
8 Giugno 2006

Sintassi

Marco Pennacchiotti

pennacchiotti@info.uniroma2.it

Tel. 06 7259 7717

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

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

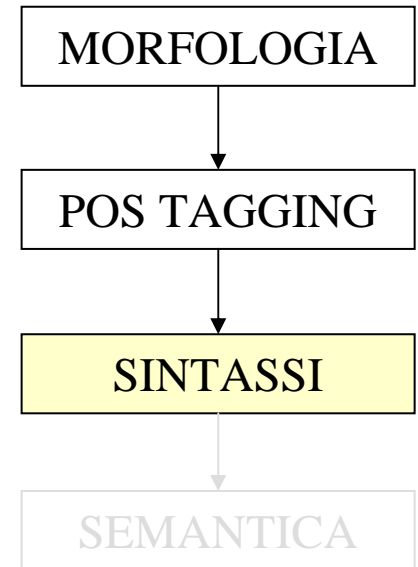
- **Semantica**

- Lexical Semantics
- Sentence Semantics

Sommario

Strumenti per la Sintassi

- **Introduzione**
- **Context-Free Grammar (CFG)**
 - Definizione
 - CFG per la sintassi
 - Limiti e problemi
- **Parsing**
 - Parsing a costituenti
 - Parsing Top-Down
 - Parsing Bottom-Up
 - Parsing misto (*left-corner*)
 - Chart parsing
 - Programmazione dinamica
 - Algoritmo di Earley
 - Valutazione
 - Parsing a dipendenze
 - Cenni
 - Conversione
 - Valutazione
 - Chaos



Sintassi: *definizioni*

La **sintassi** è lo studio delle strutture relazionali tra le parole

La **sintassi** definisce quindi il modo in cui le parole si possono “*combinare*” tra loro, per formare frasi **corrette**

ESEMPIO

“Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura”

→ **corretta**

“Oscura per mezzo nel selva del nostra mi ritrovai di cammin vita una”

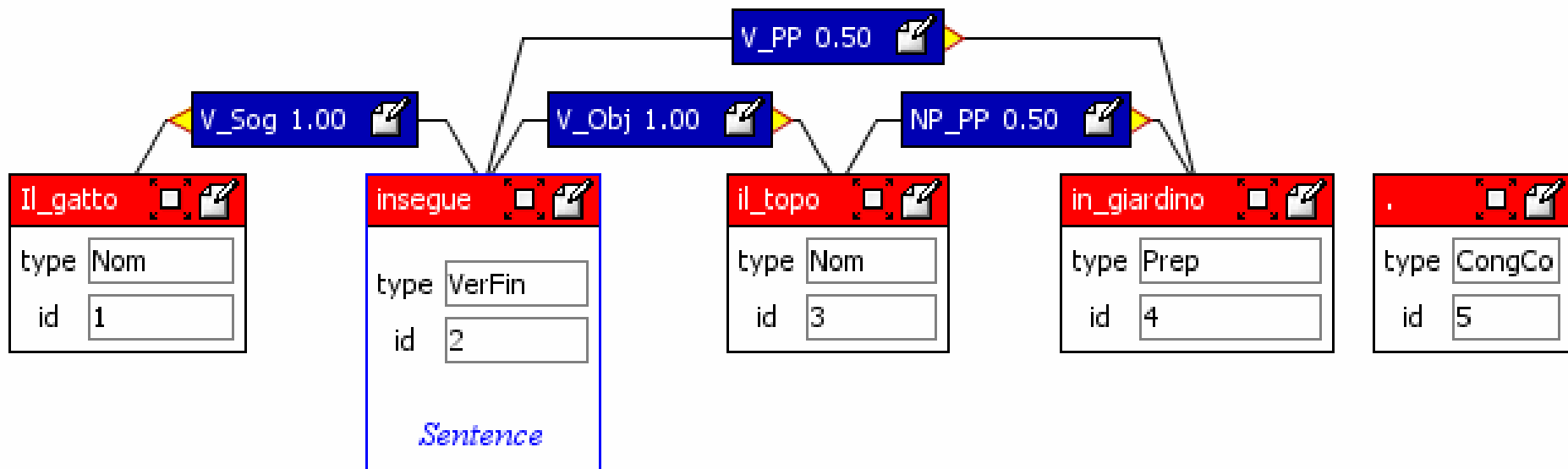
→ **sgrammaticata**

“Nella selva del mezzo cammino mi ritrovai per la nostra vita oscura”

→ **corretta ma senza semantica**

Sintassi: *esempio di analisi automatica*

Analisi a costituenti-dipendenze



Analisi Sintattica: *applicazioni*

- *Grammar checkers*

- Strumenti integrati in grado di controllare la correttezza grammaticale del testo inserito da un utente (*linguaggi naturali e linguaggi controllati*)

- *Question Answering*

- Interpretazione della question (*focus, question-words*) e rintracciamento della answer

- *Machine Translation*

- Regole di mappatura tra costrutti sintattici in lingue diverse

Analisi Sintattica: *definizioni*

- *Tre concetti fondamentali*

■ **Costituenti**

- Come le parole si raggruppano in unità (*sintagmi*) e come queste unità si comportano
- ES: [*il cane affamato*] [*ha rincorso*] [*il gatto*] [*nel giardino*]

■ **Relazioni Grammaticali**

- Quali flessioni le parole devono avere per relazionarsi
- ES: *il cane/SGMasc affamato/SGMasc*

■ **Sottocategorizzazione**

- Quali costruzioni sintattiche caratterizzano un verbo
- ES: [*il treno*] [*arriva*] [*a Roma*] (SOG vb PREP(a))

Constituency

Costituente: Gruppo di parole consecutive che si comportano come una singola unità sintattica. Generalmente costituenti dello stesso tipo occorrono in contesti sintattici simili (ad es. *Noun Phrases* precedono un verbo)

- **Noun Phrases** (Sintagmi Nominali) NP
[la mamma] compra [il gelato]
- **Prepositional Phrases** (Sintagmi Preposizionali) PP
la mamma [di Mario]
- **Verb Phrases** (Sintagmi Verbal) VP
la mamma [compra il gelato] al bambino
- **Adjectival Phrases** (Sintagmi Aggettivali) AP
lo strumento [utile a rivoltare la terra] è esposto al museo
- **Sentences** (Frase) S
[la mamma [che compra il gelato] ha un bambino]

Constituency Test

Come riconoscere un costituente ? (Alcune regole)

- **Movimento:** se è possibile effettuare un *movimento* del gruppo di parole nella frase, allora il gruppo è un costituente
 - **preposing:** un costituente si può spesso preporre, un non-costituente no
 - EN: *I can't stand [your new friend] → [Your new friend], I can't stand*
 - EN: *I can't stand [your new] friend → [Your new], I can't stand brother*
 - IT: *non sopporto [il tuo nuovo amico] → è [il tuo nuovo amico] che non sopporto*
 - **postposing:** un costituente si può spesso postporre, un non-costituente no
 - EN: *He tells [all the terrible problems] to her → He tells to her [all the terrible problems]*
 - IT: *ho preso [il caffè] al bar → ho preso al bar [il caffè]*
- **Isolabilità:** se è possibile porre una domanda su un gruppo di parole nella frase, allora il gruppo è un costituente
 - EN *You were reading [a nice book] → What were you reading? [A nice book]*
 - EN *You were reading [a nice] book → What book were you reading? [A nice]*
 - IT: *non sopporto [il tuo nuovo amico] → Chi non sopporti? [il tuo nuovo amico]*
- **Coordinazione:** Solo i costituenti si possono coordinare
 - EN *You were reading [a nice book] and [the beautiful Black's novel]*
 - IT: *non sopporto [il tuo nuovo amico] e [la signora dell'appartamento di fronte]*

Sintassi: *obiettivo*

Cosa deve essere in grado di fare un modello per la sintassi ?

- Riconoscere i costituenti
- Riconoscere una struttura corretta

“*{io}*” *Nel mezzo del cammin di nostra vita*” *mi ritrovai*” *per una selva oscura*”
NP **PP** **VP** **PP**

- Una frase italiana con la struttura (NP PP VP PP) è corretta

“*[Oscura per mezzo]* *[nel selva]* *[del nostra]* *[mi]* *[ritrovai]* *[di cammin vita una]*”
?? **PP** **??** **NP** **VP** **??**

- Una frase italiana con la struttura (?? PP ?? NP VP ??) è scorretta

Sommario

- **Strumenti per la Sintassi**
- **Introduzione**
- **Context-Free Grammar (CFG)**
 - **Definizione**
 - CFG per la sintassi
 - Limiti e problemi
- Parsing
 - Parsing a costituenti
 - Parsing Top-Down
 - Parsing Bottom-Up
 - Parsing misto (*left-corner*)
 - Chart parsing
 - Programmazione dinamica
 - Algoritmo di Earley
 - Valutazione
 - Parsing a dipendenze
 - Cenni
 - Conversione
 - Valutazione
 - Chaos

Context Free Grammar (CFG)

CFG: è' il **modello dichiarativo** più utilizzato per rappresentare i costituenti e le strutture sintattiche che essi formano

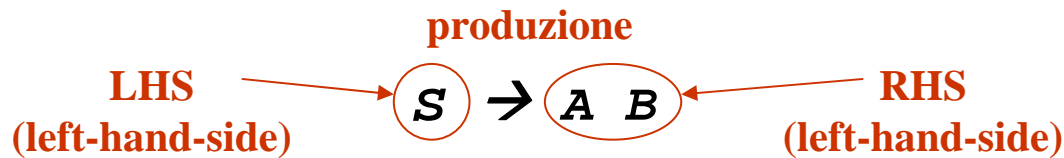
- dette anche Phrase-Structure Grammars (PSG); formalizzata da Chomsky (1956) e da Backus (1959)
- Composta da:
 - **REGOLE (o produzioni):** esprimono il modo in cui i simboli del linguaggio possono essere combinati e ordinati
 - **LESSICO:** comprende i simboli del linguaggio (parole e simboli)
- Hanno una diretta implementazione in Prolog attraverso le *DCG (Definite Clause Grammar)*
- Permettono di condurre analisi sintattiche complete, utilizzando appropriati algoritmi di parsing sintattico

ESEMPIO: $S \rightarrow NP VP$

CFG: *definizione formale*

-Una **grammatica CFG** 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 del tipo $A \rightarrow \alpha$, dove A è un non-terminale e α è una stringa di simboli $\in (\mathbf{N} \cup \Sigma)^*$



ESEMPIO (CFG)

$S \rightarrow ABa$

$A \rightarrow a$

$B \rightarrow b$

ESEMPIO (non-CFG)

$AB \rightarrow aBBa$

$B \rightarrow a$

CFG e complessità

Le CFG sono di **Tipo 2** nella Gerarchia di Chomsky, e sembrano modellare bene la sintassi sia Italiana che Inglese

- Grammatiche Context-Sensitive (*Tipo 1*)
 - Troppo complesse
 - Non esistono nella sintassi costrutti $a^n b^n c^n$
- Grammatiche Regolari (*Tipo 3*)
 - Troppo semplici
 - Non possono rappresentare costrutti sintattici $a^n b^n$ (center-embedded)

→ La sintassi IT ed EN può essere modellata con strumenti relativamente semplici, sebbene sia più complessa della morfologia

CFG e complessità

Perché “Context-Free” ?

- Vuol dire “*Grammatiche libere dal contesto*”
- Il **contesto** è l'insieme dei simboli vicini a un non-terminale in LHS
- Nelle CFG il non-terminale in LHS è libero da contesto, è cioè da solo.

ESEMPIO (CFG)

$$A \rightarrow BCa$$

A può essere riscritto come B seguito da C e da a, a prescindere dal contesto in cui si trova

ESEMPIO (non-CFG)

$$AC \rightarrow Ba$$

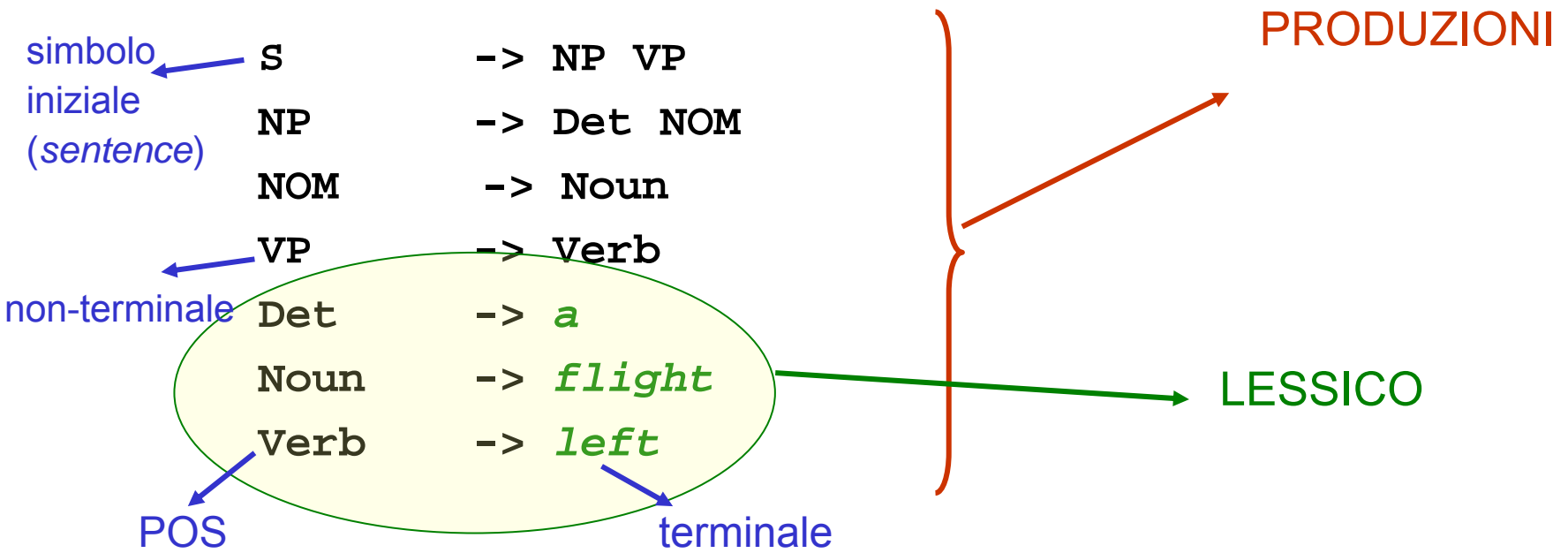
A può essere riscritto come B seguito da a, se associato ad una C

Sommario

- **Strumenti per la Sintassi**
- **Introduzione**
- **Context-Free Grammar (CFG)**
 - Definizione
 - **CFG per la sintassi**
 - Limiti e problemi
- Parsing
 - Parsing a costituenti
 - Parsing Top-Down
 - Parsing Bottom-Up
 - Parsing misto (*left-corner*)
 - Chart parsing
 - Programmazione dinamica
 - Algoritmo di Earley
 - Valutazione
 - Parsing a dipendenze
 - Cenni
 - Conversione
 - Valutazione
 - Chaos

CFG per la sintassi

- *Regole sintattiche* : modellate dalle produzioni
- *Elementi del Lessico* : alfabeto Σ
- *Costituenti* : alfabeto N



$\Sigma = \{a, flight, left\}$

$N = \{S, NP, VP, NOM, Det, Noun, Verb\}$

$L = \{a flight left\}$

CFG per la sintassi

S -> NP VP
Noun -> *flight*

Come si legge ?

- “L’ unità S consiste in un *NP* seguito da un *VP*”.

Ovvero:

- “Riscrivi S come *NP* seguito da *VP*”

- **PAROLE** simboli non terminali Σ (*a, flight, left*)
- **COSTITUENTI** simboli non terminali N (*S, NP, VP, Noun, ...*)
- **POS** LHS del lessico (*Noun, Verb, Det, ...*)

CFG per la sintassi

- Come per gli FSA/FST, una CFG può essere utilizzata per :
 - generare tutte le stringhe valide del linguaggio (**generatore**)
 - verificare se una frase appartiene al linguaggio ed assegnarle una struttura (**parser**)

DERIVAZIONE

- Applicazione delle regole di produzione ad una stringa
- ES: “a *flight*” può essere derivata da *NP*

NP	->	Det	NOM
NOM	->	Noun	
Det	->	<i>a</i>	
Noun	->	<i>flight</i>	

CFG per la sintassi

- Il *linguaggio* è definito tramite tutte le possibili derivazioni della CFG

DERIVAZIONE: definizione formale

Data una produzione $A \rightarrow \beta$ e due stringhe $\alpha, \gamma \in (N \cup \Sigma)^*$ allora:

$\alpha A \gamma \Rightarrow \alpha \beta \gamma$ **derivazione diretta**

Date le stringhe $\alpha_1, \alpha_2, \dots, \alpha_m \in (N \cup \Sigma)^*$ con $m > 1$ tali che

$\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3 \dots \alpha_{m-1} \Rightarrow \alpha_m$

Allora: $\alpha_1 \Rightarrow^* \alpha_m$ **derivazione**

ESEMPIO

NP -> Det NOM

NOM -> Noun

Det -> *a*

Noun -> *flight*

Det NOM \Rightarrow Det Noun

NOM \Rightarrow^* *flight*

CFG per la sintassi

-Generalmente le derivazioni vengono rappresentate con un **parse-tree** (*albero sintattico*)

-Un albero può rappresentare più derivazioni

ESEMPIO

“a flight left”

Grammatica

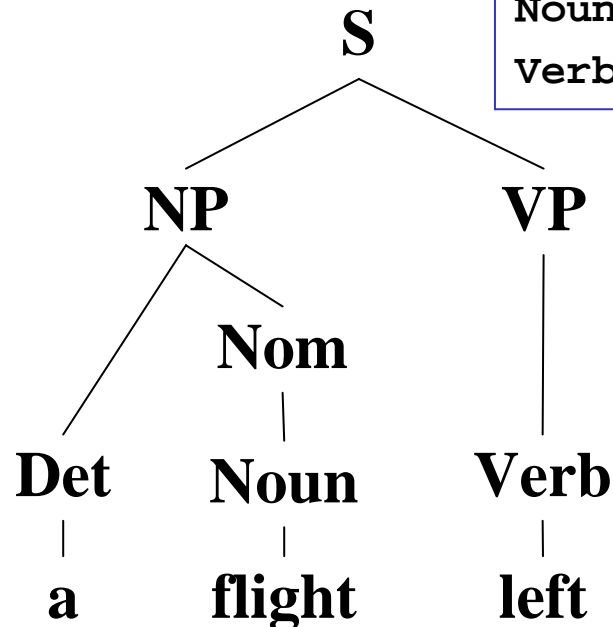
S	-> NP VP
NP	-> Det NOM
NOM	-> Noun
VP	-> Verb
Det	-> <i>a</i>
Noun	-> <i>flight</i>
Verb	-> <i>left</i>

DERIVAZIONE 1

S
NP VP
Det Nom VP
Det Nom Verb
a Nom Verb
a Noun Verb
a flight Verb
a flight left

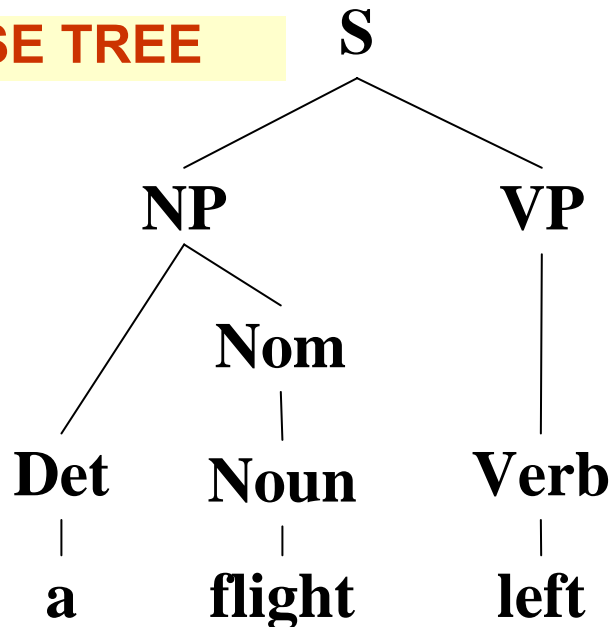
DERIVAZIONE 2

S
NP VP
NP Verb
NP *left*
Det Nom *left*
Det Noun *left*
Det *flight* *left*
a flight left



CFG per la sintassi

PARSE TREE



S	-> NP VP
NP	-> Det NOM
NOM	-> Noun
VP	-> Verb
Det	-> <i>a</i>
Noun	-> <i>flight</i>
Verb	-> <i>left</i>

- Ogni parte dell' albero è giustificato da una regola della grammatica
- Se per una frase è possibile costruire almeno un albero, essa è **corretta**

BRACKET NOTATION

[_S[_{NP}[_{Det} a] [_{Nom} [_N flight]]] [_{VP}[_V left]]]

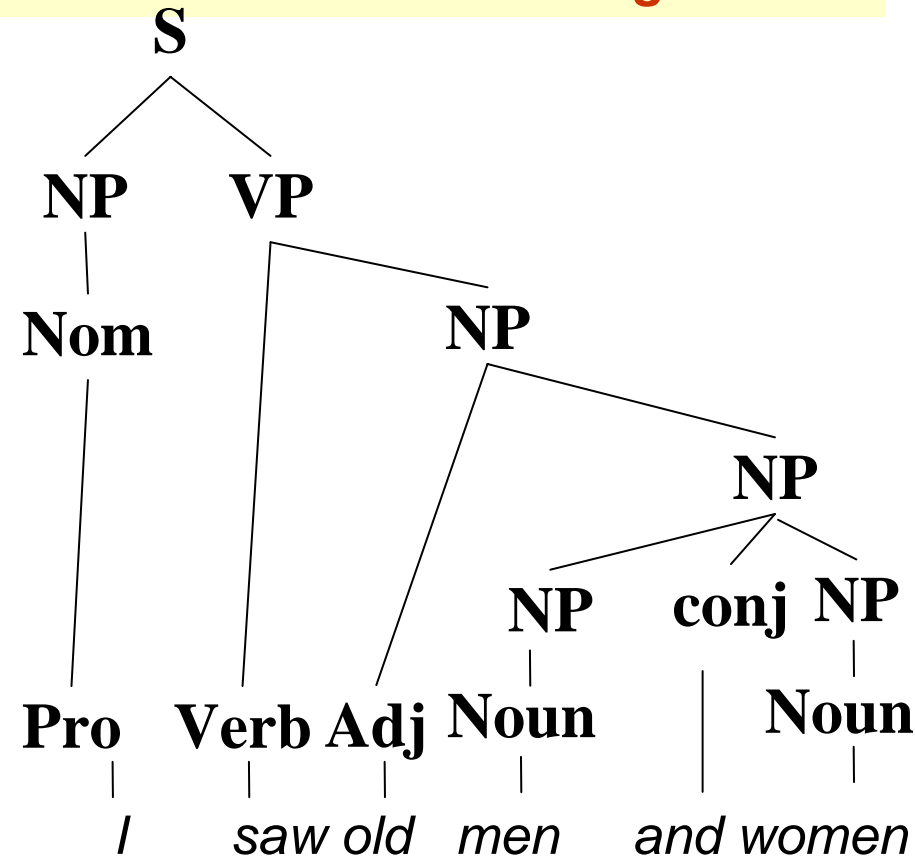
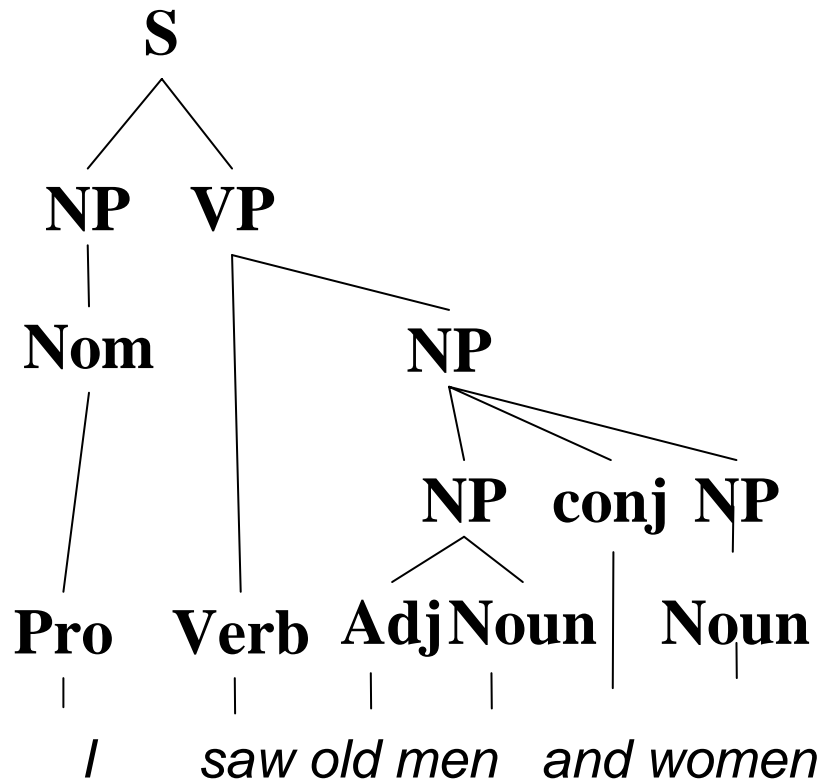
PARSING SINTATTICO

Processo che , date in input una stringa e una grammatica CFG, restituisce uno o più parse-tree per la stringa

CFG: ambiguità

AMBIGUITA'

Se è possibile costruire due alberi diversi a partire dalla stessa frase e dalla stessa grammatica, la frase è **sintatticamente ambigua**



CFG: ambiguità

AMBIGUITA'

Una frase che abbia più interpretazioni possibili (più di un parse-tree) è detta **sintatticamente ambigua** (o *strutturalmente ambigua*)

“Mario guarda Laura col cannocchiale”

Tre tipi principali di ambiguità strutturale:

- **attachment ambiguity**
 - *“We saw the Eiffel Tower flying to Paris”*
- **coordination ambiguity**
 - *“I saw old men and women”*
- **noun-phrase bracketing ambiguity**
 - *“Can you book TWA flights ?”*

Tipi di Frasi (S)

In Inglese, esistono quattro tipi fondamentali di strutture frasali:

- **Dichiarativa**

A plane left

- **S → NP VP**
- E' la struttura più usata

- **Imperativa**

Leave!

- **S → VP**
- Non c'è soggetto. Utilizzata per comandi e suggerimenti

Tipi di Frasi (S)

- **Interrogativa (Si/No)**

Does the plane leave ?

- **S → Aux NP VP**
- Domande affermative/negative

- **Interrogativa (Wh)**

- Domande introdotte da *who, which, what, ...*

- **Wh soggetto**

Which plane leaves ?

- **S → Wh-NP VP**
- come la dichiarativa, ma introdotto da *wh*

- **Wh non-soggetto**

- **S → Wh-NP Aux NP VP**

What flight do you prefer ?

Sintagmi Verbali (VP)

Un sintagma verbale è costituito da un verbo seguito da altri costituenti (ad esempio un *NP* con funzione di complemento oggetto)

Tipologie fondamentali:

- **VP → Verb NP**
- **VP → Verb PP**
- **VP → Verb NP PP**

prefer a morning flight

leaving on Monday

leave Boston in the morning

Tipologie complesse:

- **VP → Verb S**
- **VP → Verb VP**

You say John will leave tomorrow

I want to fly from Rome to London

NOTA:

- I phrasal verbs (ES: *take off*) sono trattati come verbi semplici (*Verb*)
- Un verbo può consentire solo alcune delle tipologie di VP (sottocategorizzazione)

Sintagmi Nominali (NP)

- Un sintagma nominale è costruito intorno ad un nome (*Noun*) centrale che ne guida il significato, detto **head**.
- **Modificatori** della head possono essere inseriti prima (*pre-head modifiers*) o dopo di essa (*post-head modifiers*)

Pre-head modifiers

- **determiners** NP → det Noun *a flight, the flight*
- **pre-determiners** *all the flight*
- **post-determiners** (*quantificatori, numerali, aggettivi*) *the long flight*

Post-head modifiers

- **Prepositional Phrase** NP → Nom PP *a flight from Rome*
- **Non-finite Clause** NP → Nom GerundVP *a flight leaving today*
- **Relative Clases** NP → Nom RelClause *a fight **that** goes to Rome*

Altri sintagmi

Possono essere introdotti nella grammatica diversi altri sintagmi. I più importanti sono:

- **Sintagma Preposizionale (PP)** *from Los Angeles*
 - formato da una preposizione e un sintagma nominale
 - generalmente segue un sintagma nominale o uno verbale
 - **PP → Prep NP**

- **Sintagma Aggetivale (AP)** *least expansive*
 - formato da un aggettivo, eventualmente con avverbio preposto
 - **AP → Adj**
 - **AP → Adv Adj**

Congiunzione

Sintagmi o intere frasi possono essere congiunte da *and*, *or*, *but*, ...

- **NP → NP Conj NP**

- *I'll take* [_{NP}[_{NP}*the bags*] **and** [_{NP}*the tickets*]]

- **VP → VP Conj VP**

- *What flight* [_{VP}[_{VP}*goes to London*] **and** [_{VP}*passes through Paris*]] ?

- **S → S Conj S**

- [_S[_S*I like London*] **but** [_S*I hate Paris*]] ?

In generale:

$X \rightarrow X \text{ Cong } X$

Esempio di Grammatica

Si possono costruire CFG di diversa complessità, capaci di catturare un numero maggiore o minore di fenomeni sintattici (equivalentemente a quanto accade nella sintassi con i FST)

LESSICO

POS

Noun
Verb
Adj
Pron
PropNoun
Det
Prep
Cong

-> flight | breeze | trip | ...
-> is | prefer | leave | ...
-> first | cheap | new | ...
-> me | I | you | ...
-> John | Rome | Los Angeles | ...
-> the | a | this | ...
-> from | to | with | ...
-> and | or | but | ...

GRAMMATICA

S -> NP VP
NP -> Pron | PropNoun | Det Nom
Nom -> Noun Nom | Noun
VP -> Verb | Verb NP | Verb PP | Verb NP PP
PP -> Prep NP

Esempio di Grammatica

GRAMMATICA

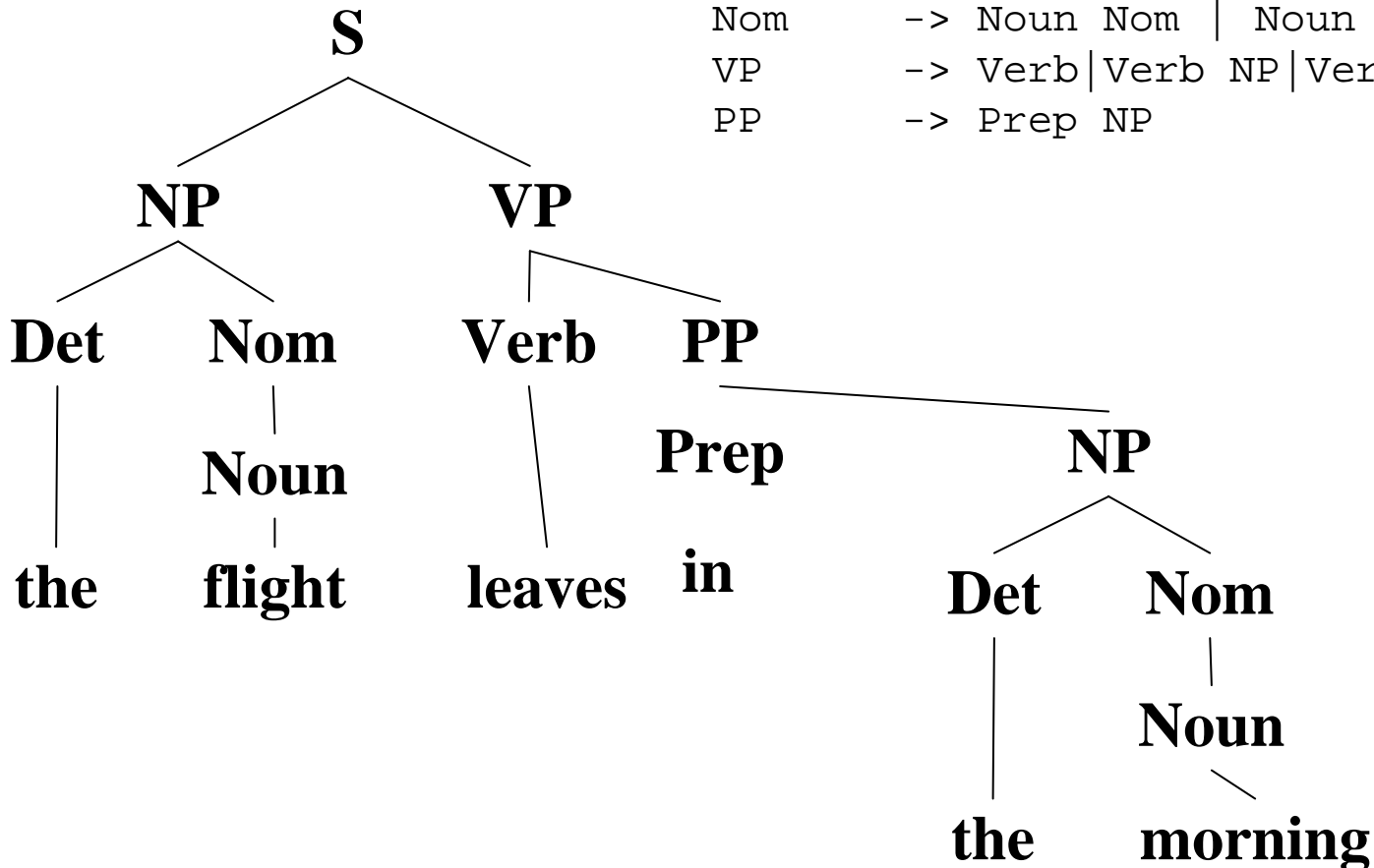
S -> NP VP

NP -> Pron | PropNoun | Det Nom

Nom -> Noun Nom | Noun

VP -> Verb | Verb NP | Verb PP | Verb NP PP

PP -> Prep NP



Ricorsione

- Una CFG è ricorsiva se esiste almeno una categoria non-terminale che possiede una derivazione in cui è incluso se stesso

LEFT- RECURSION

Una grammatica è ricorsiva a sinistra se contiene un non-terminale A tale che :

$$A \Rightarrow^* \alpha A \gamma \quad \text{e} \quad \alpha \Rightarrow^* \varepsilon$$

RIGHT- RECURSION

Una grammatica è ricorsiva a destra se contiene un non-terminale A tale che :

$$A \Rightarrow^* \alpha A \gamma \quad \text{e} \quad \gamma \Rightarrow^* \varepsilon$$

ESEMPI

NP \rightarrow NP PP NP \rightarrow Det Nom
Det \rightarrow NP 's

Ricorsione

Quindi, si ha ricorsione quando il non-terminale LHS compare anche in RHS

- Utile per rappresentare **sintagmi nominali** e **sintagmi verbali** complessi
 - **NP** → **NP PP**
 - **VP** → **VP PP**

ESEMPIO

$$\underbrace{[_{NP}[_{NP}[_{Nom}[_N \text{flights}]]]}_{\text{NP}} \underbrace{[_{PP}[_{Prep} \text{from}] [_{Nom}[_N \text{denver}]]]}_{\text{PP}}]$$

$$\underbrace{\hspace{15em}}_{\text{NP}}$$

[[[Flights] [from Denver]] [to Miami]]

[[[[Flights] [from Denver]] [to Miami]] [in February]]

[[[[[Flights] [from Denver]] [to Miami]] [in February]] [on a Friday]]

1. Scrivere una grammatica CFG, sul modello di quella precedentemente introdotta, in grado di riconoscere le seguenti frasi:
 - “Does America 487 have a first class section ? ”*
 - “ Does America Airlines have a flight between 5 and 6 ?”*
 - “ Is there an America airlines flight from Rome to London ?”*
2. Produrre i parse-tree relativi alle frasi proposte

Sommario

- **Strumenti per la Sintassi**
- **Introduzione**
- **Context-Free Grammar (CFG)**
 - Definizione
 - CFG per la sintassi
 - **Limiti e problemi**
- Parsing
 - Parsing a costituenti
 - Parsing Top-Down
 - Parsing Bottom-Up
 - Parsing misto (*left-corner*)
 - Chart parsing
 - Programmazione dinamica
 - Algoritmo di Earley
 - Valutazione
 - Parsing a dipendenze
 - Cenni
 - Conversione
 - Valutazione
 - Chaos

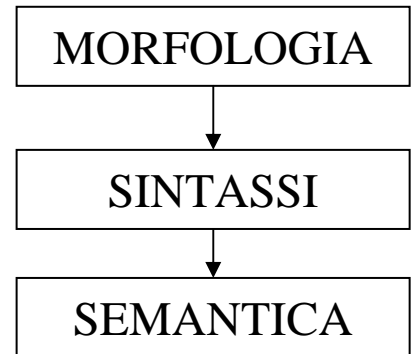
Problemi: *agreement*

...e la morfologia ?

- L' analisi sintattica non può prescindere da quella morfologica !
- Ad esempio:

✓ The flight **leaves** in the morning **OK!**

✗ The flight **leave** in the morning **NO!**



- E' necessario rappresentare nella grammatica l'**agreement** tra le proprietà morfologiche dei costituenti
- Le proprietà morfologiche sono fornite da un parser morfologico precedente (ad es.FST)

Problemi: *agreement*

■ SOLUZIONE 1: Espansione della grammatica

- Aggiungere regole esplicite per l'agreement:

S → NP VP

NP → Det pINom

pINom → pINom pINoun | pINoun

S → 3sgNP 3sgVP

3sgNP → Det sgNom

sgNom → sgNom sgNoun | sgNoun

sgNoun → flight | plane ...

pINoun → flights | planes ...

sgV → leaves | ...

pIV → leave | ...

- Per ogni fenomeno morfologico la grammatica raddoppia !!
- Quanti fenomeni è necessario catturare?
 - 3sing – 3plur
 - Aux nelle interrogative
 - Pronomi
 -

*The flight **stop** ?*

***Do** the flight stops ?*

***Her** likes I*

Problemi: *agreement*

■ SOLUZIONE 2: Parametrizzazione

- Aggiungere dei parametri per gestire l'agreement:

$S \rightarrow NP(x) VP(x)$

$NP(x) \rightarrow Det Nom(x)$

$Nom(x) \rightarrow Nom(x) Noun(x) \mid Noun(x)$

$Noun(sg) \rightarrow flight \mid plane \dots$

$Noun(pl) \rightarrow flights \mid planes \dots$

$V(sg) \rightarrow leaves \mid \dots$

$V(pl) \rightarrow leave \mid \dots$

- Le **Feature Structures** sono il formalismo più utilizzato per esprimere l'agreement attraverso parametrizzazioni

Problemi: sottocategorizzazione verbale

- Non tutte le regole per *VP* sono applicabili a tutti i verbi
- Ogni verbo ha propri *patterns di sottocategorizzazione*: strutture sintattiche caratteristiche
- I *patterns di sottocategorizzazione* esprimono quindi i vincoli che un verbo ha sul numero e il tipo sintattico dei propri complementi
 - ESEMPIO:

<i>eat</i>	Non ha il complemento oggetto
<i>give</i>	Non ha il complemento <i>with</i>
- Le regole CFG quindi *overgenerano*

Problemi: sottocategorizzazione verbale

- Categorie base:
 - VERBI INTRANSITIVI
 - VP → Verb | Verb PP
 - VERBI TRANSITIVI
 - VP → Verb NP | Verb NP PP | Verb PP | Verb
- Molte grammatiche distinguono anche centinaia di sottocategorizzazioni, dette **frame di sottocategorizzazione**

- ES:

0	eat, sleep	<i>I sleep</i>
NP	find, leave	<i>I find [NP the flight to Rome]</i>
NP PP _{with}	help, load	<i>Help [NP me] [PP with the flight]</i>

Problemi: sottocategorizzazione verbale

- **POSSIBILE SOLUZIONE :**

- Aggiungere regole ad- hoc:

VP → IntransV

VP → TransV NP

VP → TransV NP PP

... ..

- Si dovrebbero aggiungere regole specifiche per tutte le centinaia di patterns di sottocategorizzazione!
- Quindi stesso problema di aumento della grammatica che si ha con l'agreement

Problemi: *movimento*

[_S[_{NP}My travel agent] [_{VP}booked [_{NP}the flight]]

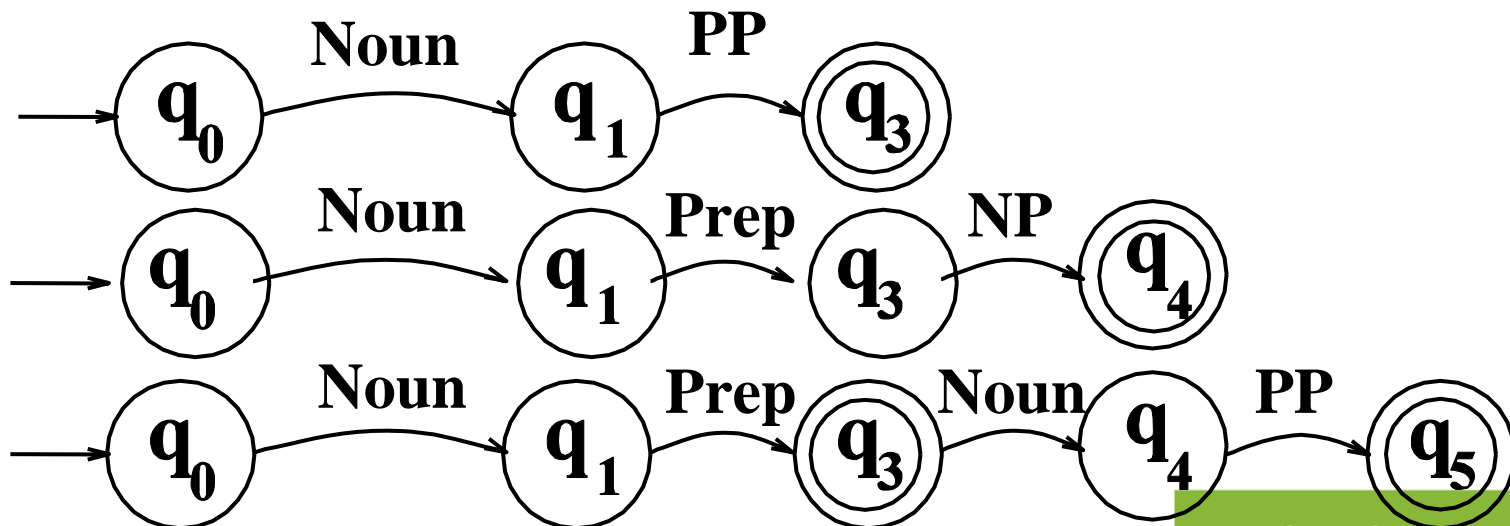
- *Book* è un normale verbo transitivo
 - VP → Verb NP
 - Attende quindi sempre il **complemento oggetto** alla sua destra

Which flight do you want me to have the travel agent to **book** ?

- Il complemento oggetto è preposto, molto lontano e separato da due verbi !
- Semplici CFG non sono in grado di modellare questi fenomeni (*long distance dependencies*)

CFG e FSA

- La presenza di **left and right recursion** non permette di utilizzare automi (Classe 3) per modellare la sintassi:
 - Automi a **stati finiti** non permettono di rappresentare ricorsione!
 - e.g.: NP \rightarrow Noun PP
 - \rightarrow Noun Prep NP
 - \rightarrow Noun Prep Noun PP
 - \rightarrow Noun Prep Noun Prep NP.....



CFG e FSA

■ Soluzione:

1. Espandere l'FSA fino ad una profondità plausibile del linguaggio
2. Utilizzare Recursive Transition Networks (RTN)

CFG: problemi

Data la seguente grammatica:

S -> NP VP
NP -> Pron | PropNoun | Det Nom
Nom -> Noun Nom | Noun
VP -> Verb | Verb NP | Verb PP | Verb NP PP
PP -> Prep NP

1. Aumentare la grammatica in modo da gestire i pronomi personali *“he, she, her, him”* nel caso corretto. Es:
 - *“He shot her”* (corretto)
 - *“Her shot he”* (errato)
2. Aumentare la grammatica in modo da gestire i seguenti patterns di sottocategorizzazione:
 - Verbi senza NP: disappear (*“I disappear”*)
 - Verbi con NP : find, leave, repeat (*“I find the flight”*)
 - Verbi con due NP: give, bring (*“give me the book”*)