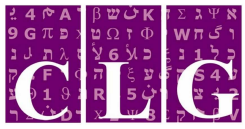

Computational Linguistics Group
Department of Computer Science
University of Haifa

בלשנות חישובית
החוג למדעי המחשב
אוניברסיטת חיפה

Challenge 3: Parsing

1

Implement a simple top-down parser along the lines of the `Parse` procedure given in class. Add control printouts whenever `Parse` is called and whenever the procedure returns.

2

Implement a bottom-up parser which realizes the simple bottom-up parsing schema given in class. Again, add printouts whenever an item is created. Provide a documented listing of the algorithm.

3

Run the two parsers on the following grammar:

$S \rightarrow NP\ VP$	$Det \rightarrow that \mid this \mid a \mid the$
$S \rightarrow Aux\ NP\ VP$	$Noun \rightarrow book \mid flight \mid meal$
$S \rightarrow VP$	$Verb \rightarrow book \mid include \mid includes$
$VP \rightarrow Verb$	$Prep \rightarrow from \mid to \mid on$
$VP \rightarrow Verb\ NP$	$Proper\text{-}Noun \rightarrow JFK \mid LA \mid TWA$
$NP \rightarrow Det\ Nominal$	$Aux \rightarrow does$
$NP \rightarrow Proper\text{-}Noun$	
$Nominal \rightarrow Noun$	
$Nominal \rightarrow Noun\ Nominal$	
$Nominal \rightarrow Nominal\ PP$	
$PP \rightarrow Prep\ NP$	

Try to parse the following strings:

this flight includes a meal

the flight from LA includes a meal

does the flight include a meal

book that flight

the flight from JFK to LA on TWA includes a meal

the flight does include a meal

the flight does not include a meal

the flight

Based on these experiments, compare the two parsers in terms of completeness, efficiency, reduplication of effort etc.

4

Propose simple modifications to each of the parsers which would guarantee termination. You do not have to implement anything here.

5

Propose an extension to each of the parsers which would result in printing one tree induced by the grammar on the input sentence. There is no need to account for all possible trees. You do not have to implement anything here.