Natural Language Processing

Course Work 2: Formal Grammars and Parsing

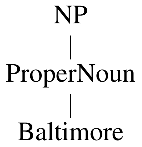
Based on chapters 11,12,13 of Jurafsky and Martin

Posted on February 18th 2018, $\boxed{\text{Due on 11th of March 2018}}$

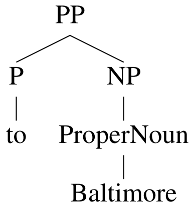
Please submit via QMPlus.

1. Use any of the rules provided in Chapter 11 (or 12 in the 2nd edition) of the Jurafsky and Martin book (available on the QMPlus webpage of the course) to draw CFG parse trees for the following phrases. In each case list the rules that you used and a page number in which the rule is presented.

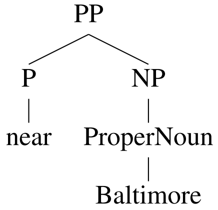
(a) Baltimore (1 point) ⇝



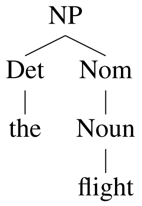
(b) to Baltimore (1 point) ⇝



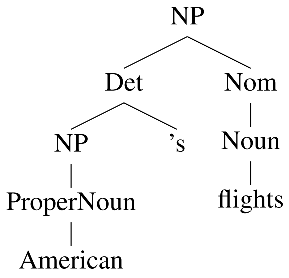
(c) near Baltimore (1 point) ⇝



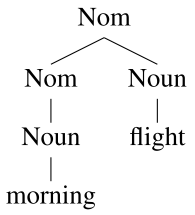
(d) the flight (1 point) ⇝



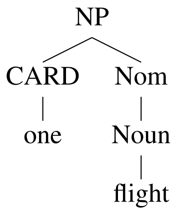
(e) American's flights (1 point) ⇝



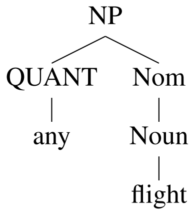
(f) morning flight (1 point) ⇝



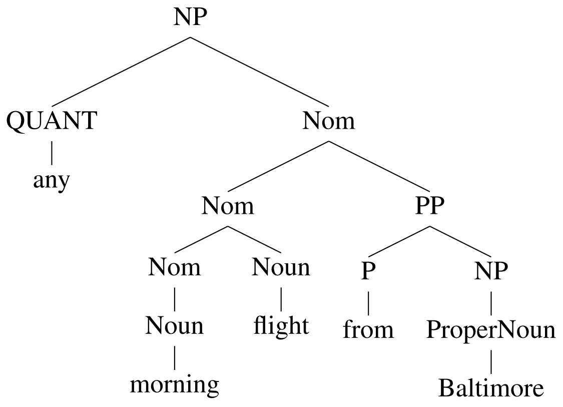
(g) one flight (1 point) ⇝



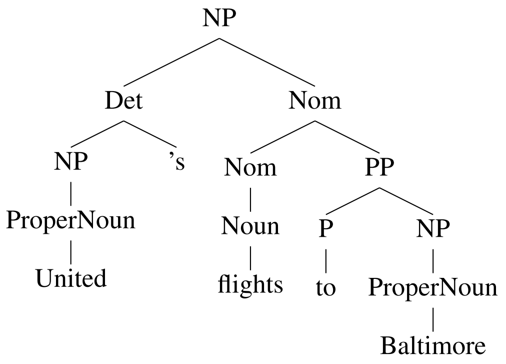
(h) any flight (1 point) ⇝



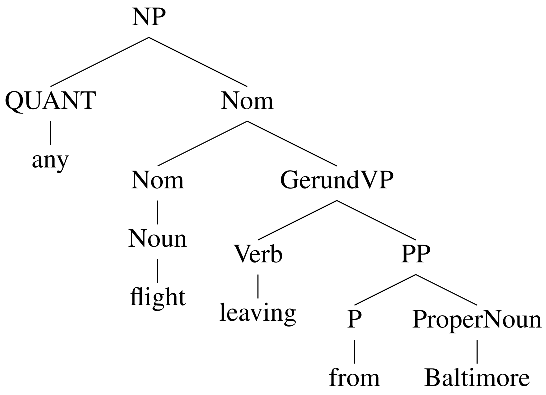
(i) any morning flight from Baltimore (2 points) ⇝



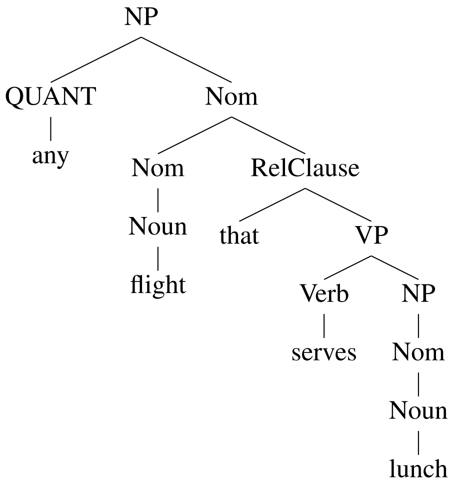
(j) United's flight to Baltimore (2 points) ⇝



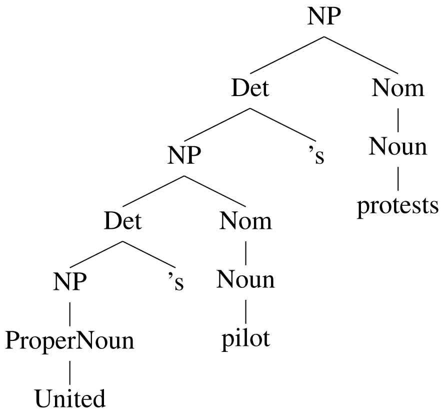
(k) any flight leaving from Baltimore (2 points) ⇝



(l) any flight that serves lunch (2 points) ⇝

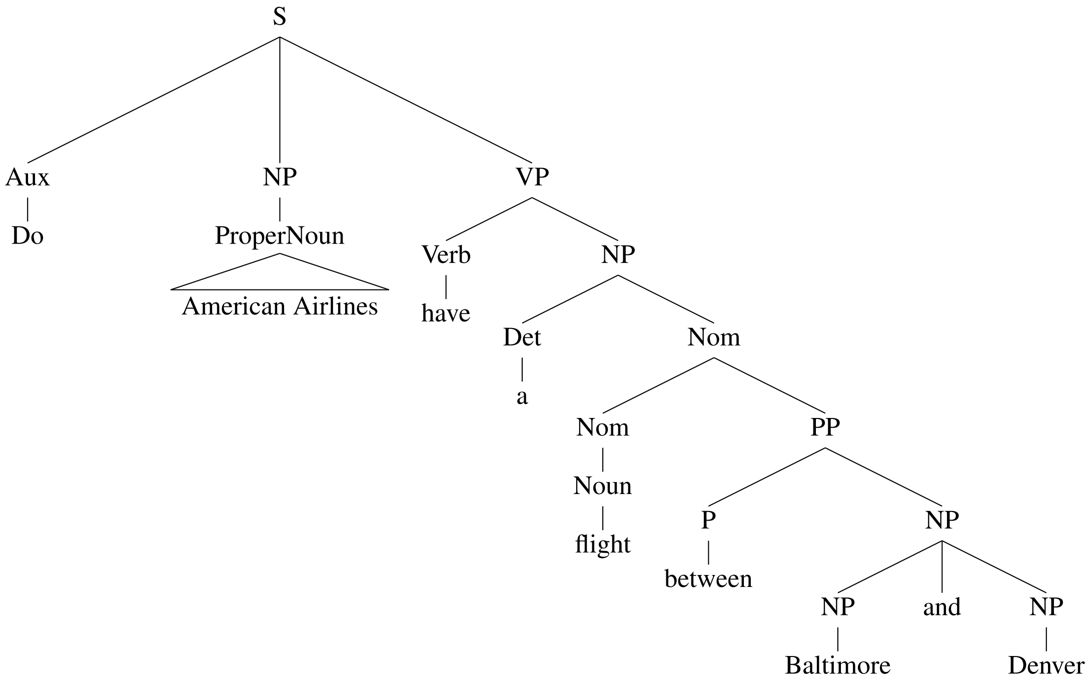


(m) United’s pilot’s protests (4 points) ⇝

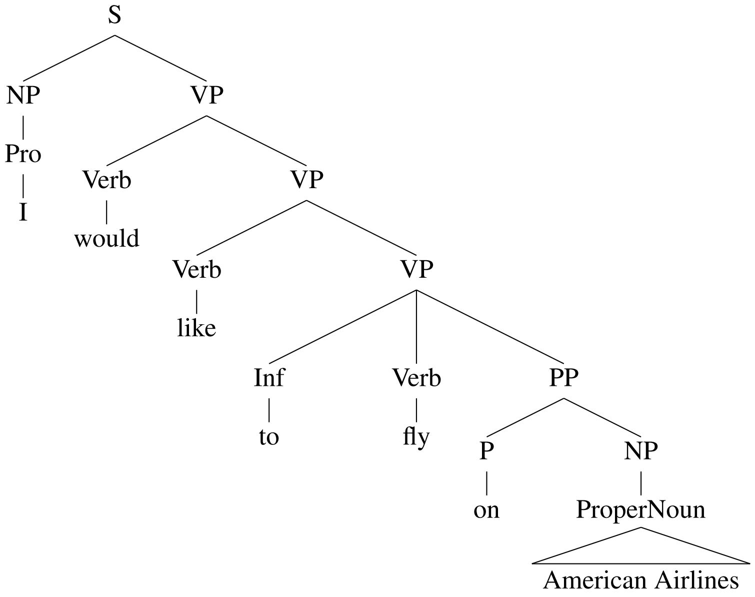


2. Do the same as in Question 1, following the same procedure, but now for the following sentences: (22 points)

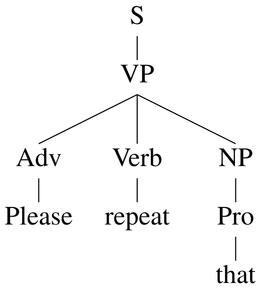
   (a) Do American Airlines have a flight between Baltimore and Denver? (3 points)



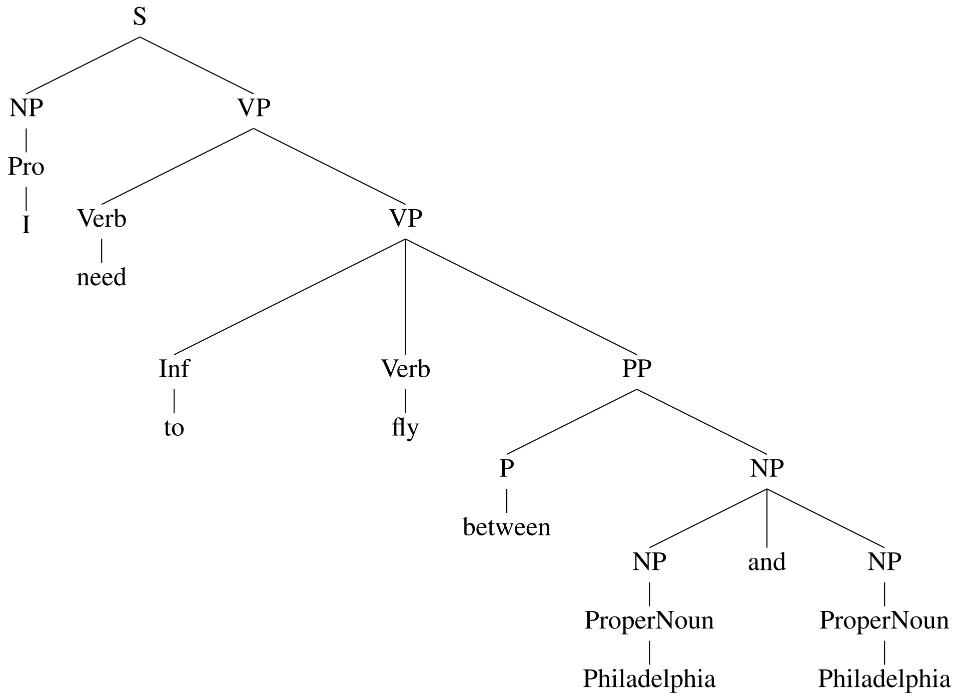
(b) I would like to fly on American airlines. (4 points)



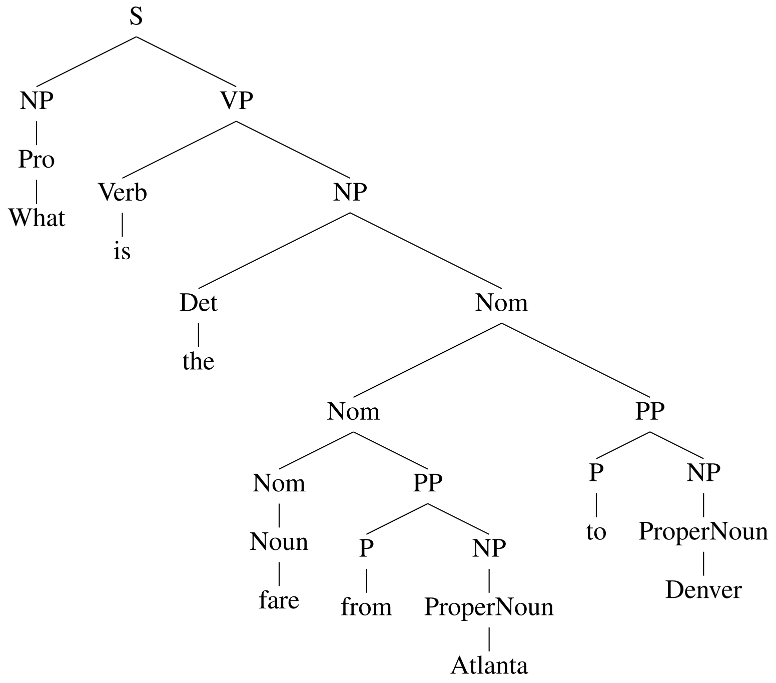
(c) Please repeat that. (5 points)



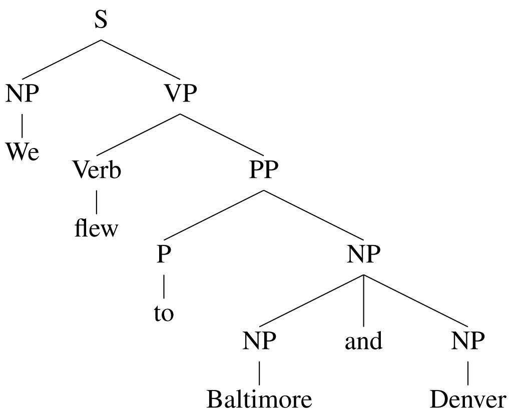
(d) I need to fly between Philadelphia and Atlanta. (3 points)



(e) What is the fare from Atlanta to Denver? (4 points)



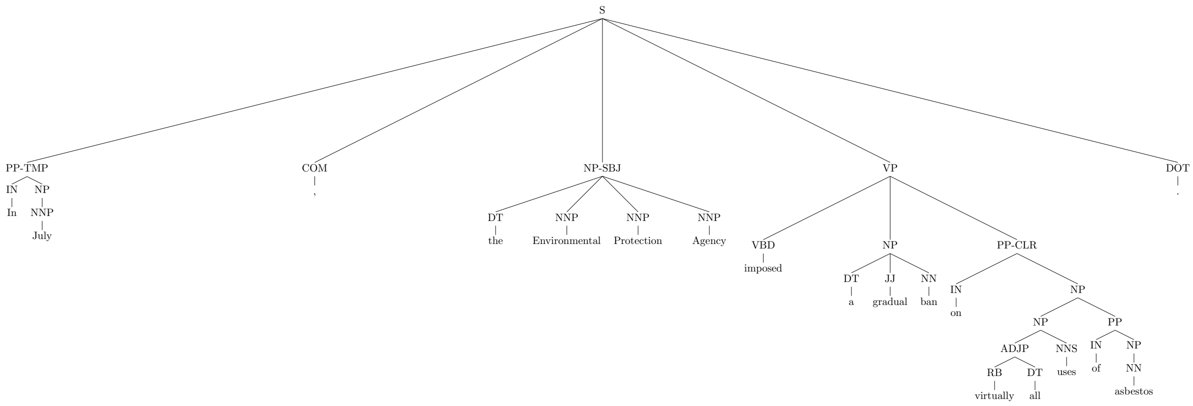
(f) We flew to Baltimore and Denver. (3 points)



3. The provided script `grammar_script.py` contains three parsed sentences from the Penn Treebank. Using this script, draw the CFG parse trees (4 points for the first and last trees, 2 points for the second tree) for these sentences and extract the set of CFG rules that are needed to parse the original sentences (2 points for the first and last trees, 1 point for the second tree). Do not use any NLTK routines, beside those given in the script. (15 points)

   Solution: please see `grammar_sol.py`. *We give the trees and the rules per tree:*

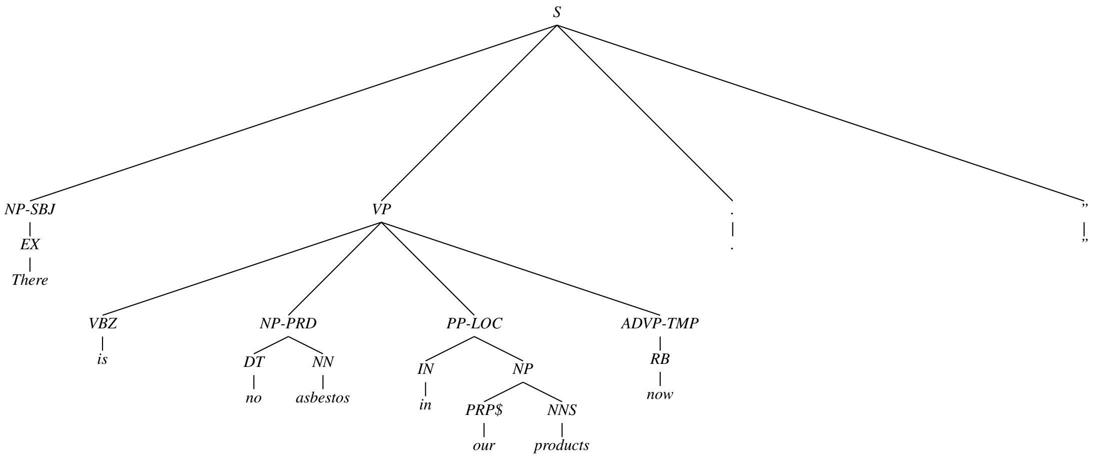
   (a) *In July*



Rules:

$$\begin{array}{cc}
S \rightarrow PP-TMP\ ,\ NP-SBJ\ VP\ . & JJ \rightarrow gradual \\
PP-TMP \rightarrow IN\ NP & NN \rightarrow ban \\
IN \rightarrow In & PP-CLR \rightarrow IN\ NP \\
NP \rightarrow NNP & IN \rightarrow on \\
NNP \rightarrow July & NP \rightarrow NP\ PP \\
, \rightarrow , & NP \rightarrow ADJP\ NNS \\
NP-SBJ \rightarrow DT\ NNP\ NNP\ NNP & ADJP \rightarrow RB\ DT \\
DT \rightarrow the & RB \rightarrow virtually \\
NNP \rightarrow Environmental & DT \rightarrow all \\
NNP \rightarrow Protection & NNS \rightarrow uses \\
NNP \rightarrow Agency & PP \rightarrow IN\ NP \\
VP \rightarrow VBD\ NP\ PP-CLR & IN \rightarrow of \\
VBD \rightarrow imposed & NP \rightarrow NN \\
NP \rightarrow DT\ JJ\ NN & NN \rightarrow asbestos \\
DT \rightarrow a & . \rightarrow .
\end{array}$$

   (b) *There is no Asbestos in our products now.”*



Rules:

$S \rightarrow NP-SBJ\ VP\ .\ ''$

$NP-SBJ \rightarrow EX$

$EX \rightarrow There$

$VP \rightarrow VBZ\ NP-PRD\ PP-LOC\ ADVP-TMP$

$VBZ \rightarrow is$

$NP-PRD \rightarrow DT\ NN$

$DT \rightarrow no$

$NN \rightarrow asbestos$

$PP-LOC \rightarrow IN\ NP$

$IN \rightarrow in$

$NP \rightarrow PRP\$\ NNS$

$PRP\$ \rightarrow our$

$NNS \rightarrow products$

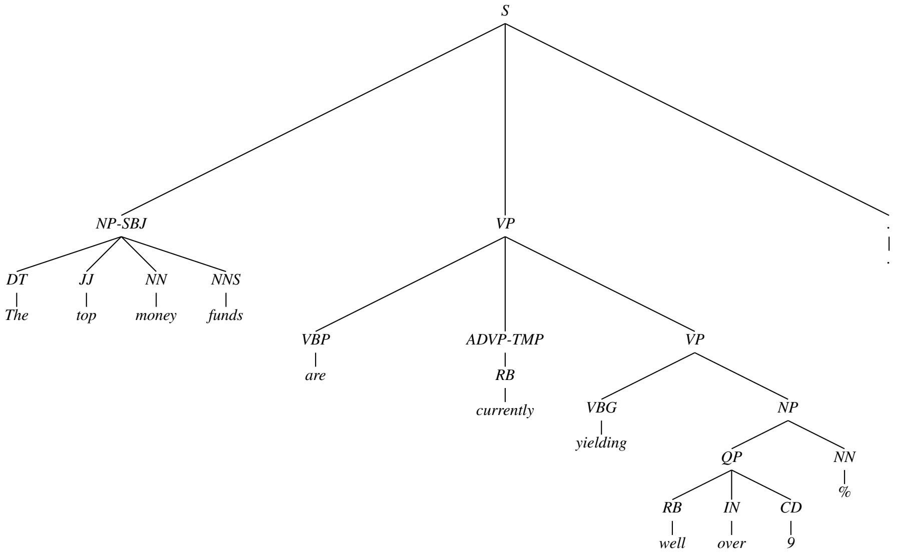
$ADVP-TMP \rightarrow RB$

$RB \rightarrow now$

$. \rightarrow .$

$'' \rightarrow ''$

(c) *The top money funds are currently yielding well over 9%.*



Rules:

$S \rightarrow NP-SBJ\ VP\ .$

$NP-SBJ \rightarrow DT\ JJ\ NN\ NNS$

$DT \rightarrow The$

$JJ \rightarrow top$

$NN \rightarrow money$

$NNS \rightarrow funds$

$VP \rightarrow VBP\ ADVP-TMP\ VP$

$VBP \rightarrow are$

$ADVP-TMP \rightarrow RB$

$RB \rightarrow currently$

$VP \rightarrow VBG\ NP$

$VBG \rightarrow yielding$

$NP \rightarrow QP\ NN$

$QP \rightarrow RB\ IN\ CD$

$RB \rightarrow well$

$IN \rightarrow over$

$CD \rightarrow 9$

$NN \rightarrow \%$

$. \rightarrow .$

4. Consider the following sentence:

   "List me the seats on the flight to Denver."

   (a) Give as many meanings for this sentence as you can (aim for more than 3). (4 points, 1 for each meaning)

   Solution: there's basically a couple of permutations on how the two prepositional phrases modify either the head noun (seats) or the main verb (List) of the phrase:

   i. The listing happens on the flight to Denver, and the seats are on the flight to Denver.

   ii. The listing happens on the flight to Denver, but the seats are not on the flight (they could be theatre seats, for instance)

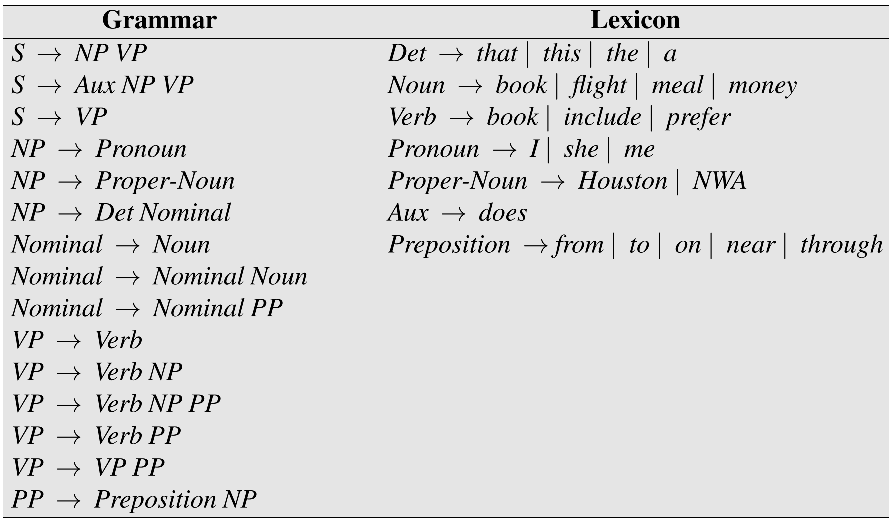
   iii. The listing does not happen on the flight to Denver, but the seats are on the flight to Denver.

   iv. The listing does not happen on the flight to Denver, and the seats are not on the flight to Denver (they could again be theatre seats)

   (b) Replace the grammar in the `grammar_script.py` file with the rules of the miniature grammar of English given in Figure 12.1, plus the rules listed below, as well as any extra lexicon rules necessary to parse the above sentence. Find all possible parses of the above sentence. (4 points, 1 for each parse tree)

   Note: in the rules below, IVP stands for imperative verb phrase, and $IVerb$ stands for imperative verb.

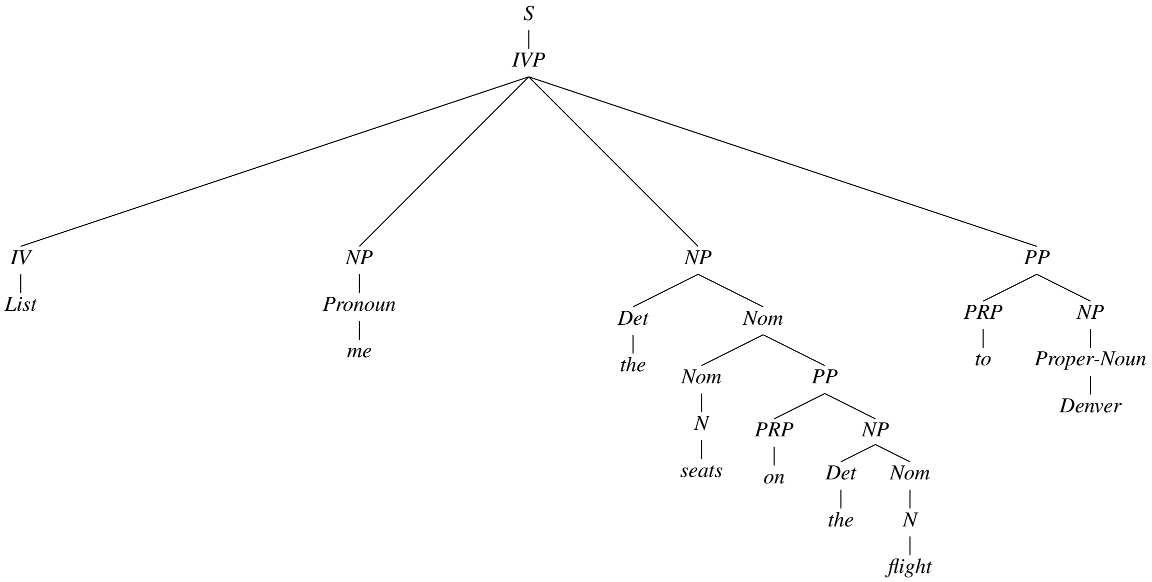
$$\begin{array}{rcl} S & \rightarrow & IVP \\ IVP & \rightarrow & IVerb\ NP\ NP \\ IVP & \rightarrow & IVerb\ NP\ NP\ PP \end{array}$$

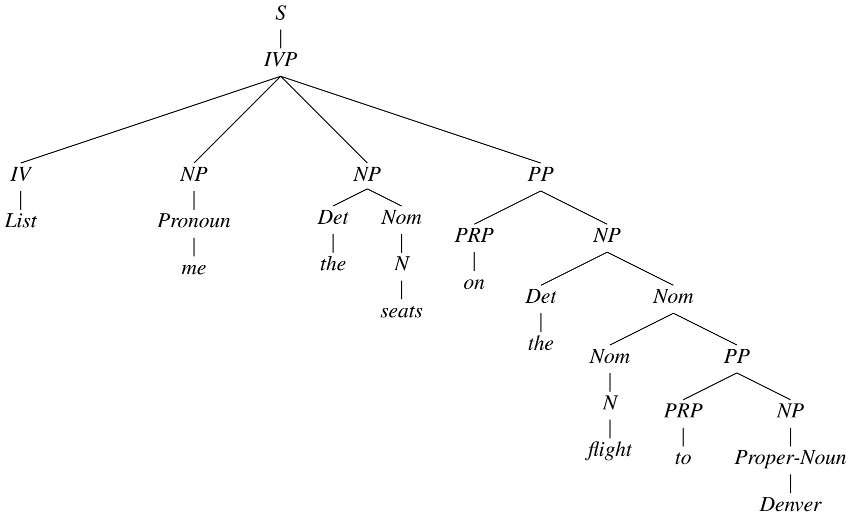


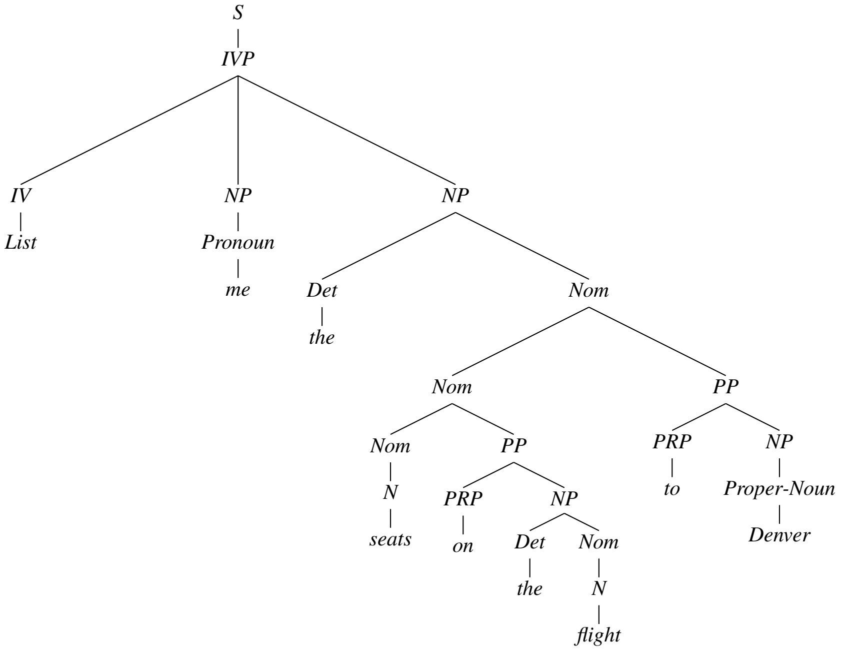
Grammar	Lexicon
S → NP VP	*Det → that | this | the | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

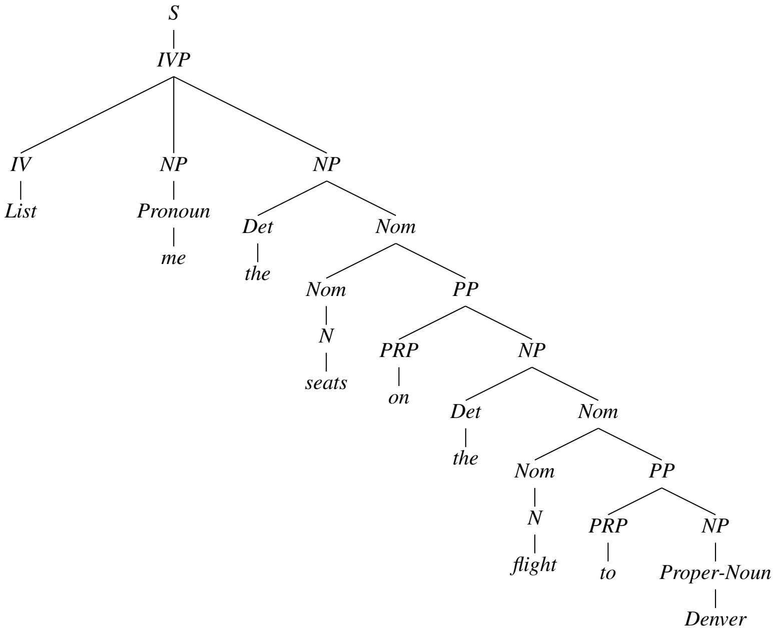
Figure 12.1 The $\mathscr{L}_1$ miniature English grammar and lexicon.

Solution: please see `grammar_sol.py`*. You should get 4 parses:*







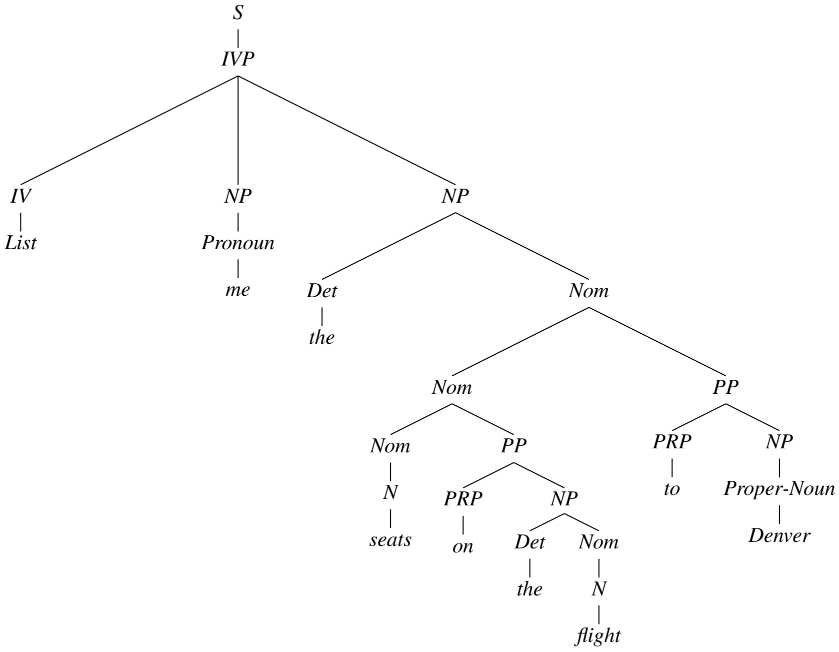


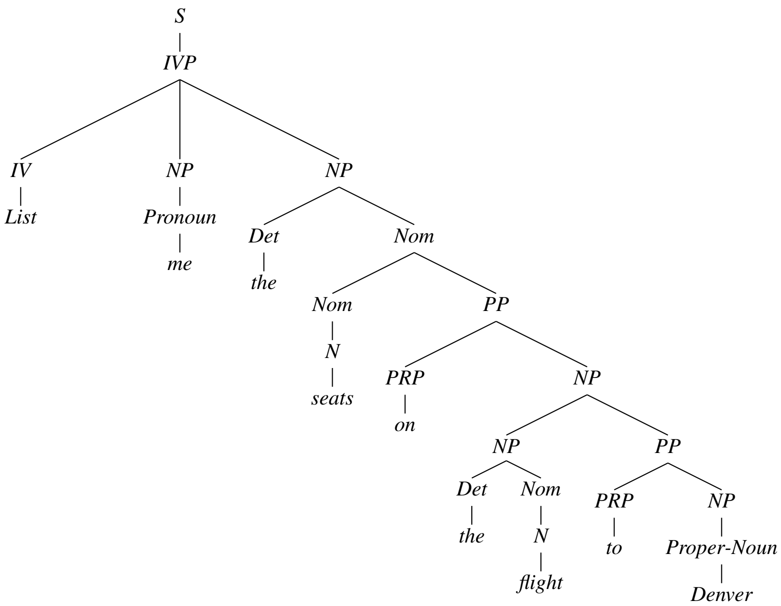
(c) Repeat the above procedure, but this time with the following rule added to your previous stack of rules:

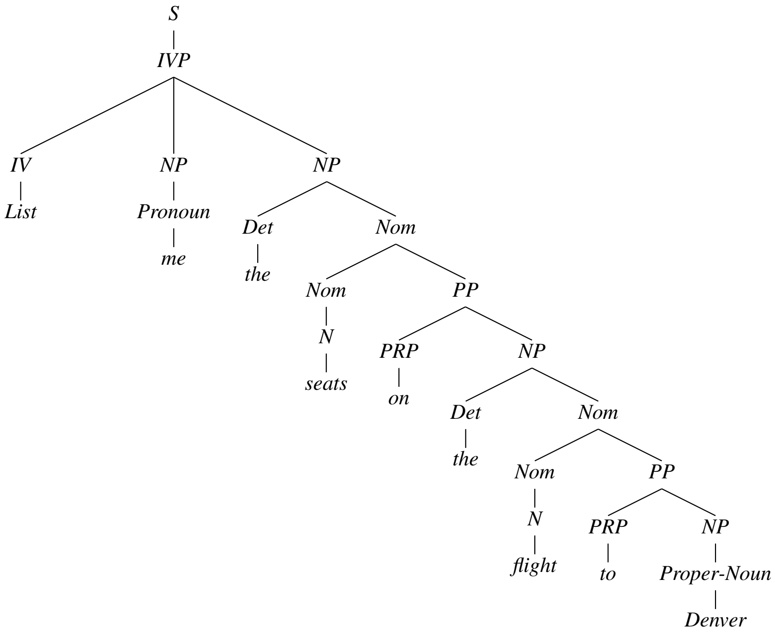
$$NP \rightarrow NP\ PP$$

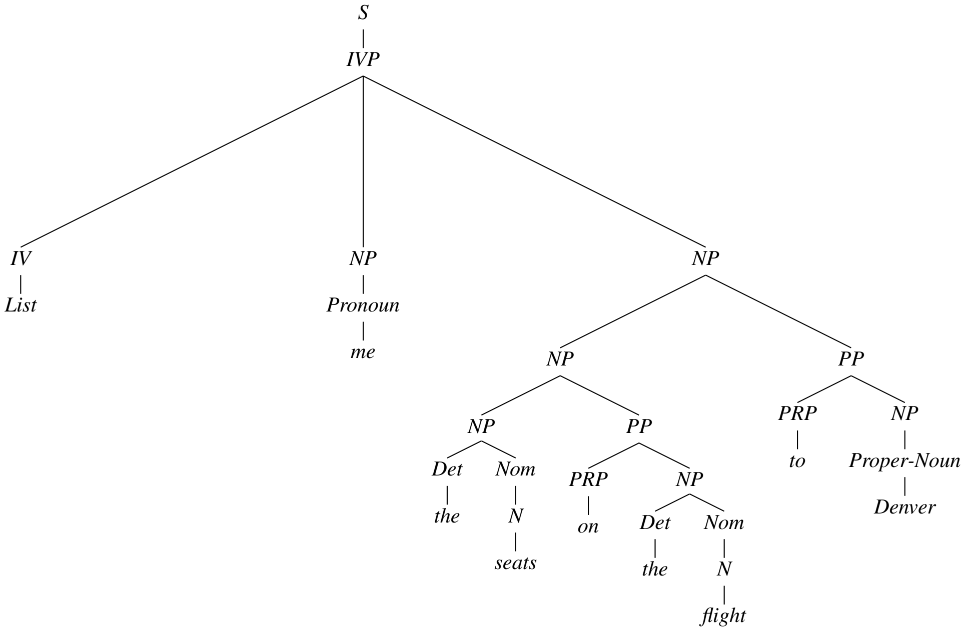
How many new parses do you get this time? (2 points)

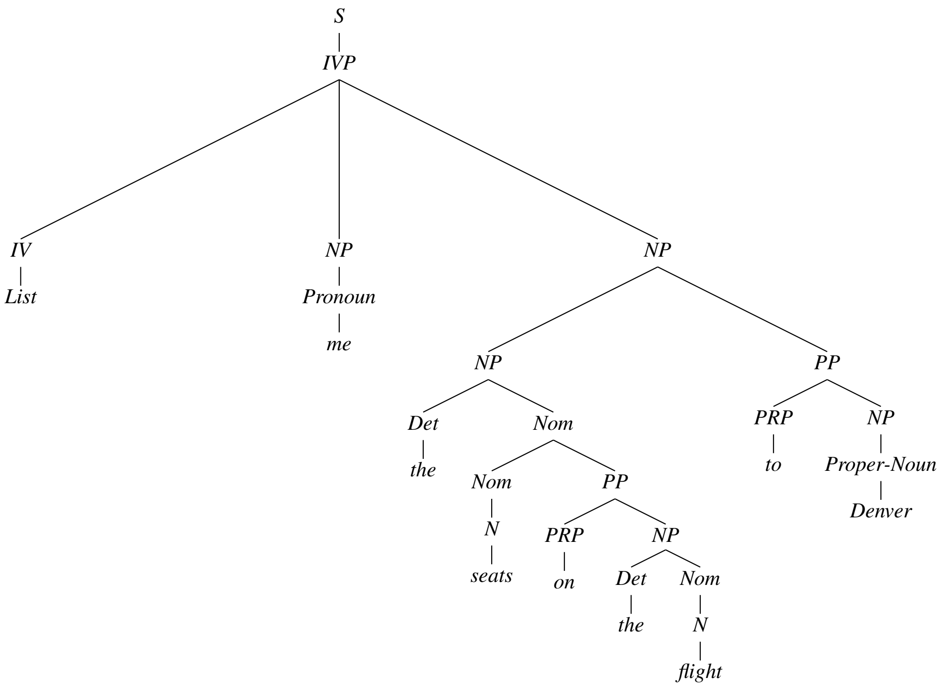
Solution: please see `grammar_sol.py`. *You should get 11 parses this time:*

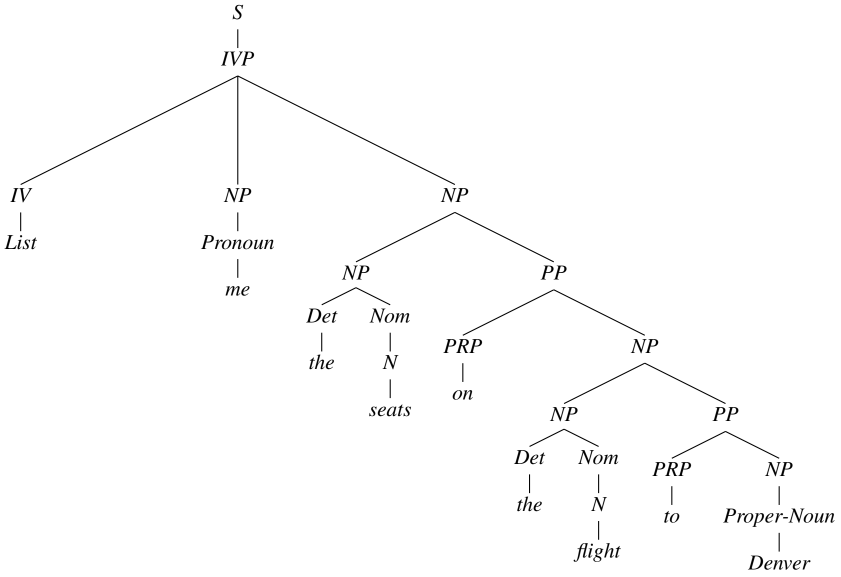


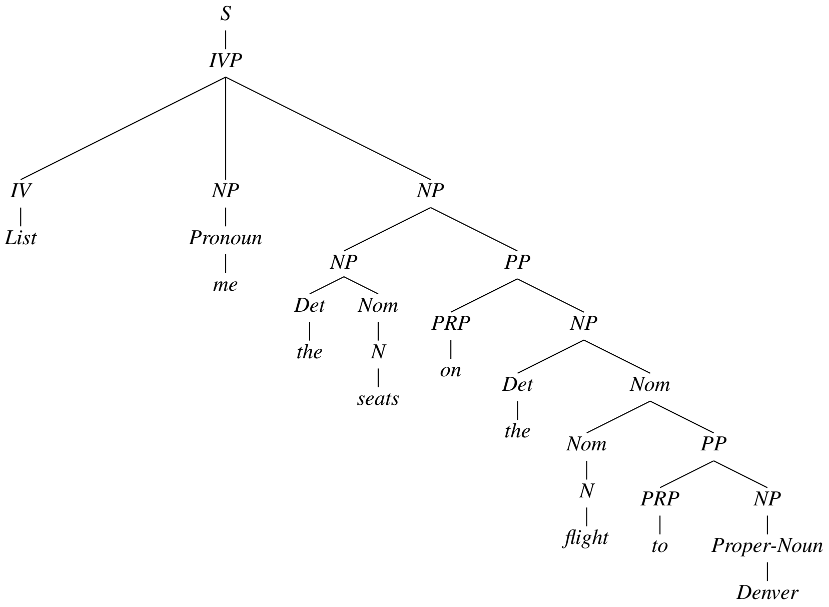

S
IVP
IV
List
NP
Pronoun
me
NP
Det
the
Nom
Nom
N
seats
PP
PRP
on
NP
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver

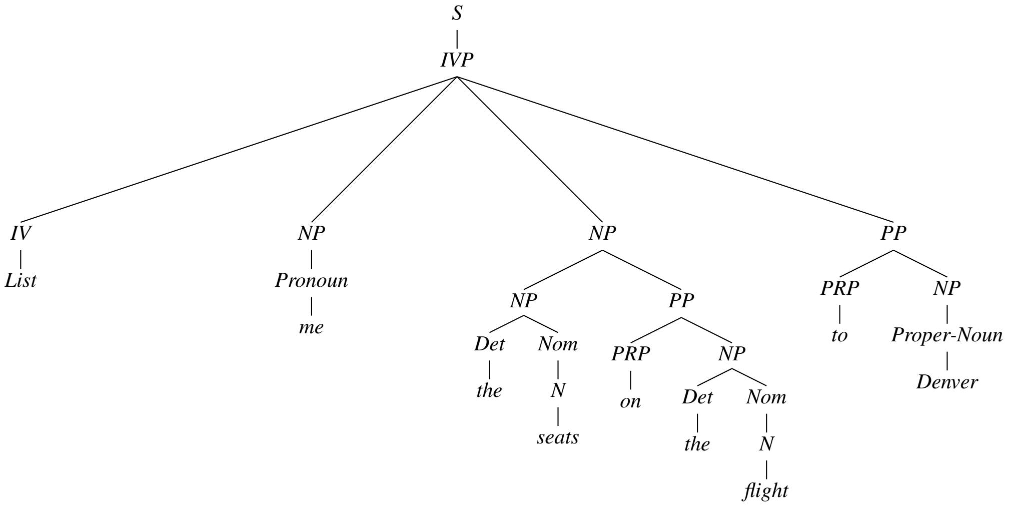

S
IVP
IV
List
NP
Pronoun
me
NP
Det
the
Nom
Nom
N
seats
PP
PRP
on
NP
Det
the
Nom
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver

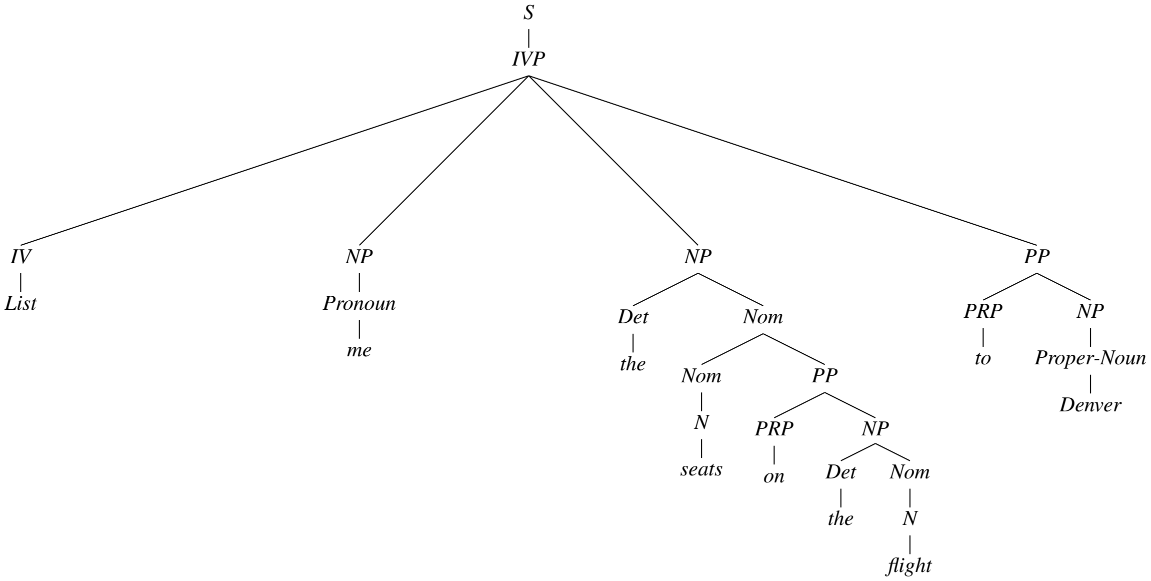

S
IVP
IV
List
NP
Pronoun
me
NP
NP
NP
Det
the
Nom
N
seats
PP
PRP
on
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver

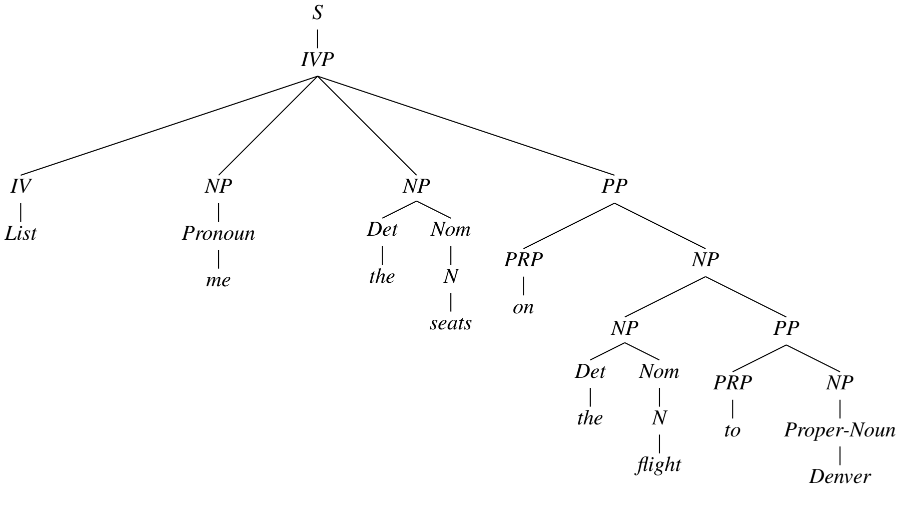

S
IVP
IV
List
NP
Pronoun
me
NP
NP
Det
the
Nom
Nom
N
seats
PP
PRP
on
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver

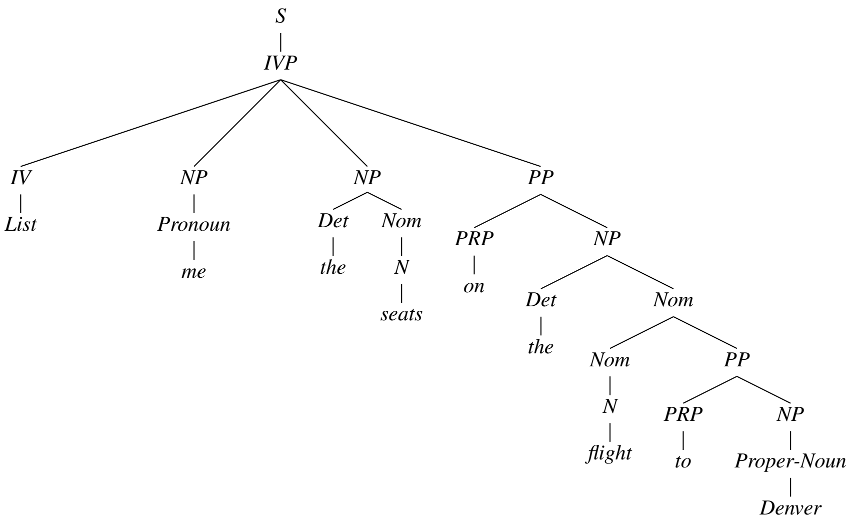

S
IVP
IV
List
NP
Pronoun
me
NP
NP
Det
the
Nom
N
seats
PP
PRP
on
NP
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver


S
IVP
IV
List
NP
Pronoun
me
NP
NP
Det
the
Nom
N
seats
PP
PRP
on
NP
Det
the
Nom
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver


S
IVP
IV
List
NP
Pronoun
me
NP
NP
Det
the
Nom
N
seats
PP
PRP
on
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver


S
IVP
IV
List
NP
Pronoun
me
NP
Det
the
Nom
Nom
N
seats
PP
PRP
on
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver


S
IVP
IV
List
NP
Pronoun
me
NP
Det
the
Nom
N
seats
PP
PRP
on
NP
NP
Det
the
Nom
N
flight
PP
PRP
to
NP
Proper-Noun
Denver



(d) Did you discover any new meanings from the parses of parts (b) and (c) above? Describe and discuss these new meanings. List any parses that are nonsensical. (4 points)

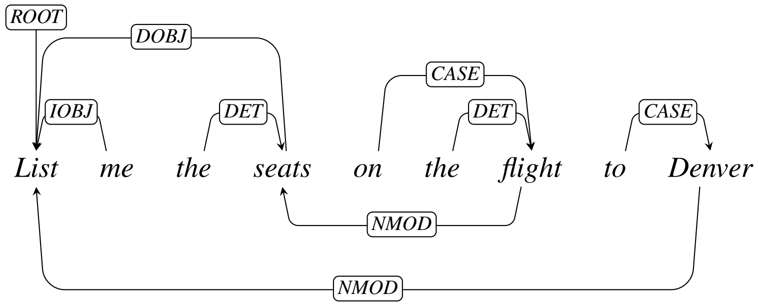
Solution: the new meanings would be cases where "to Denver" modifies the main verb, which would mean the listing happens to Denver, which is nonsensical (but would make sense for a different PP like "whilst driving").

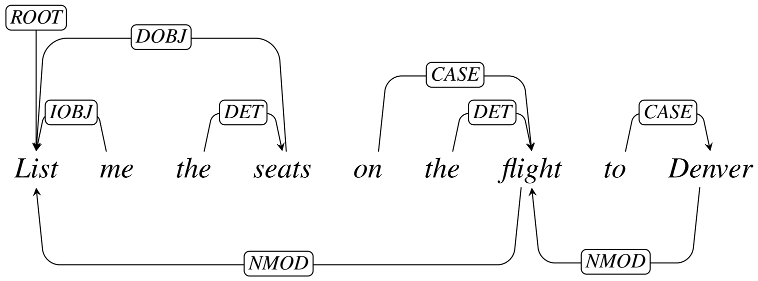
(e) Use the following dependency labels table and without using any NLTK routines, so by hand, transform the CFG parse trees provided by NLTK for part (b) above (so before adding the rule in part (c)) to dependency trees. (4 points per tree, 16 points in total)

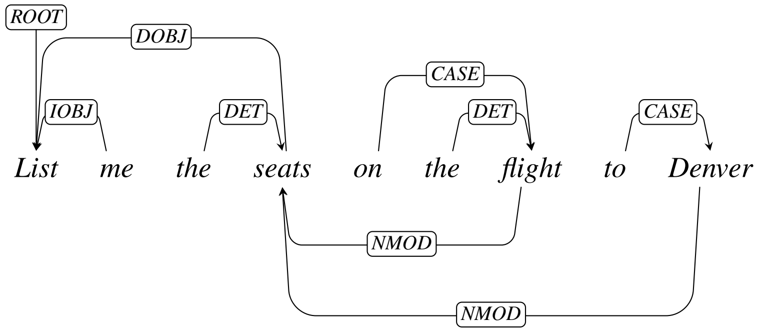
Clausal Argument Relations	**Description**
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier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

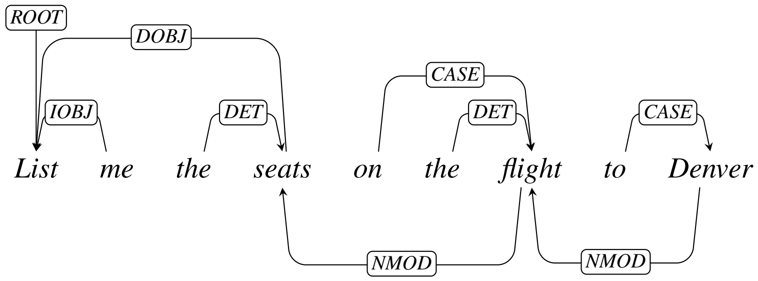
Figure 14.2 Selected dependency relations from the Universal Dependency set. (de Marneffe et al., 2014)

Solution: we get basically the same tree four times, but the modification arrows differ per tree:





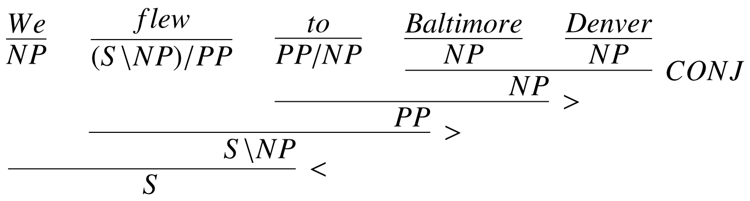




Total: 30 points

5. Using the syntactic categories and the lexicon exemplified in Ch. 11, draw a CCG parse tree for 2.(f). Clearly label your tree with the CCG rules that you are using. (5 points, 2 for using the coordination rule correctly, 1 for each of the other rules, total of 5 points)
Total: 5 points

Solution:



6. Load the parsed version of ATIS corpus into NLTK and extract the CFG grammar used to parse it. The parsed version of ATIS can be found in the `parseTrees.txt`, and some starting code is available in `grammar_prob.py`. You can use an NLTK routine here. (5 points)

   Learn the probabilities of your grammar from the parsed corpus and turn your grammar into a PCFG. This part needs programming, but do not use the `induce_pcfg` NLTK routine. (40 points)

   Now, given your PCFG, compute what is the most likely parse of the ambiguous sentence "Show me the meals on the flight from Phoenix". (8 points)

   Provide the following two items with your answer to this question (5 points):

   (a) your PCFG,

   (b) the probabilities of each of the parse trees of the sentence.

   Total: 58 points.

   Solution: this is a programming question, please see `grammar_prob_solution.py` .