI

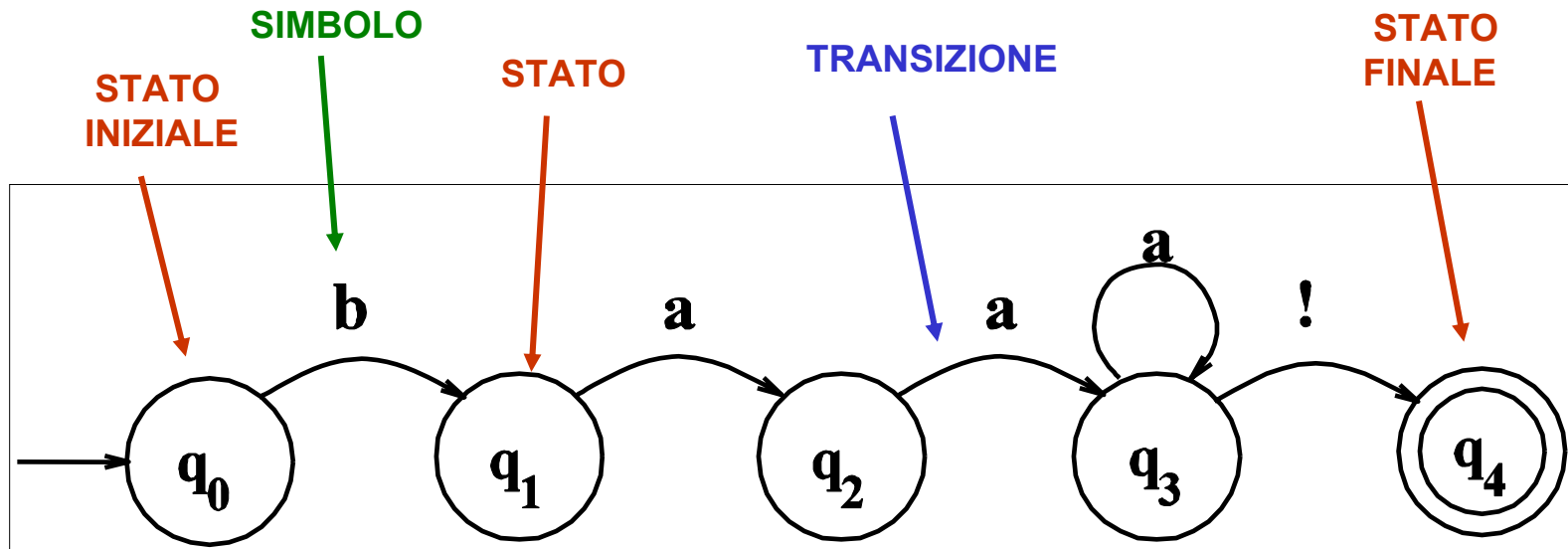


FSA

LINGUAGGI
REGOLARI

FSA: semplice esempio

FSA per riconoscere e generare sequenze di simboli appartenenti al linguaggio (regolare) delle caprette, descritto dall'espressione regolare: **/baa+!/**



FSA come riconoscitore: riconosce tutte le stringhe in input del tipo baa! , baaa! , baaaa! ,

FSA come generatore: genera tutte le stringhe del tipo baa! , baaa! , baaaa! ,

FSA: definizione formale

Un FSA è definito dai seguenti parametri:

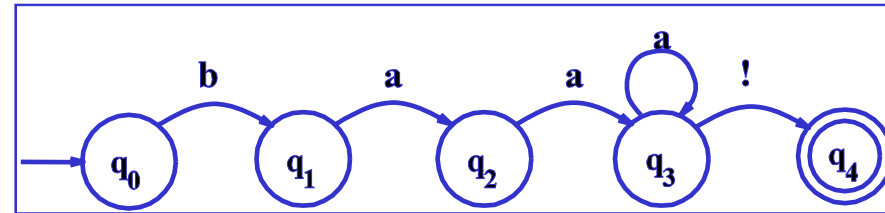
- Q : un insieme finito di N stati $q_0 \dots q_N$

- Σ : un alfabeto finito di simboli

- q_0 : lo stato iniziale

- F : un insieme di stati finali $F \subseteq Q$

- $\delta(q,i)$: funzione di transizione tra stati che restituisce un nuovo stato a partire da un dato stato e un simbolo in input



Un FSA può essere anche rappresentato attraverso una *state-transition table*

	Input		
State	b	a	!
0	1	0	0
1	0	2	0
2	0	3	0
3	0	3	4
4:	0	0	0

FSA e linguaggi formali

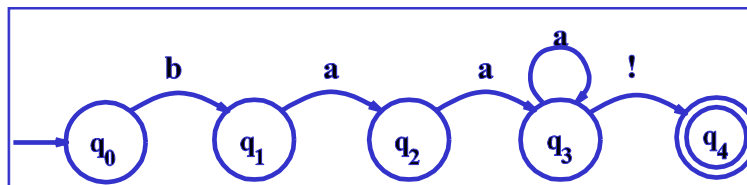
L'insieme delle stringhe *riconosciute* (o *generate*) da un FSA definiscono un **linguaggio formale**.

LINGUAGGIO FORMALE (L): insieme di stringhe composte da simboli appartenenti a un insieme finito di simboli Σ detto *alfabeto*

- L'insieme delle stringhe riconosciute da un FSA costituisce il **linguaggio accettato** dall'automa
- L'insieme delle stringhe generate da un FSA costituisce il **linguaggio generato** dall'automa
- Per un FSA, il linguaggio generato e quello accettato corrispondono

ESEMPIO

$\Sigma = \{a, b, !\}$



$L = \{baa!, baaa!, baaaa!, \dots\}$

FSA e linguaggi regolari

Un FSA (o un'espressione regolare) può definire un sottoinsieme particolare dei linguaggi formali, i **linguaggi regolari**

LINGUAGGIO REGOLARE (L):

Dato un alfabeto Σ :

- L'insieme vuoto \emptyset è un linguaggio regolare
- $\forall a \in \Sigma \cup \varepsilon$, $\{a\}$ è un linguaggio regolare
- Se $L1$ e $L2$ sono linguaggi regolari, allora lo sono anche:
 - $L1 \cdot L2 = \{xy \mid x \in L1, y \in L2\}$, concatenazione di $L1$ & $L2$
 - $L1 \cup L2$, unione di $L1$ e $L2$
 - $L1^*$, la *Kleene closure* di $L1$

FSA e linguaggi regolari

LIMITI: I linguaggi regolari hanno un basso **potere generativo** (*tipo 3*)

- Ad esempio, dato l'alfabeto $\Sigma=\{a,b\}$, nessun FSA può generare stringhe del tipo $a^n b^n$
- Gli FSA modellano quindi bene fenomeni linguistici semplici come:
 - *Morfologia*
 - *Fonetica*
- Gli FSA non possono modellare fenomeni linguistici complessi come:
 - *Sintassi*

ESEMPIO (english)

The cat likes tuna fish

The cat the dog chased likes tuna fish

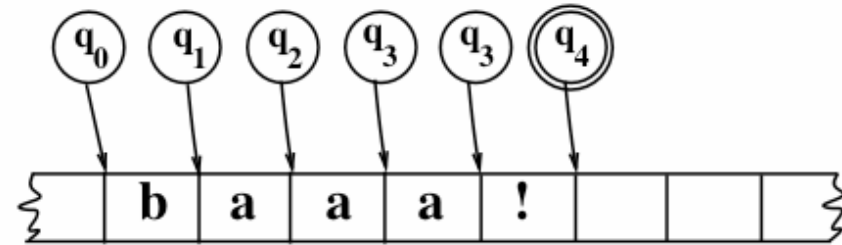
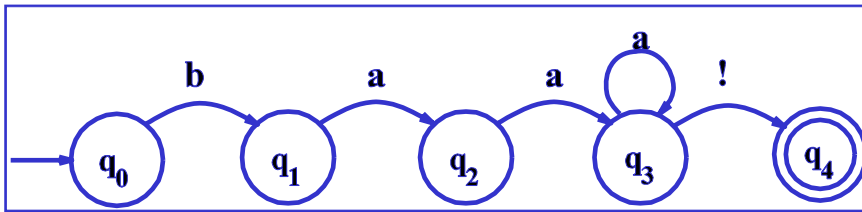
The cat the dog the rat bit chased likes tuna fish

The cat the dog the rat the elephant admired bit chased likes tuna fish

$L = x^n y^{n-1}$ likes tuna fish.

FSA come riconoscitori

SCOPO: Data una stringa in input verificare se essa appartiene al linguaggio formale definito dall'automa.



ALGORITMO DI RICONOSCIMENTO

indice \leftarrow inizio stringa in input

Stato-corr \leftarrow q_0

WHILE (input)

IF vuota(trans-table[stato-corr,stringa[indice]])

return reject

ELSE

stato-corr \leftarrow trans-table[stato-corr,stringa[indice]]

indice \leftarrow indice +1

IF (stato-corrente è stato finale)

return accept

ELSE return reject

State	Input		
	b	a	!
0	1	0	0
1	0	2	0
2	0	3	0
3	0	3	4
4:	0	0	0

1. Disegnare (se esistono) gli FSA che riconoscono/generano i seguenti linguaggi regolari:

$L_a = \{ac, abc, abbc, abbbc, \dots\}$

$L_b = \{ac, abb\}$

$L_c = \{ac, acdc, acdcdc, \dots\}$

$L_d = \{ac, ab, acdc, abdb, abdc, acdb, acdcdc, \dots\}$

2. Scrivere l'espressione regolare che descrive l'FSA
3. Scrivere le transition table relative agli FSA

1. Scrivere un FSA che riconosca espressioni “monetarie” del tipo:
“uno euro”, “due euro”, “venti tre euro”, “venti uno euro dieci cent”, “trenta due euro trenta quattro centesimi”, “due cent”....

Il cui vocabolario sia $\Sigma = \{\text{uno, due, tre, ...dieci, venti, ..., euro, cent}\}$

2. Scrivere la relativa state-transition table
3. Scrivere l'espressione regolare associata all'FSA

1. Scrivere un FSA che riconosca sintagmi nominali per l'Inglese, ovvero:
 - nomi propri (es. *“John”*)
 - nomi comuni preceduti da articoli e eventualmente da un numero variabile di aggettivi (es. *“the new yellow table”*)
 - combinazione di sintagmi dei due punti precedenti preceduti da preposizioni (es. *“the new yellow table of John”*)
2. Scrivere la relativa state-transition table