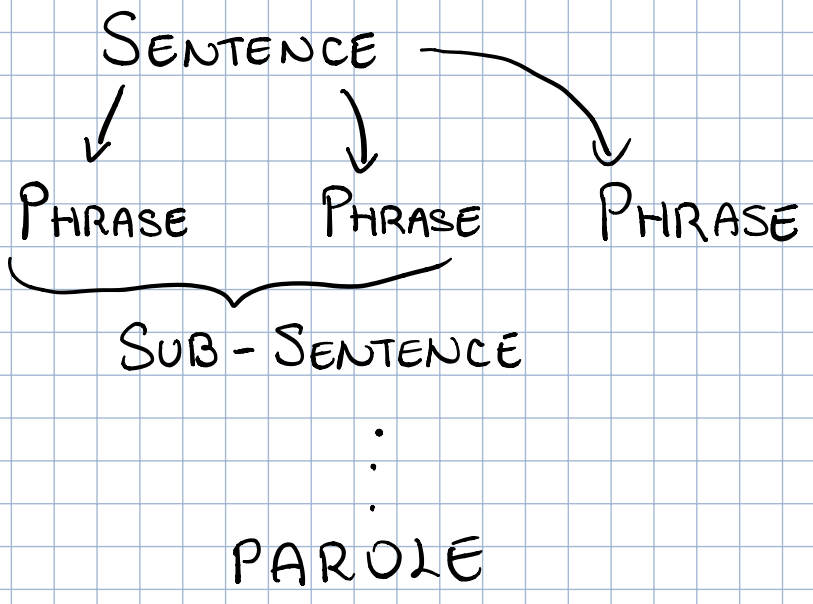
NLP - LEZIONE 09

DEL 31/10/2019

STRUTTURA GRAMMATICA

L'oggetto principale di una grammatica sono le SENTENCES, le quali che vengono composte sono le PAROLE, che vengono organizzate in PHRASES che a loro volte possono formare SUB-SENTENCES. Otteniamo quindi una struttura altamente RICORSIVA.



NOUN PHRASE

Governata dal nome. In italiano abbiamo le seguenti regole

NP → ART NOME

NP → ART ADJ NOME

NP → ART NOME ADJP

Ad esempio abbiamo la seguente cosa

PREPOSITIONAL PHRASE

Una PP è una NP a cui ci mettiamo davanti una preposizione.

PP → PREP NP

Una PP può anche essere dentro una NP o una ADJP

ADJP → ADJ PP

VERBAL PHRASE

VP → VX NP
VP → VX NP PP
VP → VX NP PP PP
VP → VX PP

VX → V
VX → AUX V

SENTENCE

Una tipica frase è data da

$$S \rightarrow NP\ VP$$

Notiamo a questo punto notiamo che la grammatica ottenuta non è in CNF, e quindi non possiamo utilizzare il CYK per PARSARE le varie sentences. Necessitiamo di un altro algoritmo di parsing che sia in grado di gestire grammatiche CF generali.

ANNOTAZIONI ~ (15:00/1 min)

Una ANNOTAZIONE è un ARCO tra due punti: un punto di INIZIO e un punto di FINE.

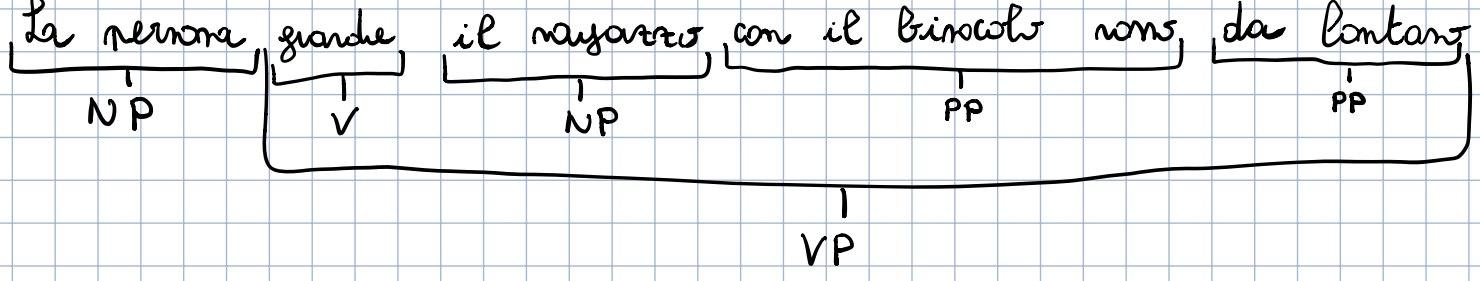
Annotare una frase ci permette di costruire un GRAFO i cui nodi sono gli SPAZI bianchi tra le parole e NON le parole stesse.

Ad una annotazione è associata una ETICHETTA, che può essere strutturata e prende il nome di FEATURE STRUCTURE.

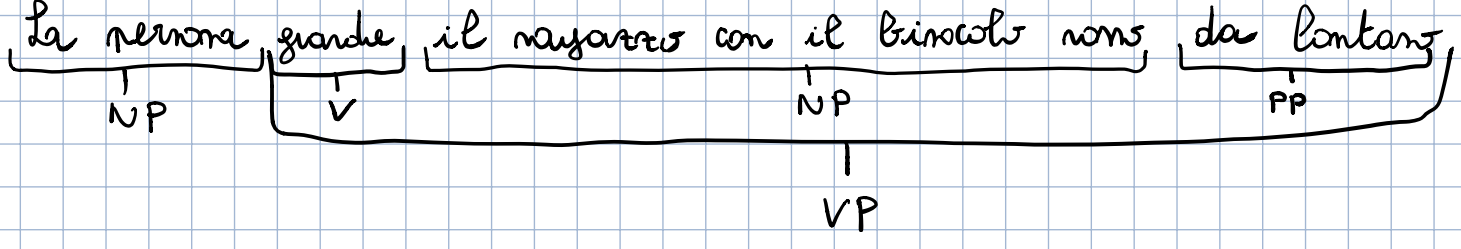
Consideriamo la frase

"La persona guarda il ragazzo con il binocolo non da lontano"

Possiamo annotare la frase in almeno i seguenti due modi:



In questa interpretazione è la persona che utilizza il binocolo non per guardare il ragazzo.



In quest'altra interpretazione è il ragazzo ad avere il binocolo non.

Un eventuale algoritmo di PARSING deve tenere in considerazione la potenziale AMBIGUITÁ della grammatica.

EARLEY ALGORITHM

The early algorithm parses the input string by constructing a CHART TABLE that is heavily used in a DYNAMIC PROGRAMMING way.

The earley algorithm uses a CURSOR, denoted by a dot (•), which is used to determine how much of a given rule in the grammar has been MATCHED in that particular step.

In particular, given a NT X and two strings of terminals/non-terminals, the notation

$$X \rightarrow \alpha \cdot \beta$$

is used to represent the fact that α has already been parsed and that β has yet to be parsed.

A combination of the following MAIN OPERATIONS are used by the EARLEY PARSER:

- PREDICT

  Used to determine all NON-TERMINALS that will be expanded in the next operation.

- COMPLETE

  Used to complete a scan operation, it moves the DOT (•) in all the production that have been MATCHED.

- SCAN

  Used to check the input string in order to match grammar's rules.

The main aspects of the EARLEY PARSERS are thus the following:

- DYNAMIC PROGRAMMING (CHART MATRIX)
- ANNOTATIONS
- NOTATION ($X \rightarrow \alpha \cdot B$)

We will now see an example of the EARLEY parser. The example has associated AUDIO that explains how the chart was filled.

EARLEY PARSER EXAMPLE

(TAKEN FROM YOUTUBE VIDEO "CYK ALGORITHM BY DEEBA KANNAN")

GRAMMAR G

S → NP VP	P → WITH
PP → P NP	V → DRINK
VP → V NP	N → PAPU
VP → VP PP	N → MILK
NP → NP PP	N → CHOCOLATE
NP → N	N → COFFEE

♫ : SPIEGAZIONE AUDIO ESEMPIO !

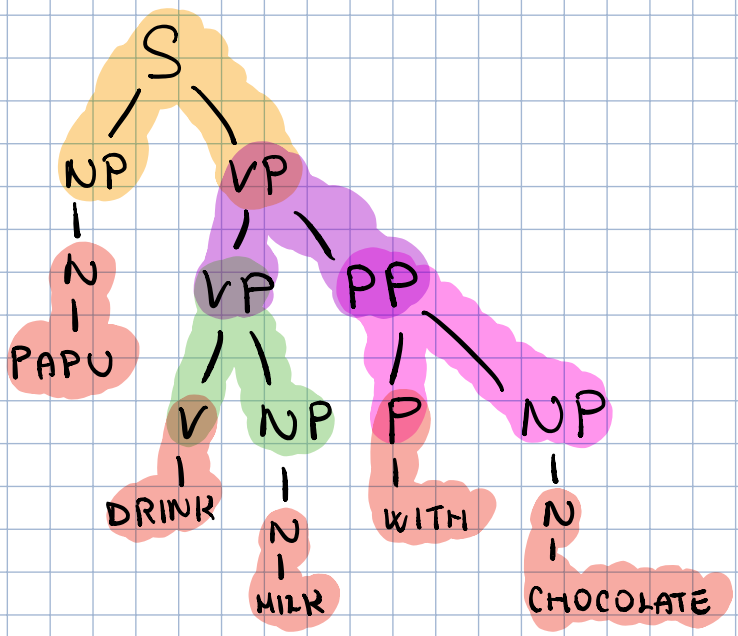
INPUT STRING

PAPU DRINK MILK WITH CHOCOLATE

PARSER CHART

				(4,4) NP → • NP PP NP → • N N → • CHOCOLATE	(4,5) N → CHOCOLATE • NP → N • NP → NP • PP
			(3,3) PP → • P NP P → • WITH	(3,4) P → WITH • PP → P • NP	(3,5) PP → P NP •
		(2,2) NP → • NP PP NP → • N N → • MILK	(2,3) N → MILK • NP → N • NP → NP • PP		(2,5) NP → NP PP •
	(1,1) VP → • V NP VP → • VP PP ~~PP → • P NP~~ V → • DRINK	(1,2) V → DRINK • VP → V • NP	(1,3) VP → V NP • P → VP • PP		(1,5) VP → VP PP •
(0,0) TOP → • S S → • NP VP NP → • NP PP NP → • N N → • PAPU	(0,1) N → PAPU • NP → N • NP → NP • PP S → NP • VP		(0,3) S → NP VP •		(0,5) S → NP VP •

The obtained PARSE TREES are as follows



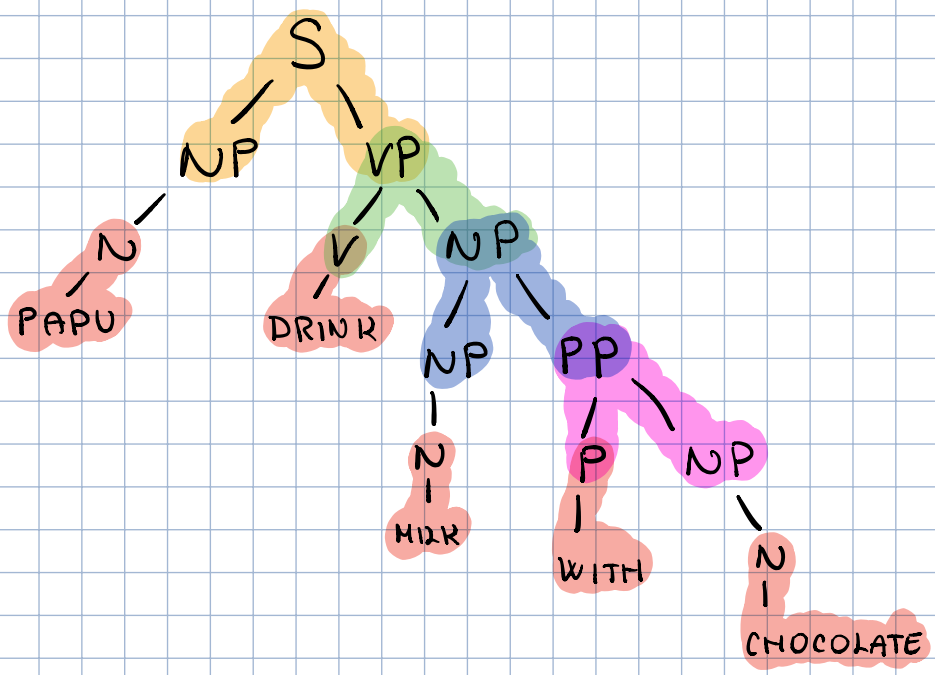


CHART PARSER ~ (03:00/2 min)

Sia il CYK che l'EARLEY PARSER si basano sui seguenti concetti fondamentali

- ANNOTAZIONE
- ETICHETTA CON REGOLA
- REGOLA ATTIVA

Per motivi di efficienza è meglio utilizzare un algoritmo BOTTOM-UP. In particolare esiste il BOTTOM-UP CHART PARSING.

A differenza del CYK, nell'EARLEY PARSER durante l'operazione di PREDICT, nella chart table vengono inserite un mucchio di REGOLE. In particolare non ci sono modi per gestire questo, e quindi il fattore $|G|$ nella SPACE COMPLEXITY si fa molto sentire, e rende l'EARLEY PARSER quasi inutilizzabile per grammatiche molto grandi.

Trattare le Congiunzioni Coordinative

Consideriamo la frase

MARIO MANGIA PANE E MARIA MANGIA SALAME

notiamo che la frase può anche essere scritta come segue

MARIO MANGIA PANE E MARIA SALAME

Come possiamo rappresentare le PARTI SOTTO INTESE del discorso?

Volendo utilizzare una rappresentazione a COSTITUENT, possiamo iniziare utilizzando la seguente regola

$$S \rightarrow S \quad CC \quad S$$

Questa regola però ci permette di gestire frasi semplici senza parti sottointese. In generale trattare le parti mancanti è un problema molto difficile.

MARIO MANGIA PANE E BEVE IL VINO

In questa frase manca il SOGGETTO

Se non trattiamo questo problema al livello della SINTASSI nel successivo livello, quello della SEMANTICA, incapperemo in varie problematiche nel dare la giusta interpretazione alle strutture sintattiche costruite.

Per trattare queste problematiche possiamo addestrare un modulo in grado di inserire dei PLACE-HOLDERS all'interno delle frasi da analizzare che vanno ad ESPLICITARE le parti sottointese della frase.

DEPENDENCIES ~ (35:00/2 min)

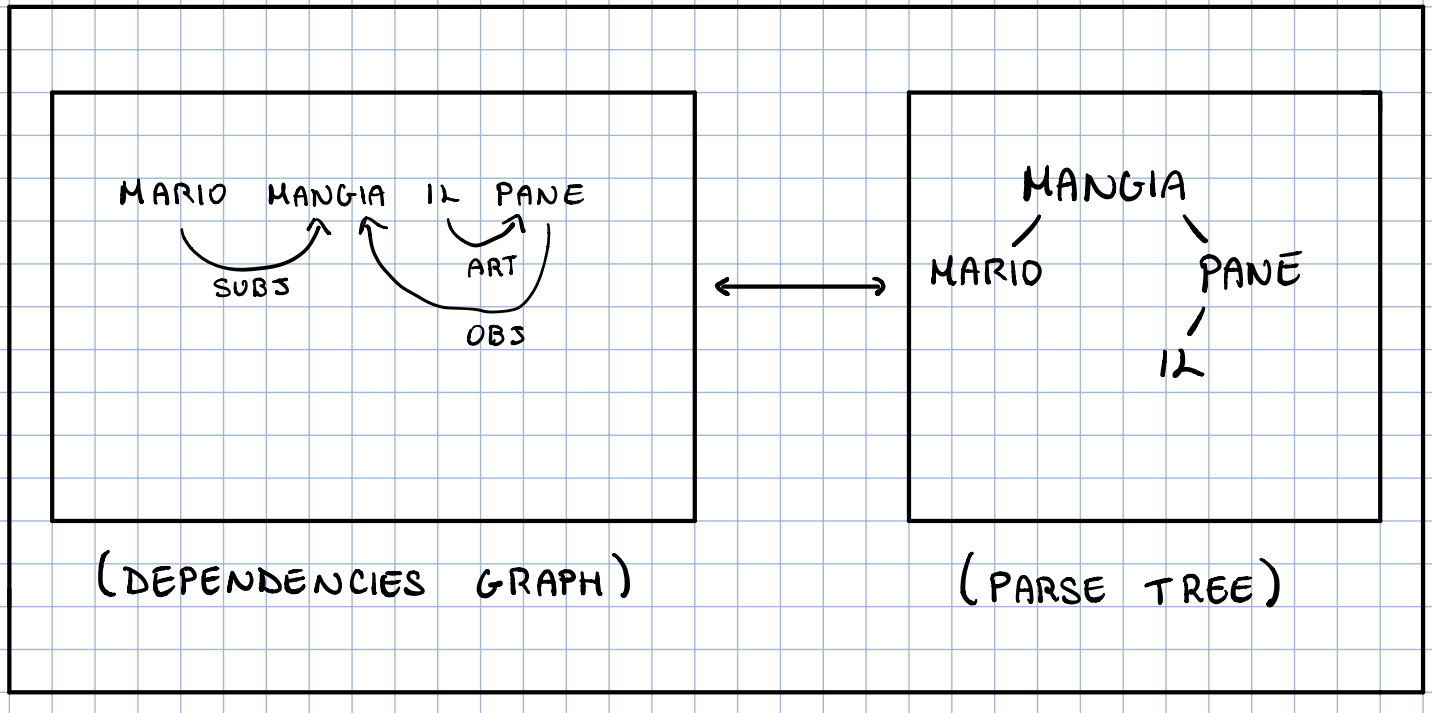
Oltre alla rappresentazione basata sui SINTAGMI che genera noi i PARSE TREE abbiamo un altro metodo per rappresentare le informazioni sintattiche, che prende il nome di RAPPRESENTAZIONE ALLE DIPENDENZE.

Questa nuova rappresentazione non introduce degli elementi intermedi ma lavora direttamente con le parole, descrivendo come queste ultime sono collegate tra loro tramite un GRAFO ETICHETTATO.

Date una frase come "MARIO MANGIA IL PANE", otteniamo la seguente rappresentazione

Notiamo che in alcuni casi è possibile passare da una rappresentazione all'altra. Per fare questo dobbiamo capire qual è l'elemento nella rappr. alle dipendenze che vogliamo far salire di livello. L'elemento che facciamo salire è detto HEAD.

Nel caso di prima abbiamo



Notiamo però che con la rappr. alle dipendenze possiamo anche far INCROCIARE gli archi. In questo caso i grafi ottenuti vengono detti NON-PROJECTIVES, e non vale più l'equivalenza tra le rappresentazioni.

OSS: Tutti gli alberi presenti nelle UNIVERSAL DEPENDENCIES non PROJECTIVE.

Per adesso l'unico modo per ottenere la rappr. alle dipendenze che conosciamo è quello di utilizzare il CYK o l'EARLEY per ottenere il PARSE TREE e poi da questo ottenere il DEPENDENCY TREE/GRAPH. Esistono poi degli algoritmi che costruiscono direttamente la rap. alle dipendenze.

OSS: Inizialmente la rappr. alle dipendenze non è stata presa molto in considerazione. Successivamente con la crescita del MACHINE LEARNING (ML) è stata rivista e ripresa in considerazione.