30 Maggio 2006

Morfologia & Part of Speech

Marco Pennacchiotti

pennacchiotti@info.uniroma2.it

Tel. 06 7259 7717

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

Info

- **Cambio Aula**
  - Giovedì → **Aula 3** (11.30 – 13.15)

- **Download Chaos**
  - http://ai-nlp.info.uniroma2.it/external/chaosproject/protectedDownload/download.html

- **Progetto**
  - Formare gruppi 3-5 persone
  - Inviare e-mail per iscriversi entro **giovedì 1 Giugno**
  - http://ai-nlp.info.uniroma2.it/pennacchiotti/teaching/

Programma

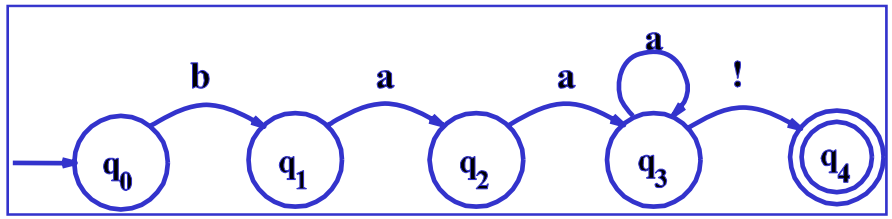
- **Breve introduzione all'NLP**
  - Linguaggi Naturali e Linguaggi Formali
  - Complessità
- **Morfologia**
  - *Teoria:* Morfologia del Linguaggio Naturale
  - *Strumenti:* Automi e Trasduttori
  - *Analisi Morfologica:* con automi e trasduttori
- **Part of Speech Tagging**
  - *Teoria:* Le classi 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

Info

FSA: definizione formale

Un FSA è definito dai seguenti parametri:

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



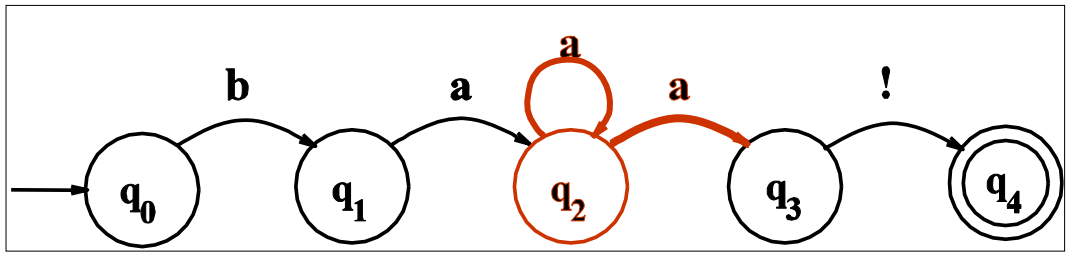
- *Σ* : un alfabeto finito di simboli
- q_0 : lo stato iniziale
- *F* : un insieme di stati finali $F \subseteq Q$
- $\delta(q,i)$: funzione di transizione tra stati che restituisce un nuovo stato a partire da un dato stato e un simbolo in input

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

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

FSA non-deterministici (NFSA)

Un automa è detto non-deterministico se ha due archi uguali uscenti dallo stesso stato.



Quindi:

- *Deterministico* vuol dire che ad ogni stato può essere presa una sola decisione
- *Non-Deterministico* vuol dire che ad ogni stato si può scegliere tra più decisioni

Equivalenza tra FSA e NFSA

- Un NFSA può essere sempre convertito in un FSA *equivalente* (che definisce cioè lo stesso linguaggio)
- NFSA e FSA hanno quindi lo stesso potere di riconoscimento/generazione
- L'FSA equivalente di un NFSA ha sempre più stati dell'NFSA

Morfologia: *definizioni*

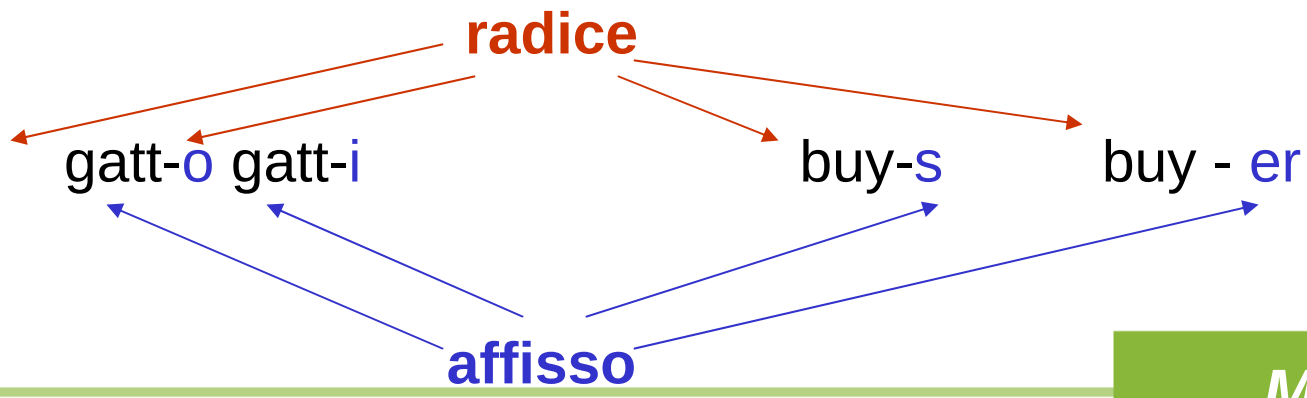
La morfologia (*morphology*) è lo studio di come le parole sono costruite a partire da unità atomiche dette *morfemi*.

I morfemi (*morphemes*) sono le più piccole unità linguistiche che possiedono un significato. Possono essere divisi in due classi:

- Radice (*stem*)→ il morfema che dà il significato principale alla parola

- Affisso (*affix*)→ particelle apposte alla radice che ne completano il significato e la funzione grammaticale

ESEMPIO



Morfologia: *definizioni*

La morfologia può essere divisa in due parti principali

- Inflectional Morphology: combinazione di una radice con un affisso che risulta in una parola (*forma flessa*) della *stessa classe* (nome, verbo, aggettivo, ecc..) con una funzione grammaticale specifica

  - *cat* (nome sing) → *cat-**s*** (nome plur)
  - *cut* (verbo base) → *cut-**ting*** (verbo progressivo)

- Derivational Morphology: combinazione di una radice con un affisso che risulta in una parola di una *classe diversa.* Il significato della nuova parola non è facilmente prevedibile

  - *trasporto* (nome) → *trasport-**abile*** (aggettivo)
  - *computerize* (verbo) → *computeriz-ation* (nome)

Morfologia inglese

Caratteristiche:

- Concatenative morphology
- Non-agglutinative (al più 4 o 5 affissi)
- Una parola può avere più affissi:
  - *Prefissi:* ***un***-*certain*
  - *Suffissi:* *eat*-***s***
  - Combinazioni: ***un***-*clear*-***ly***
- Inflectional morphology: semplice, applicata solo a *nomi, verbi* e *aggettivi*
- Derivational Morphology: complessa

Inflectional Morphology

NOMI:

Due solo inflessioni:

- Plurale: *cat* → *cat-s* *thrush* → *thrush-es*
- Possessivo: *dog* → *dog's* *children* → *childrens'*

VERBI:

Quattro forme morfologiche:

- stem: *walk*
- *s* form: *walk* → *walk-s*
- past form: *walk* → *walked*
- *ing* form: *walk* → *walking*

- **Irregolari:** **(ca. 250)** Parole che non seguono le regole morfologiche (Esempio: *mouse* → *mice* *go* → *goes, going, went*). La maggior parte dei nomi e verbi inglesi sono regolari

- La classe dei verbi regolari è **produttiva** : una nuova parola della lingua è automaticamente inclusa nella classe (Esempio: *fax* → *faxes,faxing, faxed*)

Quali strumenti usare ?

- *Lessico esteso*

  - Un lessico (lista di parole) che contiene tutte le parole della lingua in tutte le forme flesse
  - Spreco di spazio e non è *produttivo*!

- *Lessico ridotto + Automi*

  - La morfologia è generalmente produttiva (gran parte delle parole segue le regole morfologiche per formare le forme flesse)
  - Conviene quindi utilizzare:
    - Lessico contenente solo radici e affissi (ed eventualmente irregolarità)
    - Implementazione delle regole morfologiche in un dispositivo
    - FSA sono semplici dispositivi per implementare tali regole

FSA: *riconoscimento*

Un FSA può essere utilizzato per riconoscere se una parola è ammissibile in una lingua

Cosa serve:

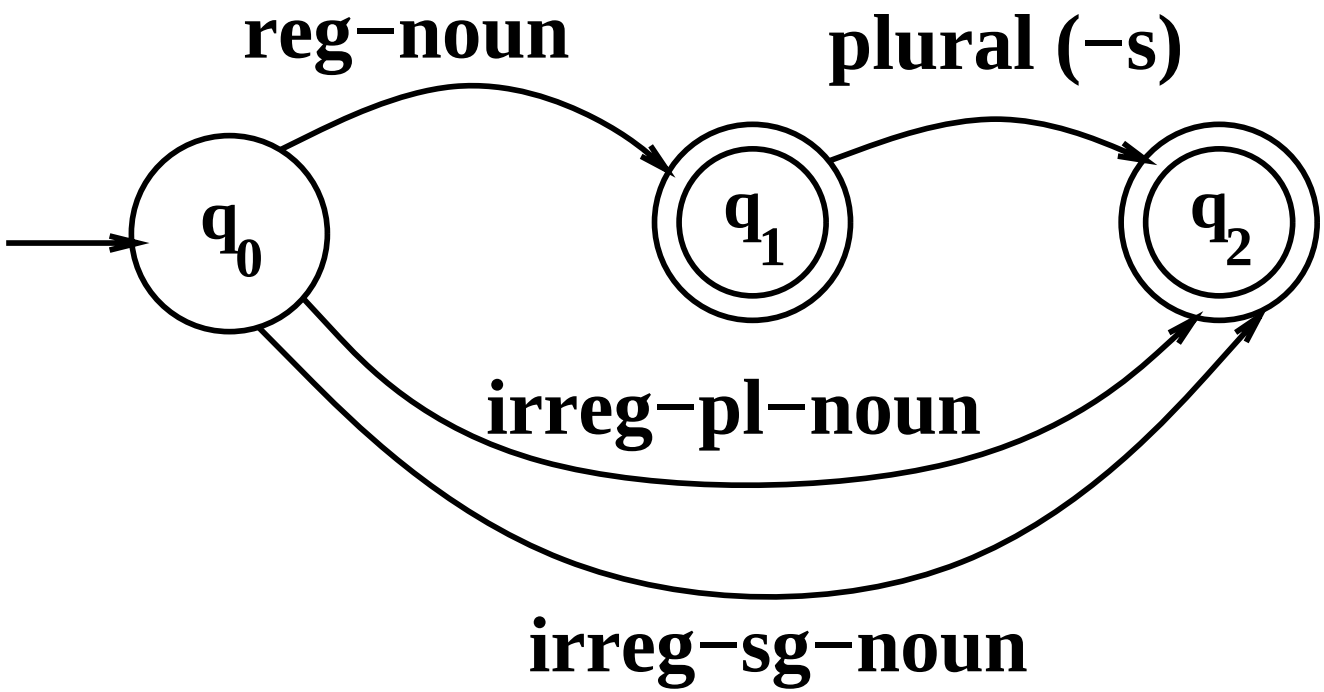
1. Lessico: lista di radici ed affissi della lingua
   - Esempio: [cat,dog,cut,go,…,-s,-ed,-ation,-able,…,un-,dis-]

2. Regole Morfologiche (*morphotactics*): le regole di costruzione dei morfemi
   - Esempio: Plurale inglese: radice + -s

3. Regole Ortografiche: cambiamenti che occorrono in una parola quando due morfemi si combinano
   - Esempio: *city* → *cities*

NOMI: *regole morfologiche*

FSA per modellare l'inflessione plurale per nomi regolari ed irregolari

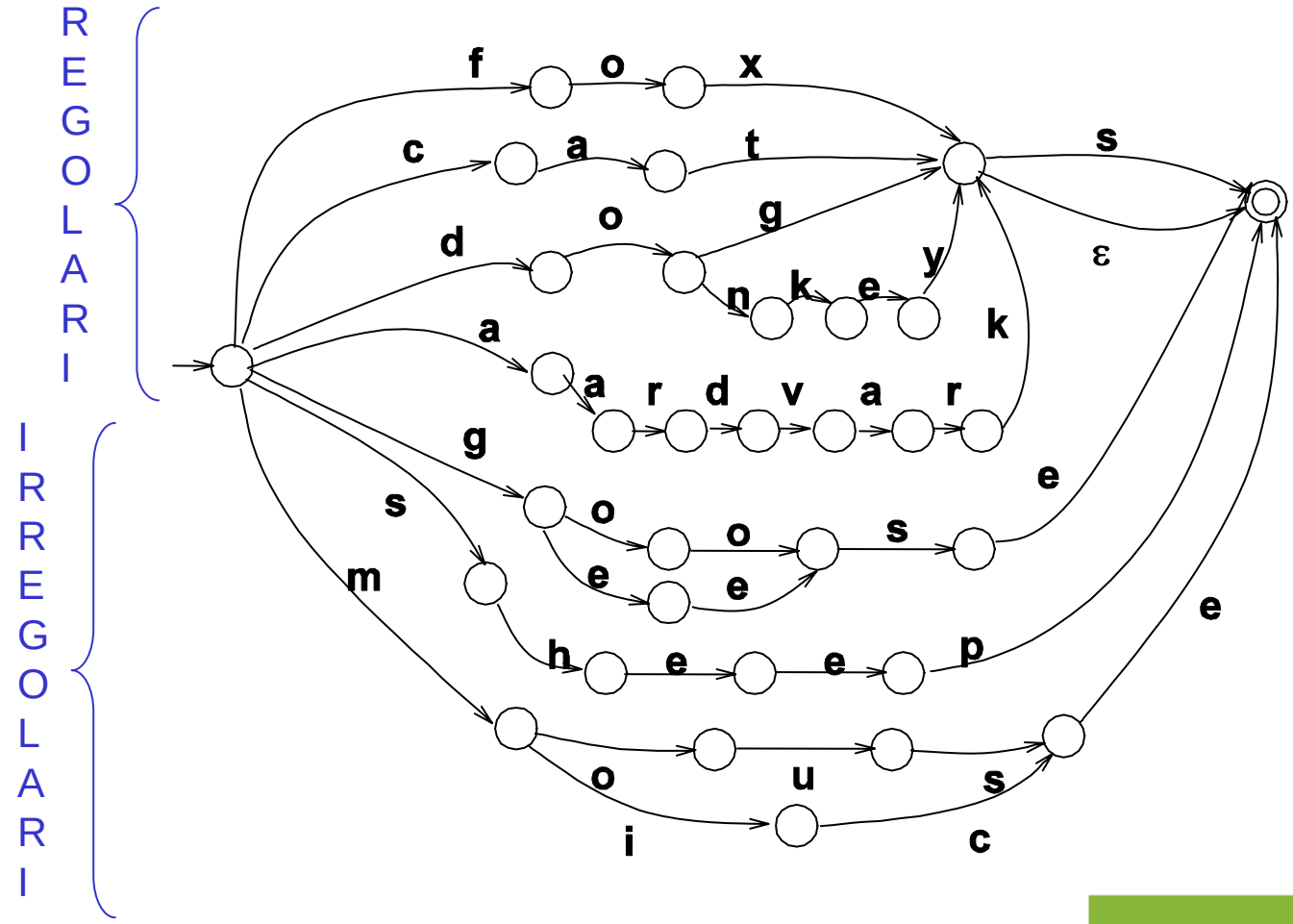


Come modellare i nomi (regolari ed irregolari) nell'FSA?

Ovvero: Come si può integrare il lessico?

NOMI: *regole morfologiche + lessico*

Integrazione del lessico dei nomi regolari ed irregolari



FSA per la morfologia

1. **Scrivere un FSA che riconosca la morfologia derivazionale degli aggettivi inglesi, ovvero:**
   - ***Un aggettivo può avere come prefisso negante* "un-"**
   - ***Un aggettivo può avere forma comparativa, superlativa e avverbiale (rispettivamente i suffissi* –er,-est,-ly*)***

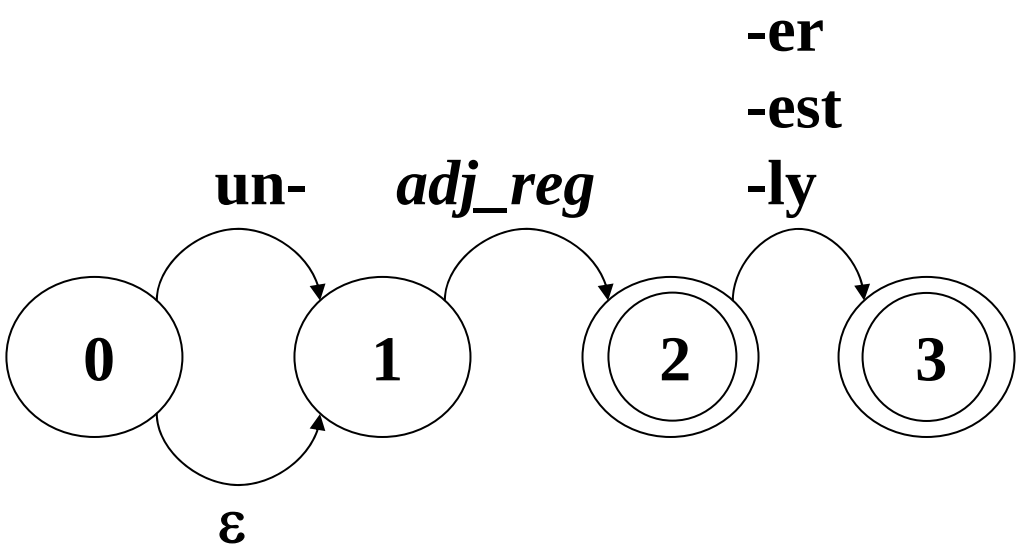
2. **Aggiungere all'FSA il seguente fatto:**
   - ***Esistono alcuni aggettivi "irregolari" che non possono prendere* "un-" *e* "-ly" (es: *big, cool*)**

3. **Integrare il lessico: regolari: *"clear, happy"*, irregolari: *"big,cool"***

ESERCIZIO 4

Soluzione esercizio 4

1

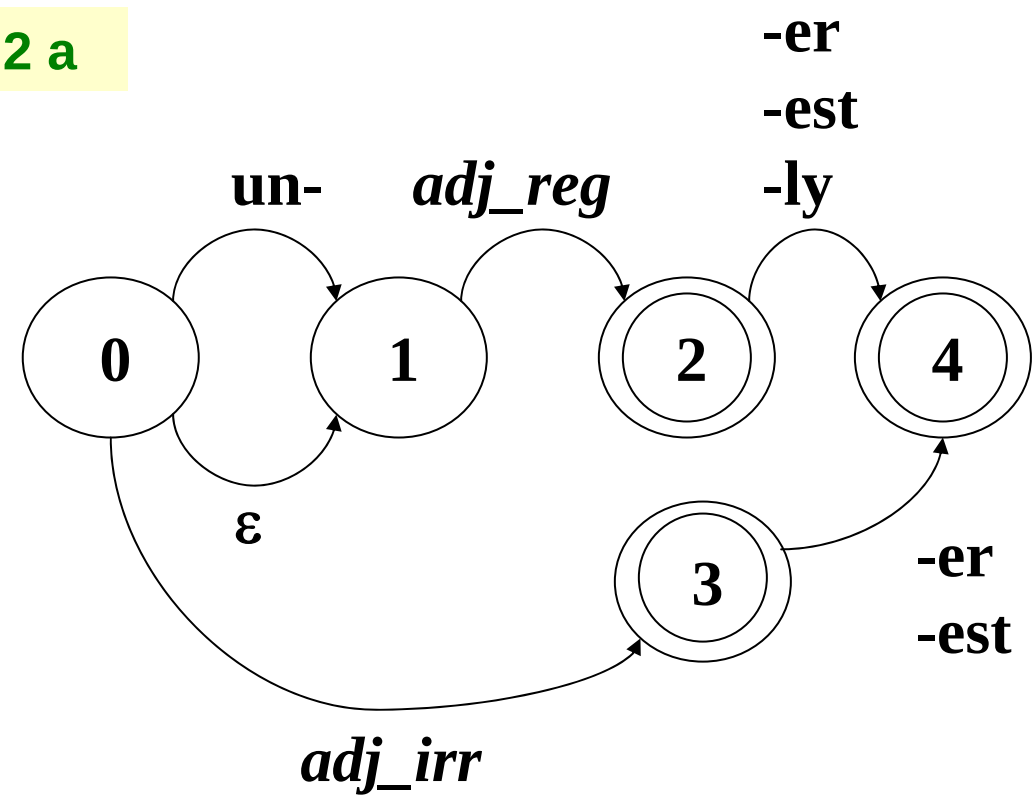


LESSICO

adj_reg
Clear
Happy

Soluzione esercizio 4

2 a

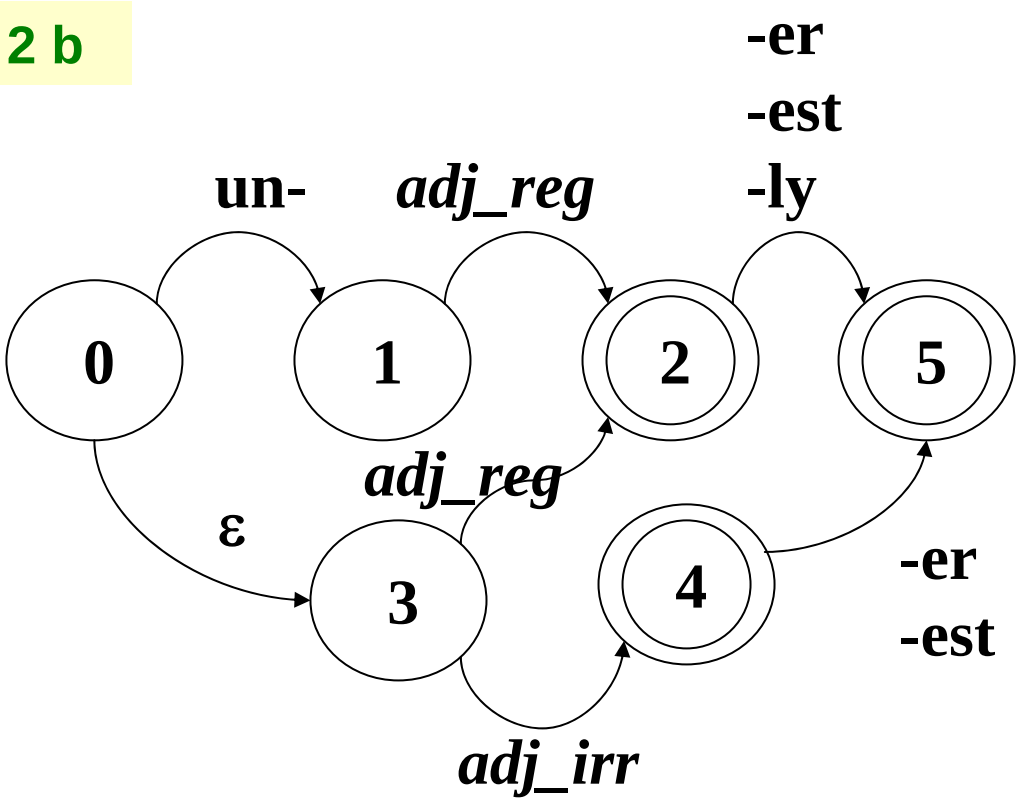


LESSICI

adj_reg	*adj_irr*
Clear	Big
Happy	cool

Soluzione esercizio 4

2 b



LESSICI

adj_reg	*adj_irr*
Clear	Big
Happy	cool

Dal Riconoscimento al Parsing

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

→ **FSA**

PARSING/GENERAZIONE :

- parsing: produce un'analisi morfologica della parola in input: data la parola in input viene restituita la sua struttura

cats → *cat* +*N* +*PL*

- generazione: data una struttura morfologica in input, produce una *forma superficiale* (parola)

cat +*N* +*PL* → *cats*

→ **FST**

Trasduttori a Stati Finiti (*FST*)

I *Trasduttori* sono automi a stati finiti con due nastri *A* e *B*

Ad es, può *leggere* da un nastro (ad es. "*cats*") e *scrivere* sull'altro ("*cat + N + PL*")

Quattro modalità di utilizzo dell' FST:

- riconoscitore: riceve in input una coppia di stringhe su *A* e *B,* e restituisce *accept* se essa appartiene al linguaggio delle coppie
  - *cats, cat+N+PL → accept*
- produttore: restituisce coppie di stringhe appartenenti al linguaggio su *A* e *B*
  - Output: tutte le parole del lessico con la loro struttura
- traduttore: riceve in input una stringa su *A (o B*) e ne restituisce un'altra su *B* (o *A*)
  - *cats → cat+N+PL (PARSING)*
  - *cat+N+PL → cats (GENERAZIONE)*

FST: definizione formale

Un FST è definito dai seguenti parametri:

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

- *Σ* : un alfabeto finito di simboli complessi. Ogni simbolo complesso è una coppia di simboli ***i:o*** appartenenti rispettivamente agli alfabeti **I** e **O** ($\Sigma \subseteq I \times O$)

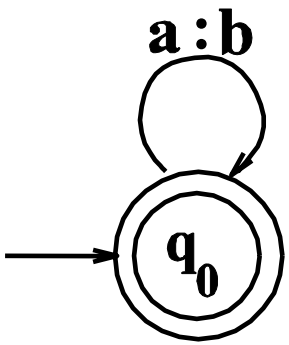
- q_0 : lo stato iniziale

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

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

---

ESEMPIO



- *riconosce* tutte le coppie di stringhe in cui una ha tutte *a* e l'altra uno stesso numero di *b*

- *produce* stringhe di *a* su un nastro e stringhe di *b* sull'altro, con la stessa lunghezza

- *traduce* stringhe di *a* in input in stringhe di *b* della stessa lunghezza in output, e viceversa

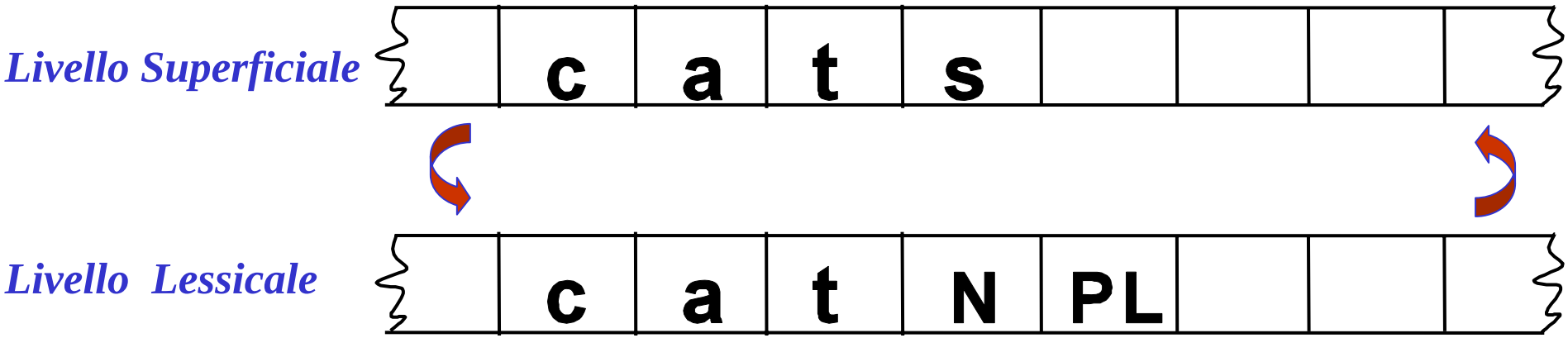
Sommario

Strumenti per la 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***

FST e morfologia: *parsing*

OBIETTIVO:

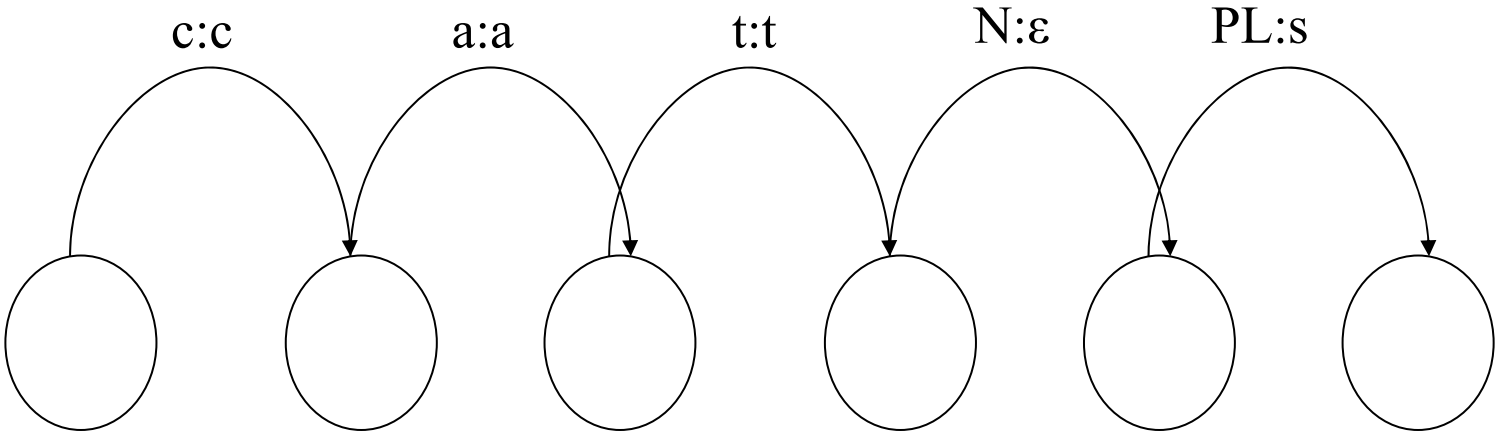


Passare da un livello superficiale ad un livello lessicale e viceversa, utilizzando un FST in funzione di *traduttore*:

- *cats → cat+N+PL (PARSING)*
- *cat+N+PL → cats (GENERAZIONE)*

FST e morfologia: *parsing*

OBIETTIVO:



Passare da un livello superficiale ad un livello lessicale e viceversa, utilizzando un FST in funzione di *traduttore*:

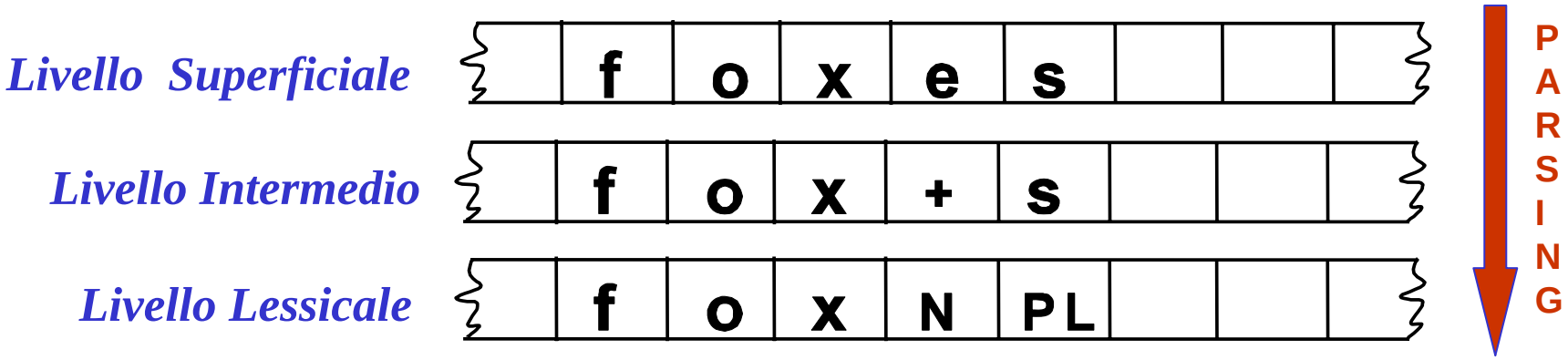
- *cats → cat+N+PL (PARSING)*
- *cat+N+PL → cats (GENERAZIONE)*

Analisi morfologica *a due stadi*

Dal livello superficiale al livello lessicale (PARSING)

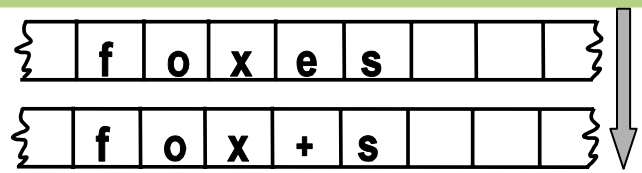
Sono necessari due stadi (→ due trasduttori)

1. IDENTIFICAZIONE DEI MORFEMI: Data la parola in input sul nastro *A,* il trasduttore la divide su *B* nei morfemi costituenti (radice + affissi)
2. IDENTIFICAZIONE DELLA STRUTTURA: Dati i morfemi costituenti sul nastro *A*, il trasduttore identifica la categoria della radice e il significato degli affissi



Stadio 1: *Identificazione dei morfemi*

ESEMPIO: nomi singolari/plurali



Obiettivo

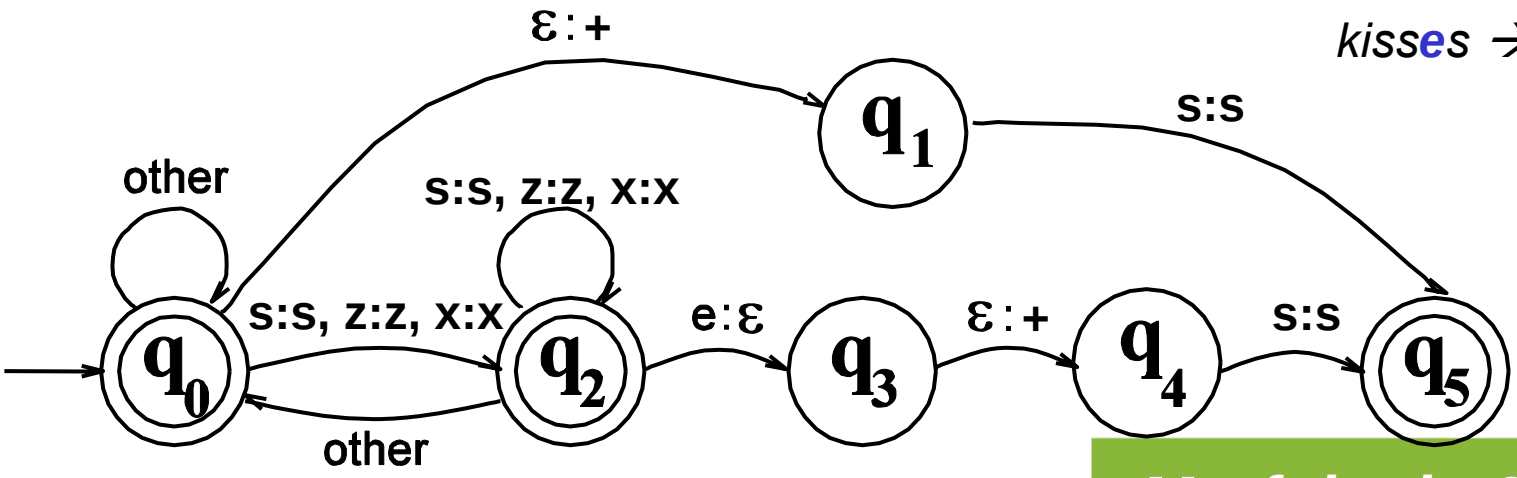
- Rappresentare con un FST le *regole ortografiche* della lingua per i nomi regolari e irregolari
- Input: *cats* Output: *cat+s*

Regola *e-insertion*

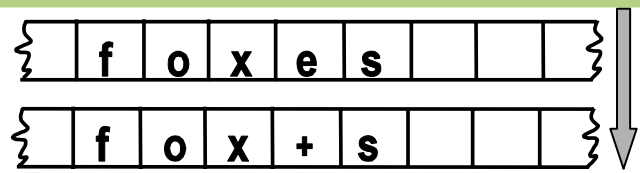
cats → *cat+s*

foxes → *fox+s*

kisses → *kiss+s*



Stadio 1: *non è così facile …*



Problemi

- Il trasduttore gestisce solo la regola della e-insertion, e non altri casi:
  - Regola y-replacement: *berries* → *berry +s* *(berrie +s)*
  - Regola raddoppio consonanti: *beg* → *begging*
  - *Ecc. ecc.*
- Bisogna quindi implementare più regole ortografiche, nello stesso trasduttore, o in trasduttori paralleli!
- Ambiguità locale: *foxes* produce due forme di cui solo la prima è corretta: *fox+s* , *foxe+s, foxes.*

Identificazione dei morfemi

A

1. **Scrivere l' FST che modelli la morfologia dei nomi singolari/plurali prendendo in considerazione la regola ortografica dell' *y-replacement:***
   - ***-y* cambia in *–ie* prima della –s**
   - **ES: berry → berries**

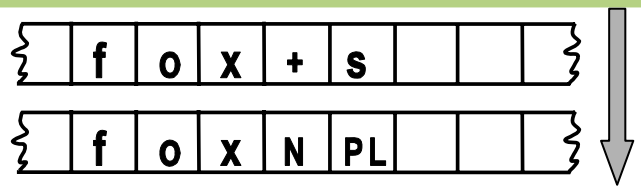
B

1. **Scrivere l' FST che modelli la morfologia dei verbi presente/passato prendendo in considerazione la regola ortografica della *k-insertion*, sapendo che:**
   - **In generale la forma passata si forma dal presente apponendo come suffisso la particella *–ed* (ES: *press→press-ed*)**
   - **La *k-inserion* prevede che ai verbi terminanti in *vocale+c* sia aggiunta la *k* (ES: *panic → panic + k +ed*)**

ESERCIZIO 5

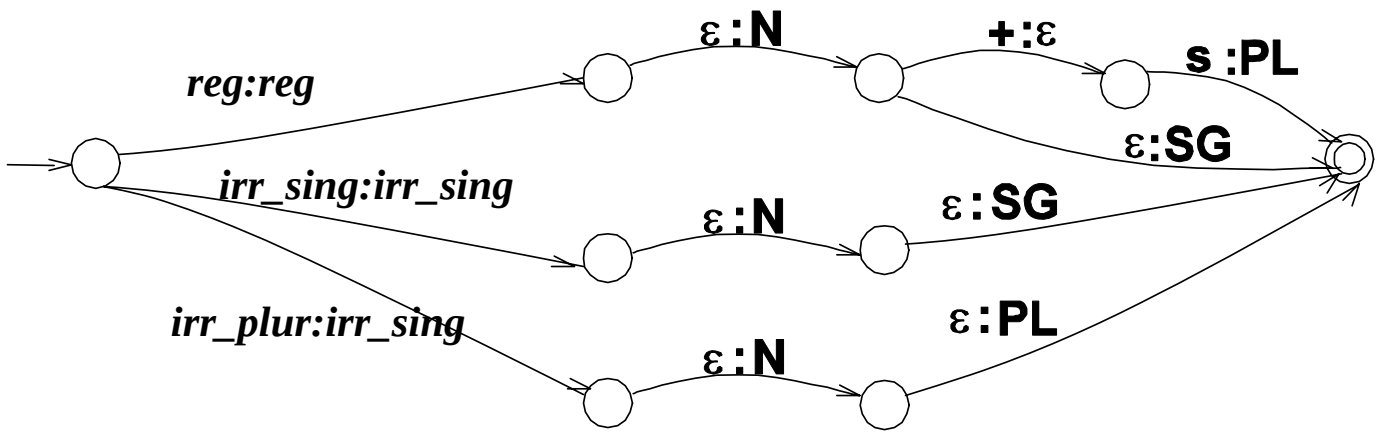
Stadio 2: *Identificazione della struttura*

ESEMPIO: nomi singolari/plurali



Obiettivo

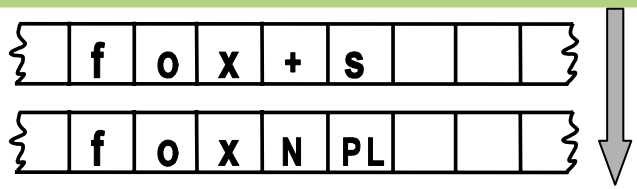
- Rappresentare con un FST le *regole morfologiche* della lingua per i nomi regolari e irregolari
- Input: *cat+s, mouse, mice* Output: *cat N PL, mouse N SG, mouse N PL*

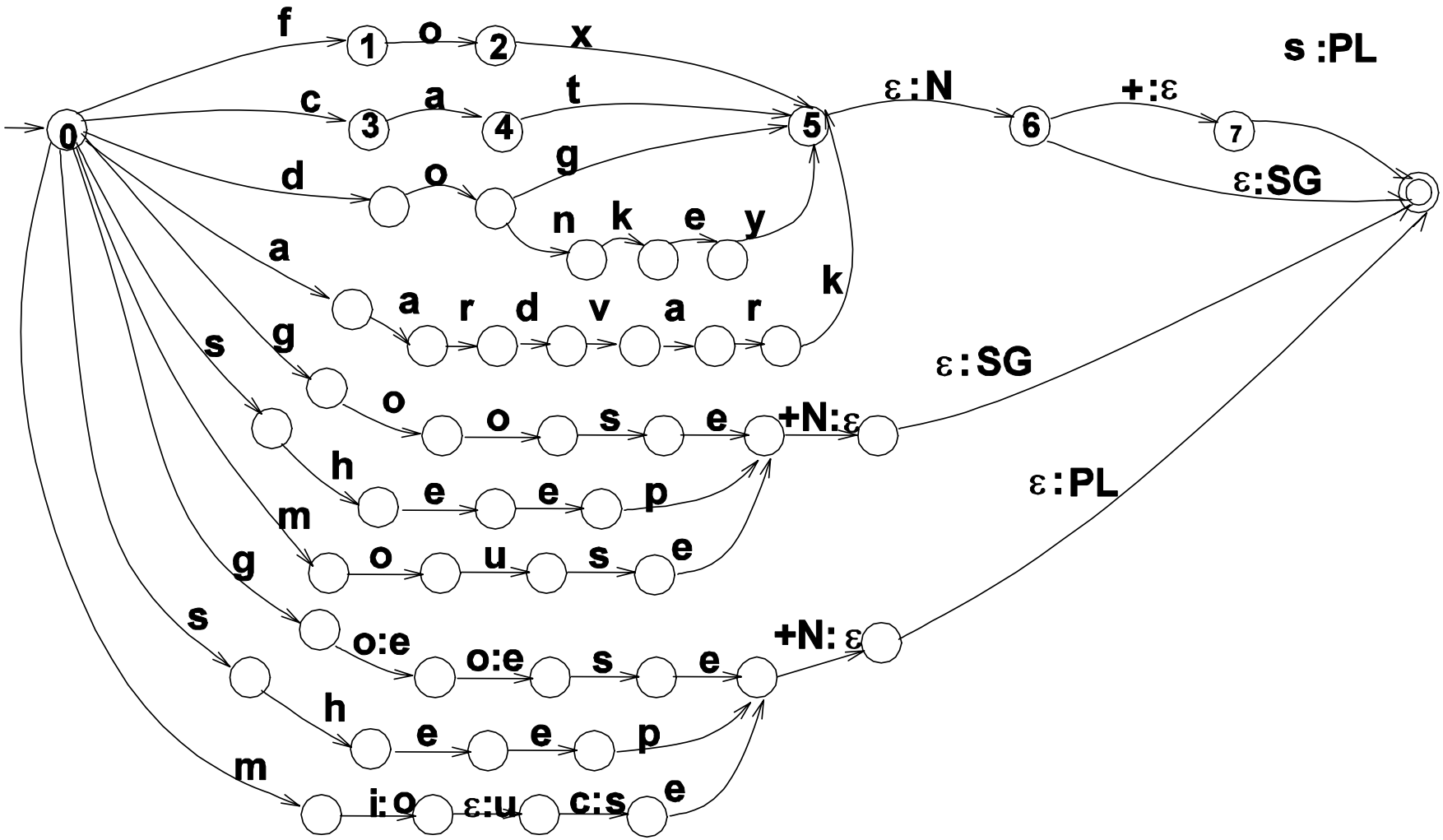


Bisogna aggiungere il lessico !

Stadio 2: *Identificazione della struttura*

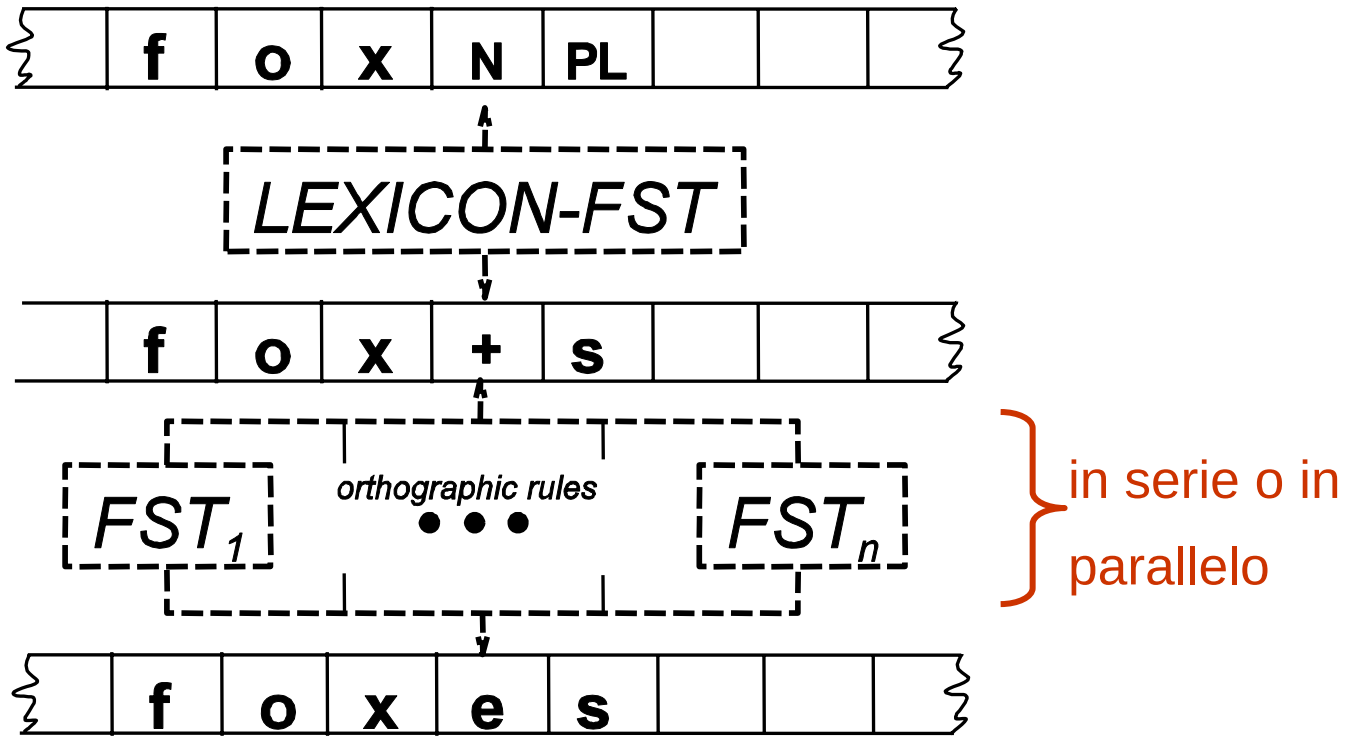
ESEMPIO: nomi singolari/plurali





Stadio 1+2: *Combinare lessico e regole*

- E' possibile combinare i due stadi mettendo *in cascata* (in serie) i due trasduttori: l'output dell'uno sarà l'input dell'altro



- Oppure, è possibile fondere i due trasduttori, attraverso un'operazione di *intersezione*

Vantaggi e problemi

Vantaggi degli FST

- Computazionalmente efficienti
- Semplici
- Doppio uso: parsing e riconocimento
  - Qual'è la morfologia di *foxes* ? *foxes* → *fox+N+PL*
  - Qual è il plurale di *fox* ? Fox+N+PL → foxes

Problemi

- Laborioso costruire e codificare un trasduttore per gestire ogni regola ed eccezione:
  - Soluzione: Tool automatici di traduzione *regola→FST*
- Ambiguità globale: *kisses* può essere sia verbo che nome
  - *kisses→ kiss+N+PL kisses→kiss+V+3SG*
  - Per disambiguare sono necessarie risorse esterne (non morfologiche), ad esempio il contesto sintattico

Tool per il parsing morfologico

CHAOS *Morpho-Analizer* [/data/KB/it/db_morfo]

- Basato su lessico esteso (*dizionari*)
  - il lessico esteso è precedentemente costruito ed arricchito utilizzando *ALMA* (generatore morfologico basato su FST ed implementato in Prolog)
  - Nessun utilizzo di FST “on-line”
- Utilizza diversi dizionari:
  - Dizionario principale (IT: 115.000 parole, EN: 35.000 parole)
  - Dizionario nomi propri (IT: 80.000 nomi, EN: 52.000 nomi)
  - Dizionario terminologico (IT: 5000 termini)

morphg [www.cogs.susx.ac.uk/research/nlp/carroll/morph.html]

- Tool per la *inflectional morphology* basato su FST
- Accuracy 99.9% in parsing/generazione

Qualche link…

DEMO di automi per la morfologia

- http://www.xrce.xerox.com/competencies/content-analysis/demos/english
- http://www.xrce.xerox.com/competencies/content-analysis/demos/italian

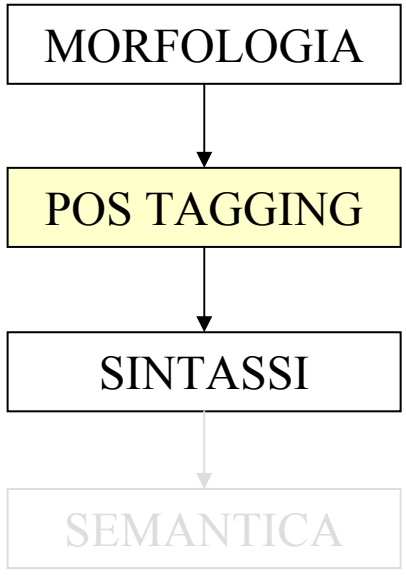
TOOLS per costruire e gestire automi

- http://www.research.att.com/sw/tools/fsm/

Sommario

Part of Speech Tagging

- **Part of Speech Tagging**
  - Cos'è
  - Part of Speech
  - Part of Speech Tagging
    - » Rule-bales
    - » Stochastic
    - » Misto
  - Prestazioni



Part Of Speech (POS)

Part of Speech (*classi morfologiche*)

Categoria morfo-sintattiche cui una parola appartiene

Categorie principali

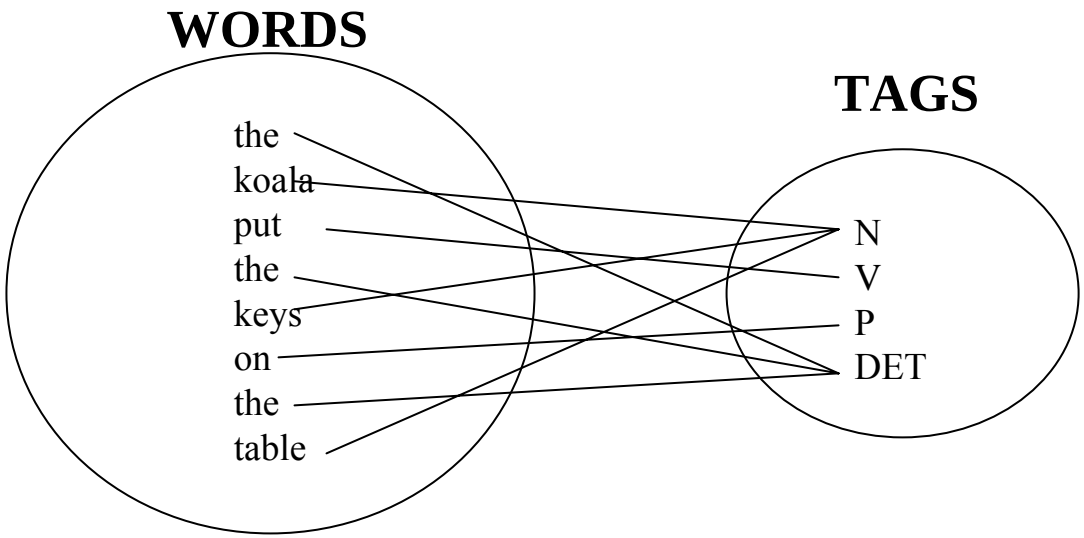
- Nomi, verbi, aggettivi, avverbi, articoli, pronomi, congiunzioni …

Esempio

N	*noun*	chair, bandwidth, pacing
V	*verb*	study, debate, munch
ADJ	*adj*	purple, tall, ridiculous
ADV	*adverb*	unfortunately, slowly,
P	*preposition*	of, by, to
PRO	*pronoun*	I, me, mine
DET	*determiner*	the, a, that, those

POS Tagging

Processo di assegnazione della Part of Speech ad ogni parola di un corpus (insieme di documenti)



MODELLI

- Modelli a regole
- Modelli probabilistici
- Modelli misti

A cosa sere il POS Tagging?

STEMMING IN INFORMATION RETRIEVAL

- Sapendo la classe di una parola, si possono produrre le forme flesse
- Ricerca effettuata utilizzando tutte le forme flesse

PARSING

- La classe di una parola può aiutare a predire la struttura sintattica di una frase
- ES: un pronome possessivo è sempre seguito da un nome

INFORMATION EXTRACTION / QUESTION ANSWERING

- Si estrae o si restituisce solo l'informazione di una data classe
- ES: si può cercare un luogo (nome) o una azione (verbo)

Quale relazione con la morfologia ?

ANALISI MORFOLOGICA

- Data una parola, trovare le sue interpretazioni morfologiche
- Può essere presente ambiguità:

  ES: talks → talk+s → talk V 3PS

  → talk N PL

POS TAGGING

- Data una parola, trovare la sua unica interpretazione morfologica
- Analisi morfologica + disambiguazione
- → metodi dell'analisi morfologica (es. FST) + algoritmi di disambiguazione

Sommario

Part of Speech Tagging

- **Part of Speech Tagging**
  - Cos'è
  - **Part of Speech**
  - Part of Speech Tagging
    - » Rule-bales
    - » Stochastic
    - » Misto
  - Prestazioni

Part Of Speech

Tradizionalmente, la definizione di una POS è basata su caratteristiche:

- **Morfologiche:** gli affissi che compongono una parola
- **Sintattiche:** il contesto sintattico in cui si trova la parola

Non sono significative di solito *caratteristiche semantiche*, sebbene le classi presentino solitamente un buon grado coerenza semantica.

ESEMPIO (nomi)

I *nomi* in inglese hanno generalmente una forma singolare e una plurale (affisso –*s*)

I *nomi* in inglese sono solitamente preceduti da articoli o avere una forma possessiva (ES: *IBM's revenues*)

I *nomi* in inglese possono esprimere diverse categorie semantiche: persone, cose, astrazioni (ES*. relationship),* termini simil-verbali (ES: *pacing*)

Classificazione delle POS

Possono essere identificate due categorie principali:

- **CLASSI APERTE:**
  - Classi a cui vengono spesso aggiunte nuove parole
  - Generalmente produttive
  - ES: in Inglese ed Italiano sono quattro: *nomi, verbi, aggettivi, avverbi*
  - *Tutte le lingue hanno almeno le classi verbo e nome (lingua universale?)*

- **CLASSI CHIUSE:**
  - Classi cui appartengono un insieme relativamente statico di parole
  - ES: *articoli, preposizioni, congiunzioni, parole*
  - ***Function Words:*** parole grammaticalmente significative, generalmente molto corte e frequenti nel linguaggio
    - ES: *of, and, or, you …*

Classi Aperte

NOMI (N): *cose, persone, luoghi…*

- Nomi Comuni (NN) (*house, dog, …*)
- Nomi Propri (NNP) *(Gino, Pino,…)*
- Mass *vs* Count
  - *Count nouns*: hanno il plurale, sono enumerabili *(due panini)*
  - *Mass nouns*: gruppi omogenei, non enumerabili *(neve,sale…)*

VERBI (V): *azioni, processi …*

- Diverse classificazioni in base a: morfologia, sintassi.

AGGETTIVI (JJ): *proprietà, qualità…*

- Base, Superlativi, Comparativi

AVVERBI (RB): *modificatori di altre classi*

- Direzionali/Locativi *(here)*, Temporali *(now)*, Modali (*slowly*), di Gradazione *(very)…*

Classi Chiuse

PREPOSIZIONI : *relazioni temporali, spaziali…*

- Precedono i nomi *(of, in, for, …)*
- Molto comuni

PARTICLE:

- Si combinano con i verbi, formando *phrasal verbs*
- ES: *go on, take off, …*
- Si distinguono dalle preposizioni solo per caratteristiche sintattiche

ALTRE CLASSI:

- Articoli: *the, a, an*
- Congiunzioni: *and, or, but*
- Pronomi: personali *(I,you,us….)* e possessivi *(mine, yours,…)*
- Ausiliari: copulativi *(be, do, have)* e modali *(should, mus, can)*
- Interiezioni, numerali, negazioni, greetings….

Tagset per l'inglese

Possono essere definite moltissime classi, in base alle caratteristiche morfo-sintattiche.

- Tagset generici (solo macro-classi: nomi, verbi, aggettivi…)
- Tagset molto specifici (fino a 200-300 tag)
- Tagset a taglia media (C5 tagset, PeenTreebank)

PENN TREEBAK:

- E' uno dei tagset più utilizzati: applicato a numerosi corpora (*Brown Corpus*)
- Comprende 45 tag: troppo generico per alcune applicazioni

ESEMPIO (dal Brown Corpus)

```
The/DT grand/JJ jury/NN commmented/VBD on/IN a/DT
      number/NN of/IN other/JJ topics/NNS ./.
```

Penn Treebank tagset

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	*+,%, &*
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*'s*	”	Right quote	(’ or ”)
PRP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PRP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

POS Tagging: ambiguità

Processo di assegnazione della Part of Speech ad ogni parola di un corpus (insieme di documenti)

AMBIGUITA'

- Ogni parola dovrebbe avere un solo POS assegnato
- Molte parole sono **ambigue** (più POS tag possibili)
- Un *POS tagger* deve *disambiguare*, restituendo se possibile un solo tag:
  - Utilizzando evidenze contestuali
  - Utilizzando evidenze probabilisitiche da corpora annotati

ESEMPIO

–The ***back/JJ*** door

–On my ***back/NN***

–Win the voters ***back/RB***

–Promised to ***back/VB***

ESEMPIO

"La vecchia porta la sbarra "

…proviamo a costruire regole per disambiguare

POS Tagging: ambiguità

Quanto sono ambigue le parole inglesi ?

NON AMBIGUE (1 tag): **35,340** (***88,5%***)

AMBIGUE (2-7 tag): **4,100** (***11,5%***)

2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1

Brown Corpus (De Rose, 1988)

POS Tagging

Caratteristiche generali dei POS tagger:

INPUT

- Tagset
- Dizionario con tag
- Frase da annotare

OUTPUT

- Frase annotata

METODI

- Rule-based: database di regole di disambiguazione
- Stochastic: risolvono le ambiguità con evidenze probabilistiche estratte da un corpus annotato (*HMM, Markov models*…)
- Misti: utilizzano regole di disambiguazione ricavate con metodi stocastici

Sommario

Part of Speech Tagging

- **Part of Speech Tagging**
  - Cos'è
  - Part of Speech
  - **Part of Speech Tagging**
    - » Rule-bales
    - » Stochastic
    - » Misto
  - Prestazioni

POS Tagging: Rule Based

I sistemi rule-based prevedono due fasi

1. Assegnazione ad una parola di tutti i possibili POS tag, utilizzando un **dizionario**
2. Applicazione di **regole di disambiguazione** create <u>manualmente</u>, per rimuovere tag ambigui.

ESEMPIO (fase 1)

<u>DIZIONARIO</u>

- she: *PRP*
- promised: *VBN , VBD*
- to *TO*
- back: *VB, JJ, RB, NN*
- the: *DT*
- bill: *NN, VB*

	VBN		NN RB JJ		VB
PRP	VBD	TO	VB	DT	NN
She	***promised***	***to***	***back***	***the***	***bill***

POS Tagging: Rule Based

ESEMPIO (fase 2)

Regola 1: Rimuovere *VBN* se è il alternativa a VBD e se segue “<inzio frase>*PRP*”

Regola 2: Rimuovere *VB* se è in alternativa a NN e se segue *DT*

… …

	~~VBN~~		JJ		~~VB~~
			RB		
			NN		
PRP	VBD	TO	VB	DT	NN
She	***promised***	***to***	***back***	***the***	***bill***

POS Tagging: Rule Based

ENGTWOL *(Voutilainen, 1995)*

FASE 1:

- Lessico di 56,000 parole
- FST a due livelli per il parsing morfologico

FASE 2:

- 1,100 regole di disambiguazione in *espressione negativa*

ESEMPIO

Pavlov had shown that salivation …

Pavlov	PAVLOV N NOM SG PROPER
had	HAVE V PAST VFIN SVO
	HAVE PCP2 SVO
shown	SHOW PCP2 SVOO SVO SV
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	N NOM SG

Given input**: "that"**

If

(+1 A/ADV/QUANT)

(+2 SENT-LIM)

(NOT -1 SVOC/A)

Then **eliminate non-ADV tags**

Else **eliminate ADV**

POS Tagging: Stochastic

Differisce dall'approccio a regole nella fase di disambiguazione:

- Il tag corretto viene selezionato in base ad evidenze statistiche e alla teria della probabilità
- **Approcci semplici:** *Most Frequent Tag*
- **Approcci complessi:** *HMM, Transformation-based tagging*

METODO *MOST FREQUENT TAG*

- **IDEA:** Le parole ambigue utilizzano un tag più spesso di altri.
- **Metodo:**
  1. Creare un dizionario e annotare manualmente un corpus
  2. Per ogni parola ambigua in un nuovo testo calcolare probabilità di annotazione
  3. Assegnare il tag più probabile

POS Tagging: Stochastic

ESEMPIO

```
I/PP give/VB you/PP a/? pen/NN
Section/NN 381/CD a/?
```

- **Possibili tag per *a:*** **DT** **NN** **FW**
- **Qual è il più probabile ?**

1. **Utilizzo di un corpus**
   - Corpus: insieme di documenti annotato maualmente con tag non ambigui (ES: *Brown Corups,* 1 mil di parole)
   - Calcolare occorrenze di *a* con i diversi tag:

a/DT	21,830
a/NN	6
a/FW	3

POS Tagging: Stochastic

2. Calcolo della probabilità di annotazione con un determinato tag

- ES: Qual è la probabilità che la parola abbia un certo tag?

$$P(tag \mid word) = \frac{Count(word\ is\ tag)}{total\ Count(word)}$$

- La probabilità viene stimata con delle *conte statistiche* nel corpus (ad esempio nel Brown corpus, *P(Verb|race) = 96/98 = .98*)
- Nell'esempio:

$$P(DT \mid a) = \frac{Count(a\ is\ DT)}{total\ Count(a)} = \frac{21{,}830}{21{,}839} = 0{,}99996$$

$$P(NN \mid a) = \frac{Count(a\ is\ NN)}{total\ Count(a)} = \frac{6}{21{,}839} = 0{,}00002$$

POS Tagging: Stochastic

3. Assegnazione del tag più probabile

```
I/PP give/VB you/PP a/DT pen/NN
Section/NN 381/CD a/DT
```

ERRATO (*NN*)

LIMITI

- Annota bene in molti casi
- Nei casi più rari sbaglia sempre
- Per aumentare le prestazioni è necessario prendere in considerazione altre informazioni.
  - Ad esempio: guardare il tag della parola precedente e successiva risolverebbe il caso precedente
  - Implementare automaticamente in versione probabilistica le regole dei sistemi rule-based *(learning)*
- **Approcci complessi:** *HMM, Transformation-based tagging*