Queen Mary
University of London

Natural Language Processing (NLP)

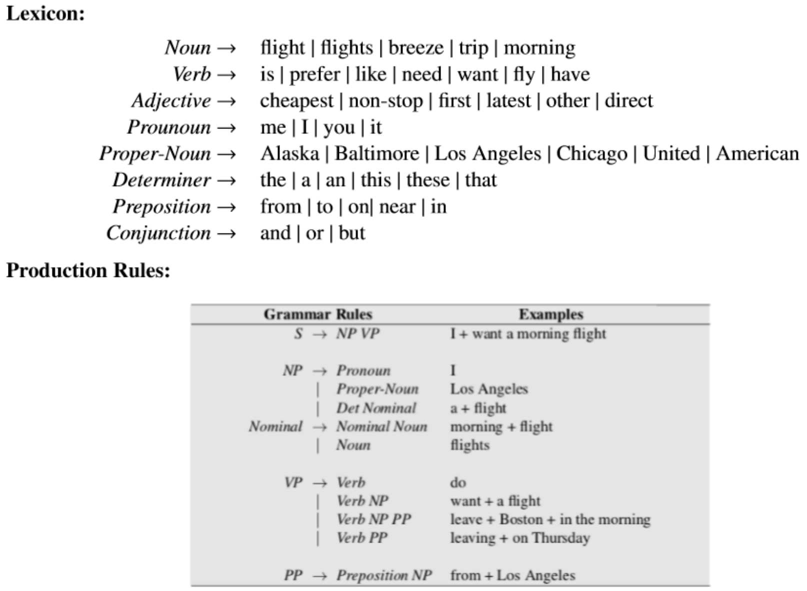
Lab3-4

Spyridon Roumpis

181004877

Lab3

PartA.GenerativeGrammars



Lexicon:

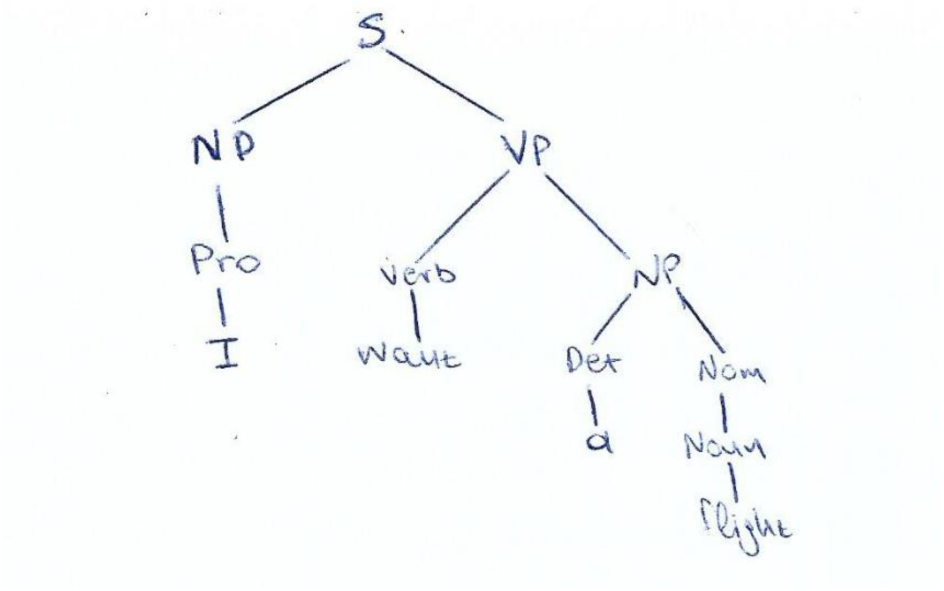
Noun →	flight | flights | breeze | trip | morning
Verb →	is | prefer | like | need | want | fly | have
Adjective →	cheapest | non-stop | first | latest | other | direct
Prounoun →	me | I | you | it
Proper-Noun →	Alaska | Baltimore | Los Angeles | Chicago | United | American
Determiner →	the | a | an | this | these | that
Preposition →	from | to | on| near | in
Conjunction →	and | or | but

Production Rules:

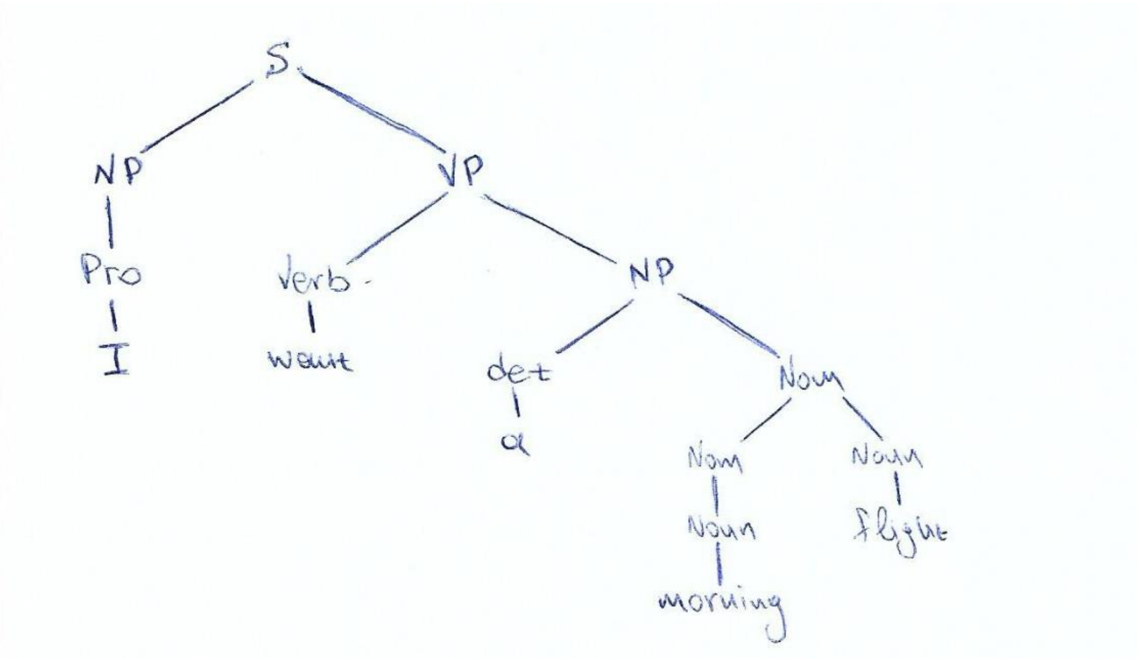
Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

1. Draw parse trees for the following sentences of natural language:

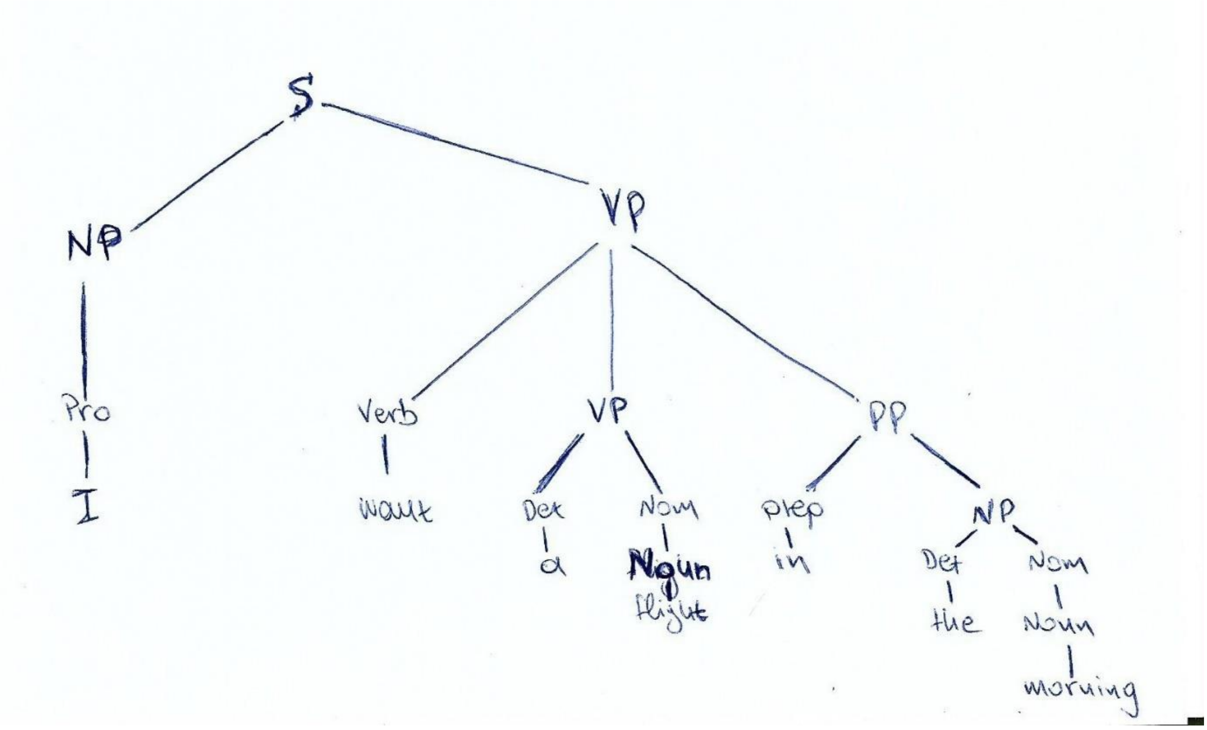
1. I want a flight



2. I want a morning flight



3. I want a flight in the morning.



2.Add production rules with VP on the left-hand side to be able to parse the following sentences. Write down all the new rules needed to be able to parse all three sentences. You only need to give the new rules required, not the parse trees:

1. I want a flight in the morning from Baltimore.

2. I want a flight from Baltimore to Los Angeles.

3. I need a trip in the morning from Baltimore to Los Angeles.

The extra rules are: VP→verb NP PP PP and VP→verb NP PP PP PP

3. To be able to parse questions, add the following items to the lexicon

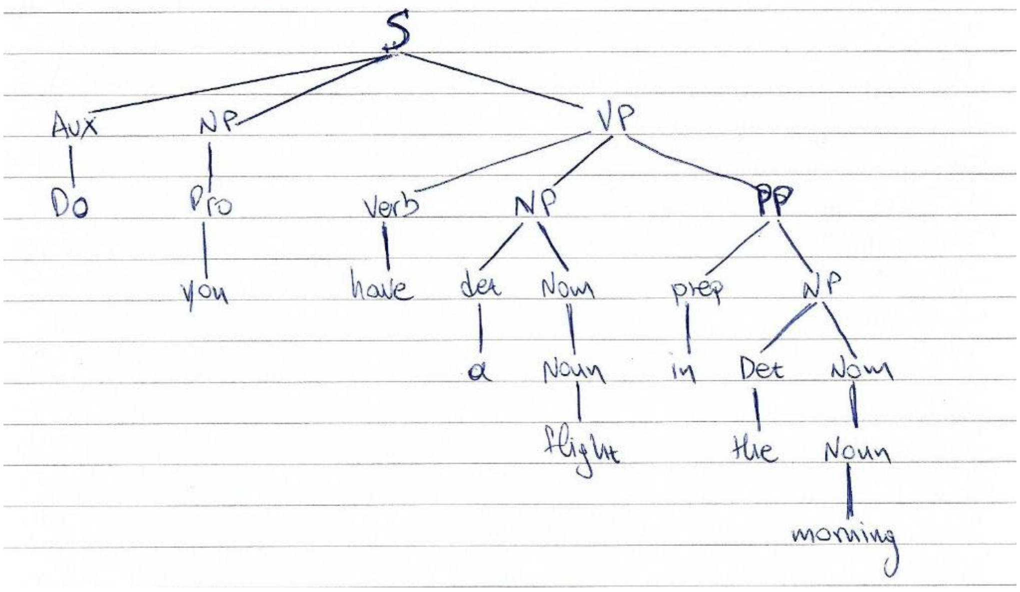
Aux→do and Wh-word→what

and add the following production rules to the grammar:

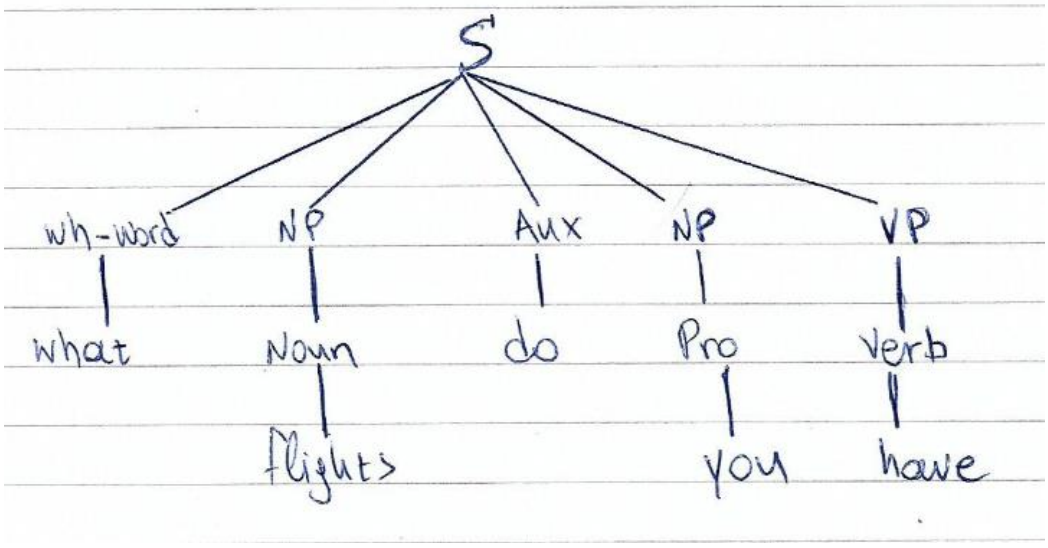
NP→Nominal Wh-NP→Wh-word NP S→Aux NP VP S→Wh-NP Aux NP VP

With these new additions to the grammar included, draw parse trees for the following questions:

1. Do you have a flight in the morning?



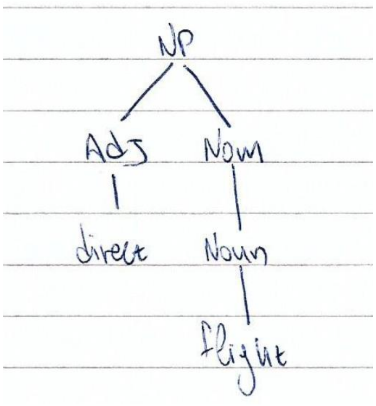
2. What flights do you have?



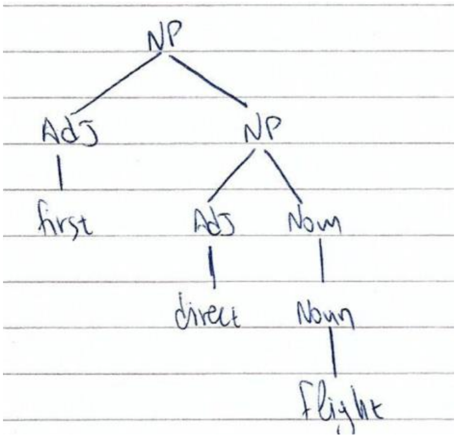
4. Add the minimum number of necessary rule(s) using adjectives needed to be able to parse the following noun phrases. Write down the rule(s) needed and then draw the parse trees for the following noun phrases (note, they are not full sentences and the top of the trees should be NP):

The extra rules are: NP→Adj Noun and NP→Adj NP

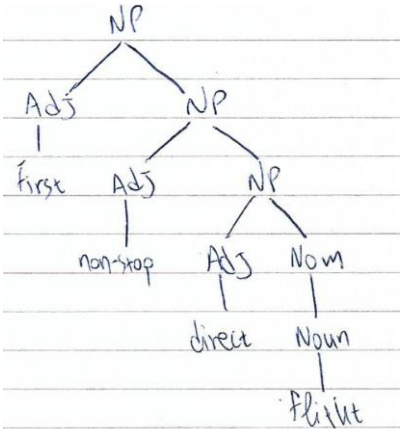
1. direct flight



2. first direct flight



3. first non-stop direct flight



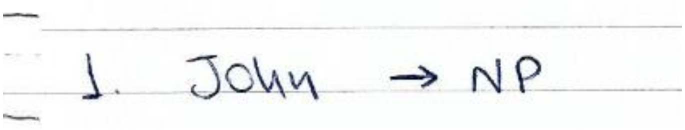
PartB.LogicalGrammars

1. Use the following type assignments and the directional AB calculus of Bar-Hillel to derive syntactic trees for the following phrases and sentences of natural language:

Word	Type
John | Mary | pizza | pasta | bed	NP
slept | snored	$NP\backslash S$
ate	$(NP\backslash S)/NP$
and	$X\backslash X/X$

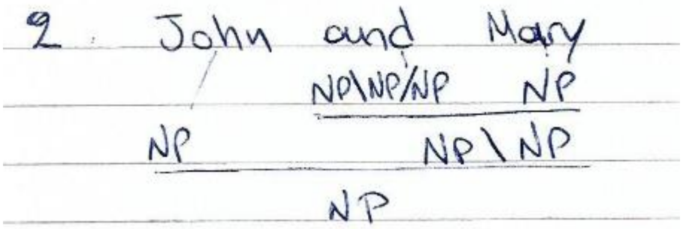
Whenever the conjunctive word "and" is used, please clearly specify what X is, as you can specify any type. Note these are not all full sentences, so the S won't be derived in all cases.

1. John.



1. John → NP

2. John and Mary



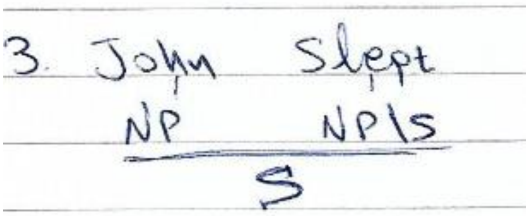
2. John and Mary

NP\NP/NP NP

NP NP\NP

NP

3. John slept

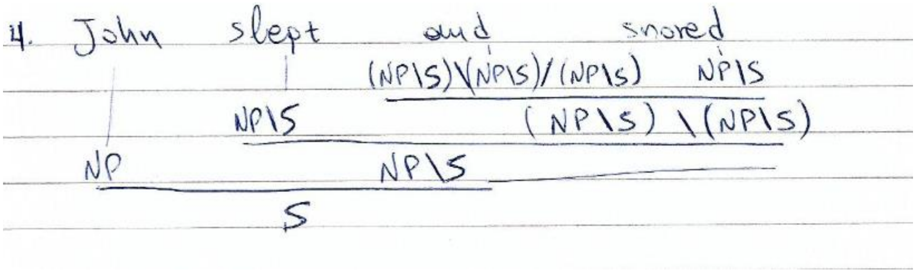


3. John Slept

NP NP\S

S

4. John slept and snored



4. John slept and snored

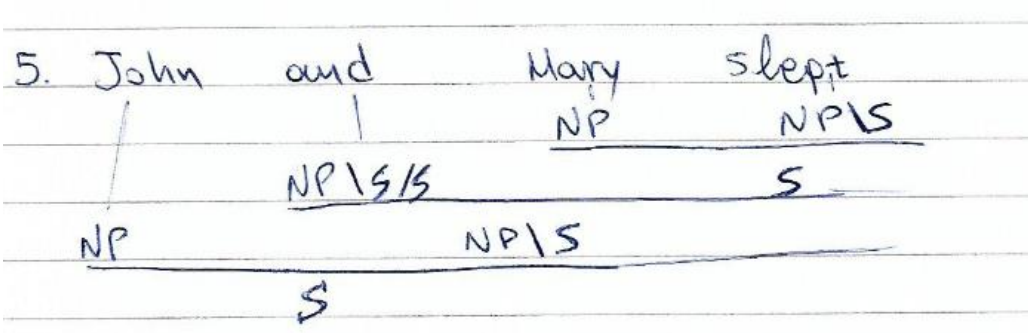
(NP\S)\(NP\S)/(NP\S) NP\S

NP\S (NP\S) \(NP\S)

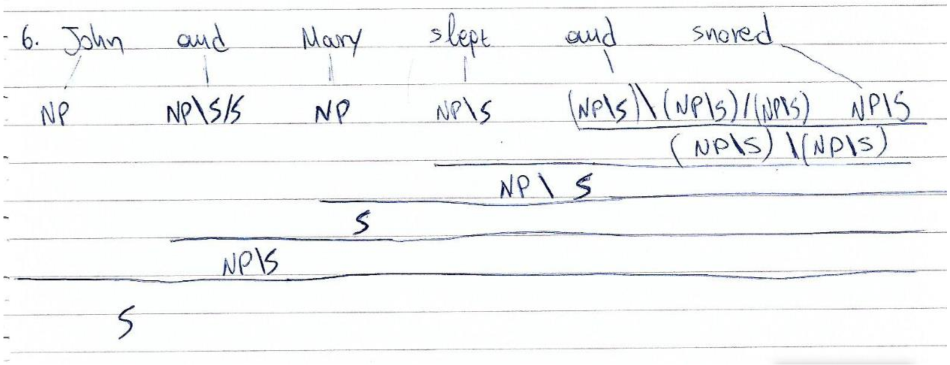
NP NP\S

S

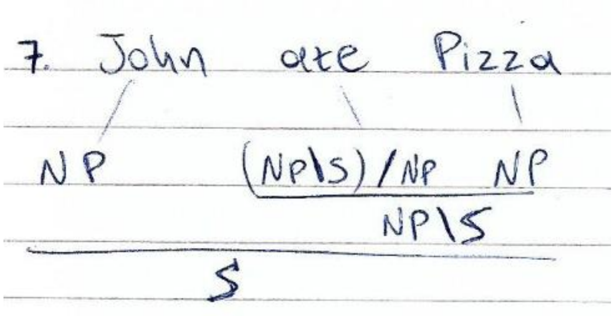
5. John and Mary slept



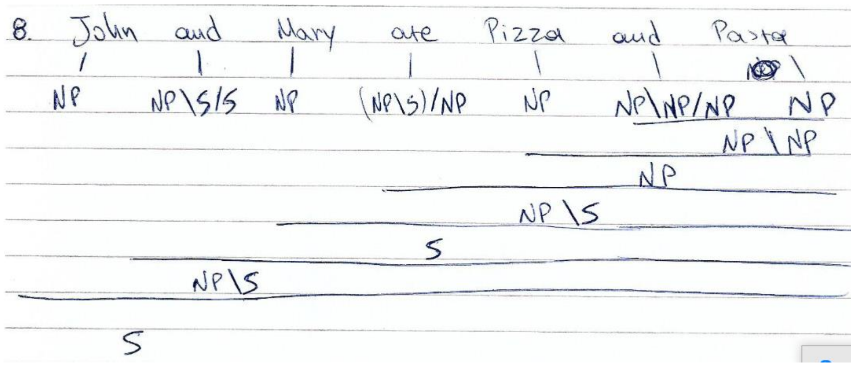
6. John and Mary slept and snored



7. John ate Pizza



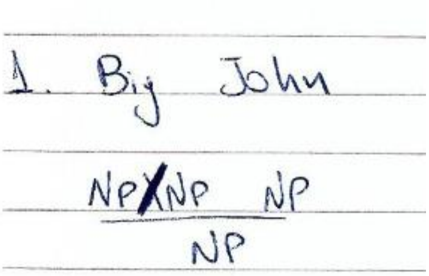
8. John and Mary ate Pizza and Pasta.



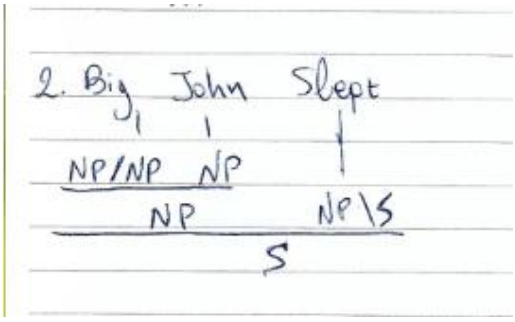
2. Come up for a type for the adjective “big” and use it to derive syntactic trees for the following. Give the tree derivations:

The new type for adjective “big” is NP/NP

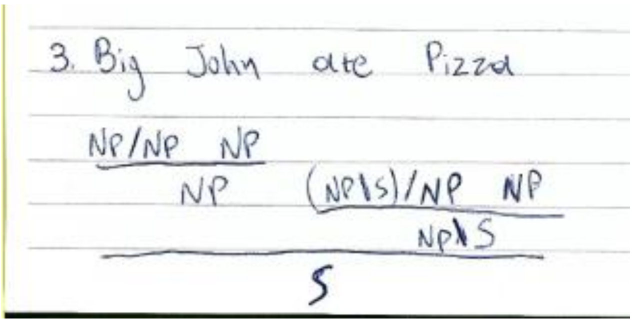
1. Big John.



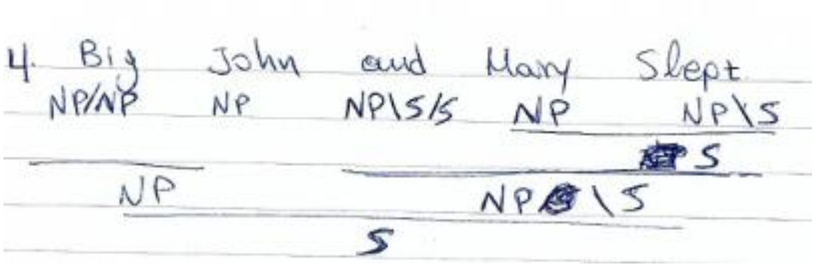
2. Big John slept.



3. Big John ate Pizza.



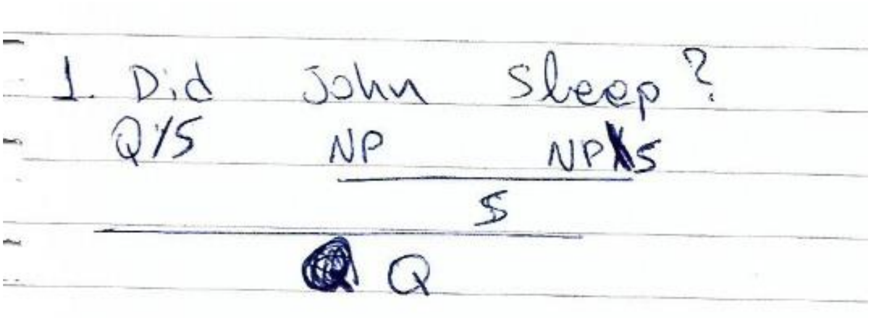
4. Big John and Mary slept.



3. Add the type Q to the designated types of your AB grammar. Assign a type to "did" to be able to provide syntactic trees for the following questions. Note that these sentences don't necessarily need to lead to an S derivation- they can lead just to the Q symbol. You can therefore think of Q as alternative to S for the final derived type- it stands for a question rather than a declarative sentence.

The new type for adjective "did" is Q/S

1. Did John sleep?



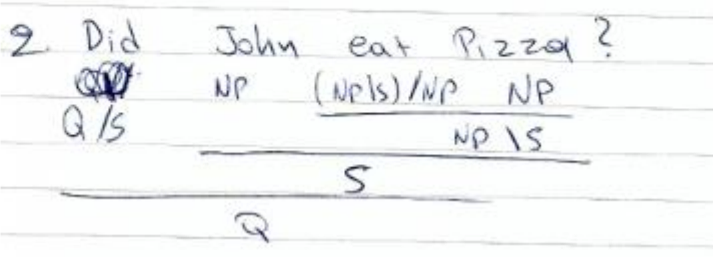
1. Did John sleep?

Q/S NP NP\S

S

Q

2. Did John eat Pizza?



2. Did John eat Pizza?

Q/S NP (NP\S)/NP NP

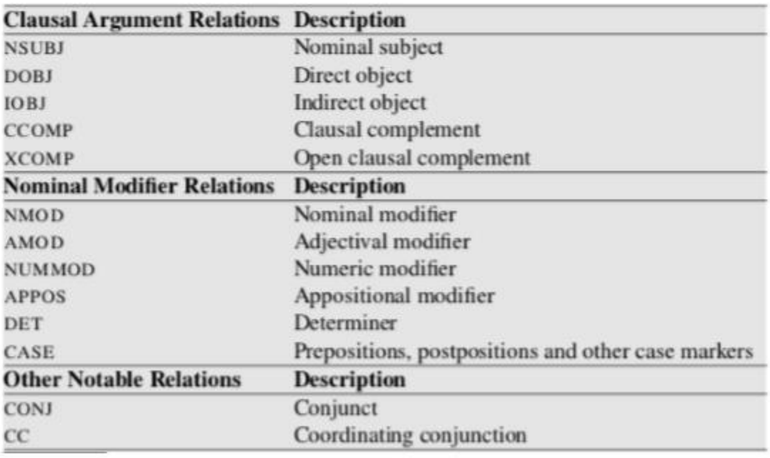
NP\S

S

Q

Lab4

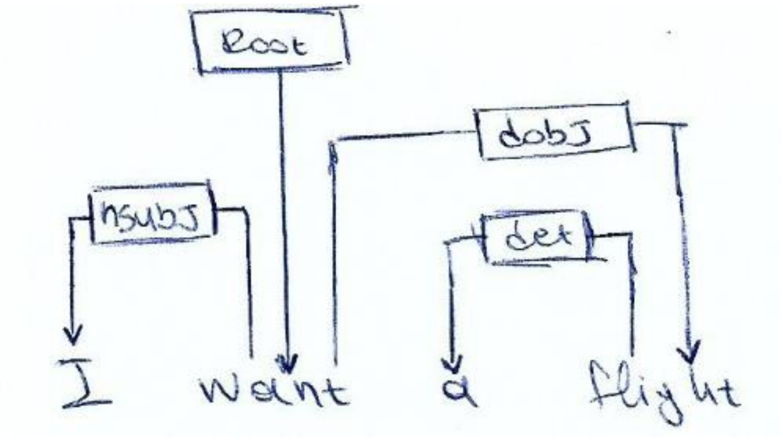
PartA.DependencyGrammars



Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

1. Draw dependency graphs for the following sentences. For each case, specify the longest path (or paths) of the tree and its(their)length(s). Based on this, argue which sentence is the one with the most "involved" grammatical structure.

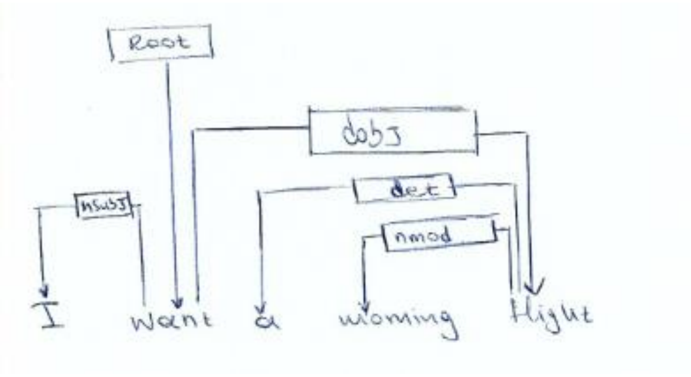
1. I want a flight



Longest Path: Want→flight→a

Length : 2

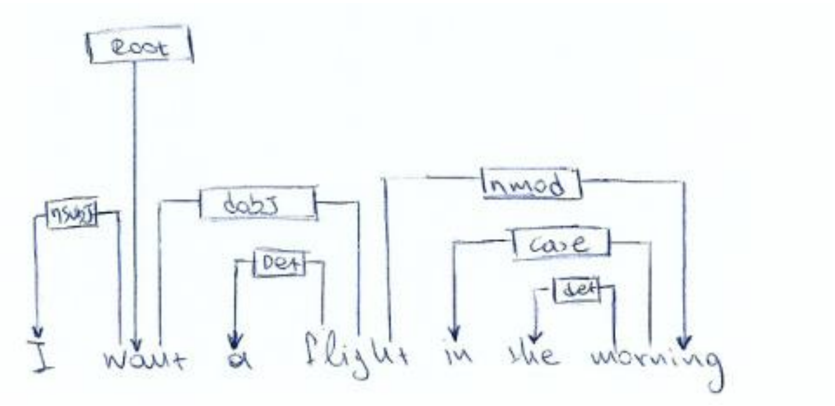
2. I want a morning flight



Longest Path: Want→flight→a or Want→flight→morning

Length: 2

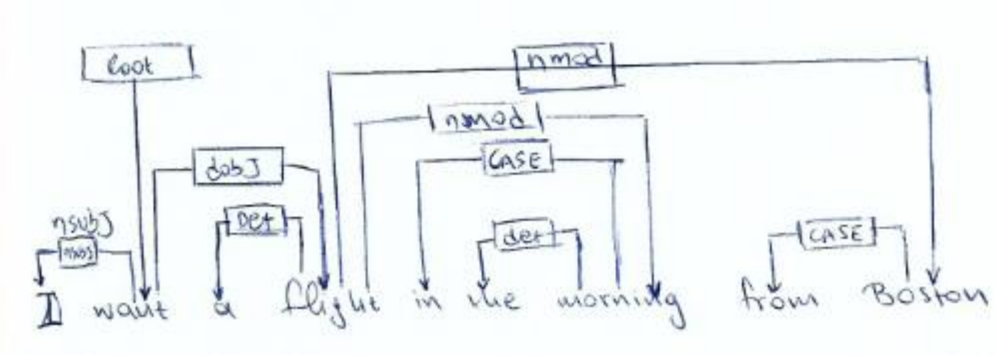
3. I want a flight in the morning



Longest Path: Want→flight→morning→in or Want→flight→morning→the

Length: 3

4. I want a flight in the morning from Boston



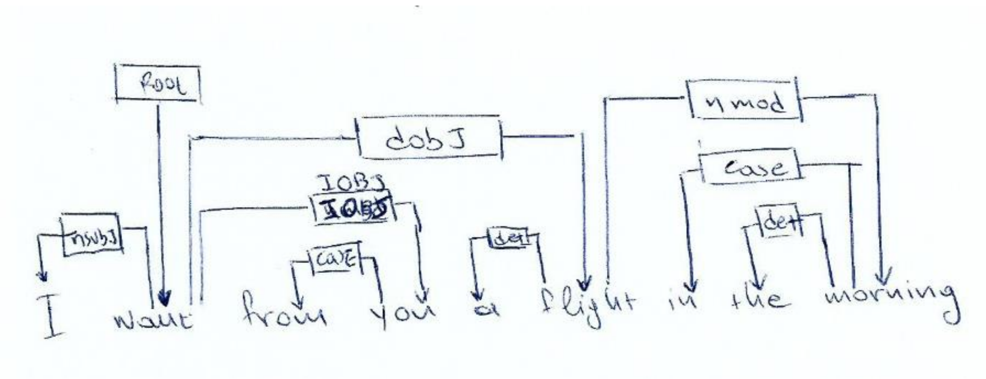
Longest Path: Want→flight→morning→Boston→from

Length: 4

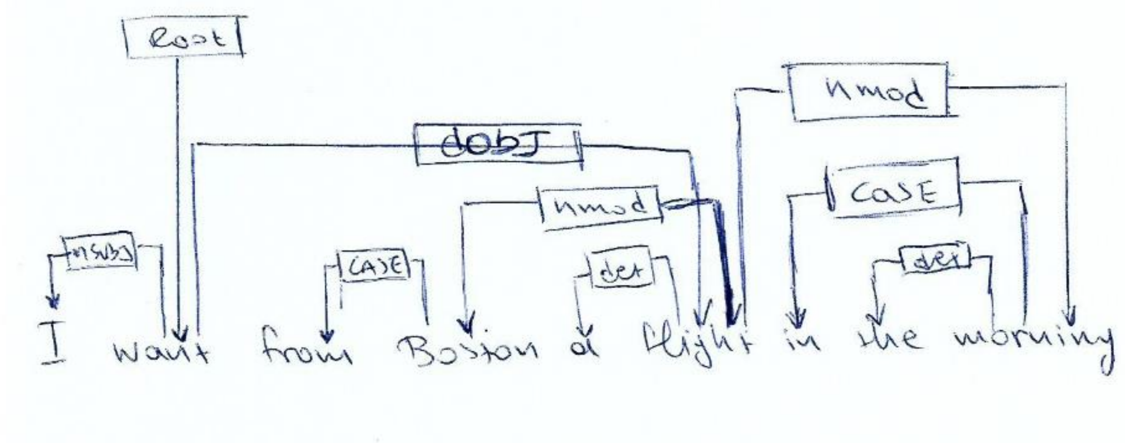
2. Draw dependency graphs for the following sentences and in each case explain (briefly but clearly, similar to the explanation of the examples in the lecture and the Jurafsky and Martin book) if the graph has an arc which is not 'projective':

From theory an arc between v and w in a dependency tree is called projective if there is a path between v and every other word that occurs in the sentence between v and w. A dependency tree is projective if all its arc are.

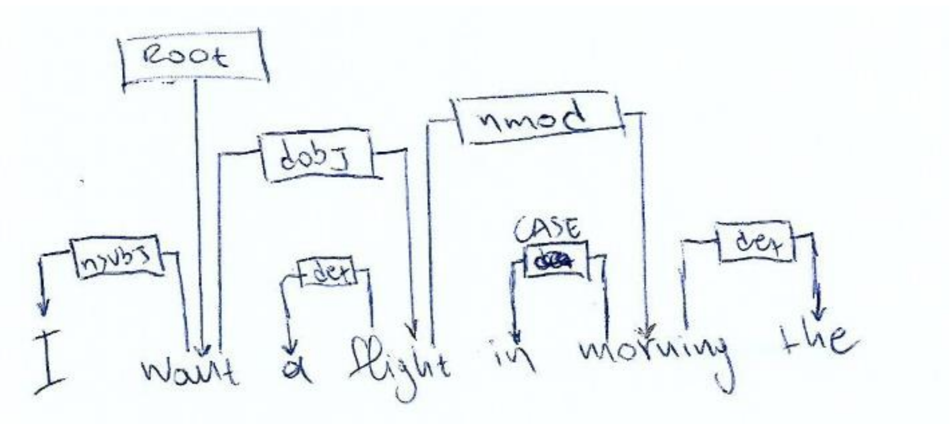
1. I want from you a flight in the morning, it is projective.



2. I want from Boston a flight in the morning, it is projective.

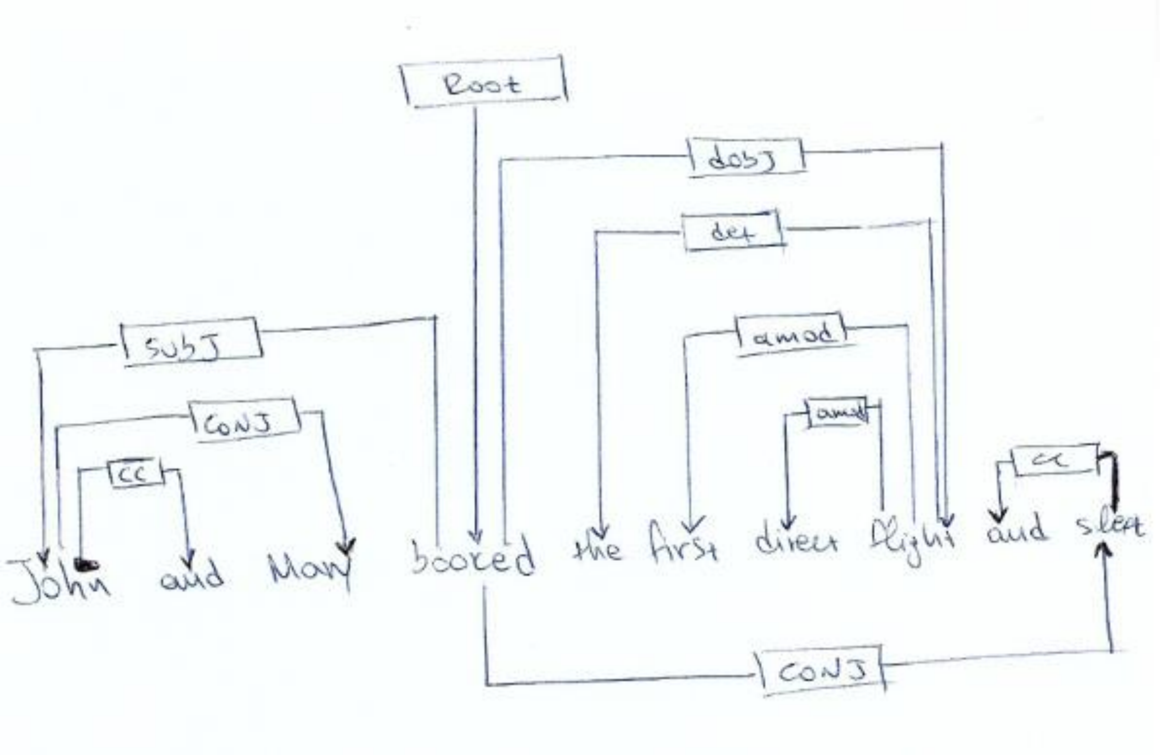


3. I want a flight in morning the, it is projective.

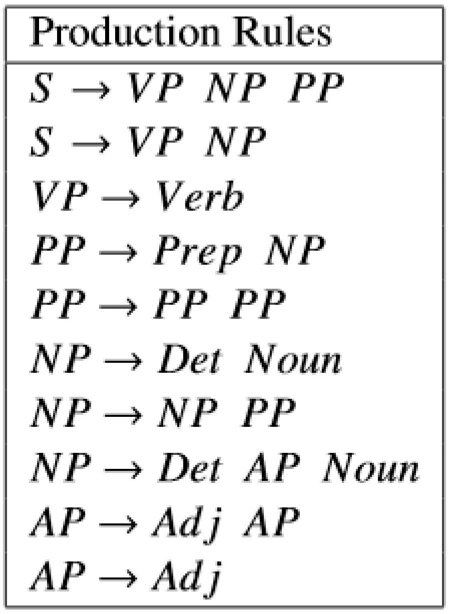


3. Using the table of universal dependency relations, draw a dependency tree for the following sentence.

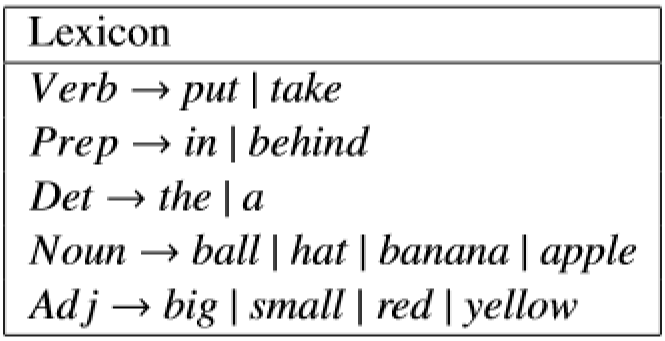
1. John and Mary booked the first direct flight and slept



PartB.CKYParsingwithContextFreeGrammars



Production Rules
$S \rightarrow VP\ NP\ PP$
$S \rightarrow VP\ NP$
$VP \rightarrow Verb$
$PP \rightarrow Prep\ NP$
$PP \rightarrow PP\ PP$
$NP \rightarrow Det\ Noun$
$NP \rightarrow NP\ PP$
$NP \rightarrow Det\ AP\ Noun$
$AP \rightarrow Adj\ AP$
$AP \rightarrow Adj$



Lexicon
$Verb \rightarrow put \mid take$
$Prep \rightarrow in \mid behind$
$Det \rightarrow the \mid a$
$Noun \rightarrow ball \mid hat \mid banana \mid apple$
$Adj \rightarrow big \mid small \mid red \mid yellow$

1. To use CKY parsing to parse sentences with a Context Free Grammar (CFG), the CFG production rules first have to be converted to Chomsky Normal Form. Re-write the below CFG grammar production rules (and lexical rules, if needs be) such that the resulting rules and lexicon are all in Chomsky Normal Form, adding extra rules where necessary. Some rules may already be in the correct form, so do not need to change.

The new production rules are:

X0→NP PP

S→VP X0

X1 →VP NP

S →X1 PP

S →X1

VP→Verb

PP→ Prep NP

PP→ PP PP

NP→ Det Noun

NP→ NP PP

X2 →Det AP

X3→AP Noun

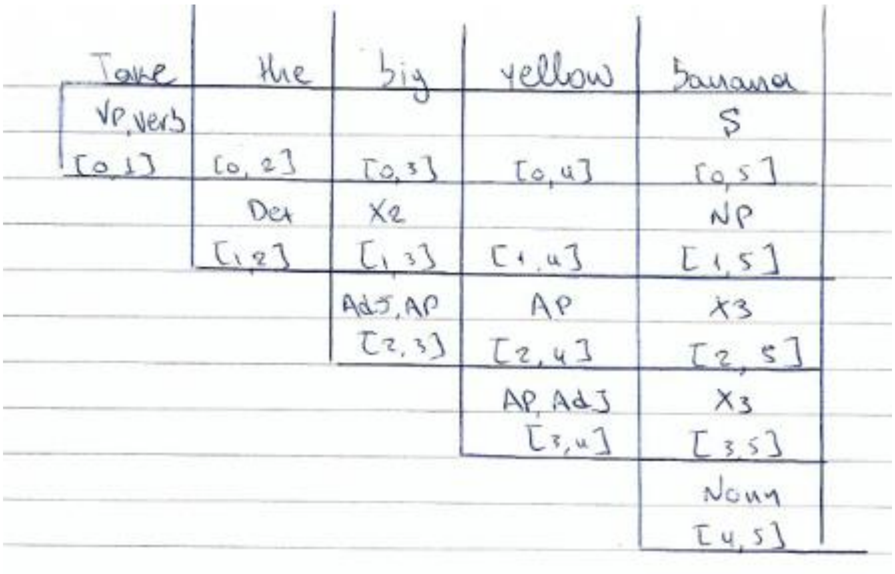
NP→ X2 Noun

NP→Det X3

AP→ Adj AP

AP→ Adj

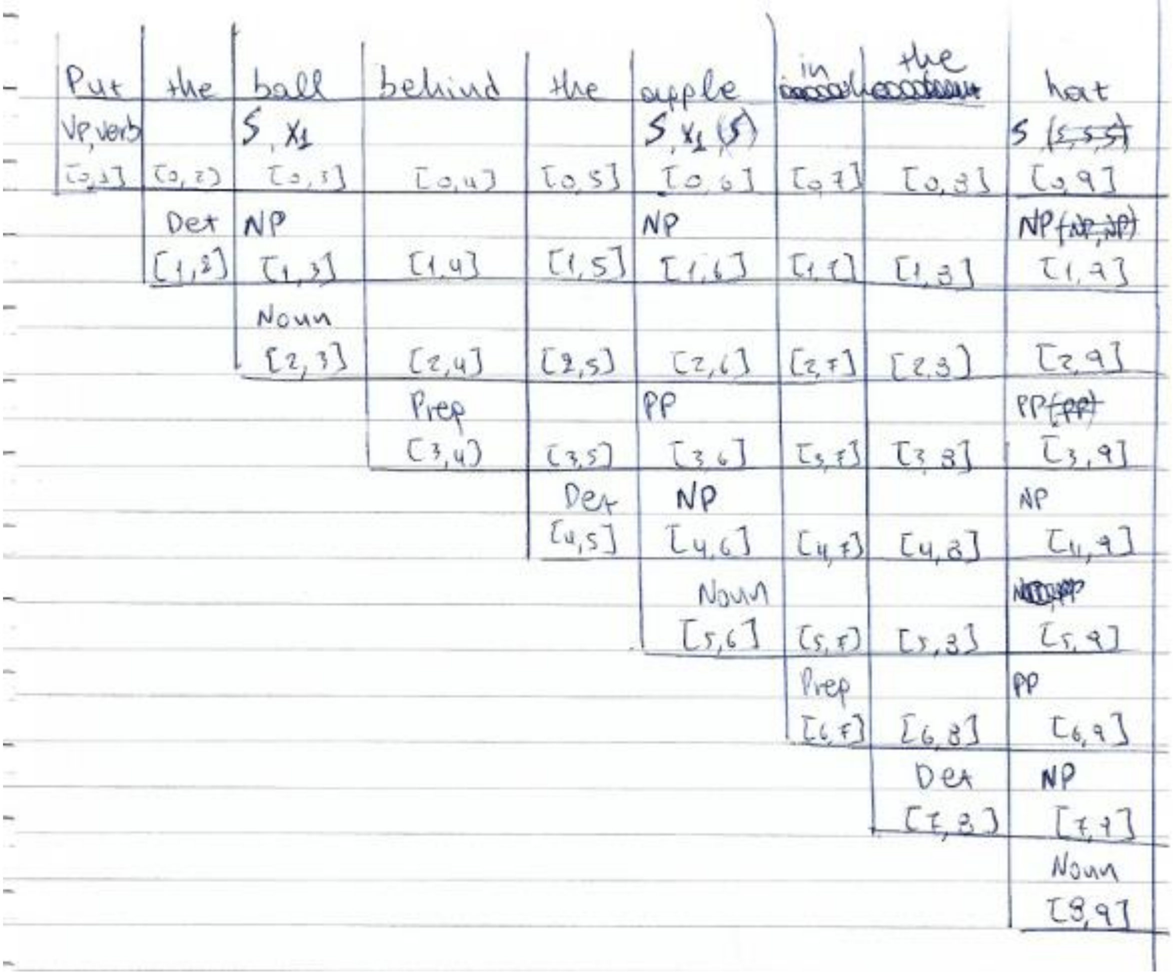
2. Given the new CNF rules you have just written, now use them to parse the below sentence (by hand, not with code) using the CKY algorithm: “Take the big yellow banana”



Take	the	big	yellow	banana
VP,verb [0,1]	[0, 2]	[0,3]	[0,4]	S [0,5]
	Det [1,2]	X2 [1,3]	[1,4]	NP [1,5]
		Adj,AP [2,3]	AP [2,4]	X3 [2, 5]
			AP, Adj [3,4]	X3 [3,5]
				Noun [4,5]

There is one full parse (that derive S) from the above sentence “Take the big yellow banana”

3. With the same grammar you used for the last question, fill in a parse matrix using the CKY method for the following sentence: ‘Put the ball behind the apple in the hat’.



Put	the	ball	behind	the	apple	in	the	hat
VP, verb [0,1]	[0,2]	S, X_1 [0,3]	[0,4]	[0,5]	S, X_1 (S) [0,6]	[0,7]	[0,8]	S ~~(S, S, S)~~ [0,9]
	Det [1,2]	NP [1,3]	[1,4]	[1,5]	NP [1,6]	[1,7]	[1,8]	NP ~~(NP, NP)~~ [1,9]
		Noun [2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]	[2,9]
			Prep [3,4]	[3,5]	PP [3,6]	[3,7]	[3,8]	PP ~~(PP)~~ [3,9]
				Det [4,5]	NP [4,6]	[4,7]	[4,8]	NP [4,9]
					Noun [5,6]	[5,7]	[5,8]	[5,9]
						Prep [6,7]	[6,8]	PP [6,9]
							Det [7,8]	NP [7,9]
								Noun [8,9]

There are three full parses (that derive S) from the above sentence ‘Put the ball behind the apple in the hat’.