

Assignment 6

L545

Due Wednesday, March 27

1. Using the grammar in figure 13.1 on page 428, provide the following:
 - (a) A top-down, depth-first search for *I prefer a meal*.
 - (b) A top-down, breadth-first search for *I prefer a meal*.
 - (c) A left-corner search for *I prefer a meal*.

The bottom-up search is given for *Book that flight* in figure 13.4 on page 431, which might help you in your answer.

2. The CYK algorithm can only be used with rules in Chomsky Normal Form (CNF)—i.e., where rules are only of the form $X \rightarrow A B$ or $X \rightarrow w$ ($A, B, X \in N$ and $w \in \Sigma$). Explain where exactly the CYK algorithm would change and how it would be more inefficient if arbitrary CFG rules were allowed.
3. Show the (passive) chart for *I prefer a flight to Houston*, using the CNF grammar in Figure 13.8 on p. 438—be sure to note that the original lexical entries also carry over from the previous grammar (Figure 13.1, p. 428), as stated in the caption. (Figure 13.9 might be of help.)
4. Explain how Earley parsing avoids the left recursion problems which plagued top-down parsing.
5. This page provides a way to implement a **reverse** function in Prolog:

http://www.cse.psu.edu/~catuscia/teaching/cg428/exercises/Prolog_solutions.html

It gives two solutions:

```
% Naive, inefficient (quadratic) solution:
naive_reverse([], []).
naive_reverse([X|L],K) :- naive_reverse(L,M), append(M,[X],K).

% Fast (linear), tail-recursive solution:
fast_reverse(L,K) :- rev_aux(L,K, []).
rev_aux([],K,K).
rev_aux([X|L],K,M) :- rev_aux(L,K,[X|M]).
```

Explain in prose how the `fast_reverse` function is able to efficiently reverse a list.