HW 2
Due Wed. 22^{nd} Nov. (before mid-night)
Total points: 10

1. Consider the following context free grammar with start symbol S. Terminal symbols are indicated in *italic* font. Convert this grammar to Chomsky Normal Form. 1 points

S → NP VP	PP → Pre NP
S → *I* VP PP	V → *ate*
NP → Det N	Det → *the* | *a*
VP → *ate* NP	N → *fork* | *salad*
VP → V	Pre → *with*

2. Consider the PCFG below:

S	→	Subj VP (1.0)
VP	→	V Obj (0.5) | V Obj Obj (0.3) | V Small (0.2)
Small	→	Obj V (1.0)
Subj	→	*I* (0.3) | NP (0.7)
Obj	→	*her* (0.2) | NP (0.8)
NP	→	N (0.5) | Det N (0.5)
V	→	make (0.6) | duck (0.4)
N	→	duck (0.5) | goose (0.5)
Det	→	her (1.0)

Use a probabilistic CYK-style algorithm to find the most probable (Viterbi) parse for the sentence
I make her duck

Note: For this example, do **not** convert the grammar to Chomsky Normal Form. Instead, you should draw up a CYK-style parse chart for the grammar as it stands. For small examples this is perfectly feasible, but you should check that you understand why a CYK- style algorithm has difficulties with non-CNF grammars in general.

Hint: The sentence has three readings or meanings; ‘I make a duck for her’ , ‘I make a duck that is her’s ’, and finally, one in which *duck* is a verb and *her duck* forms a “small clause”, corresponding to a reading in which ‘she’ is ‘ducking’.

(a) For cells marked **A** and **B**, fill in categories and their probability (upto 3 decimal places), as shown for the cell *her* below. For your own purpose, you will have to fill in the entire chart, but do not show this in your answer. 2 points

I			**A**
	make		**B**
		Det 1.0 Obj 0.2 *her*	
			duck

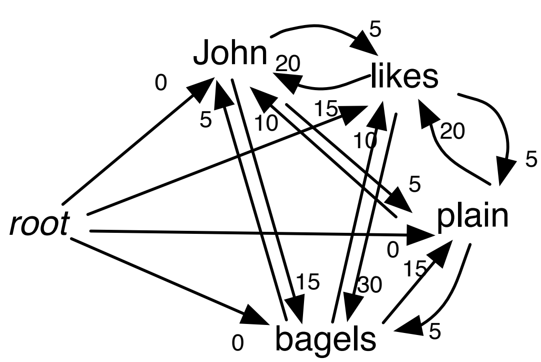
(b) Draw the most probable parse (either as a tree or in bracket notation) 1 points

3. Dependency parsing / MST

Imagine that you are using an arc-factored non-projective dependency parsing model. You are given a sentence: *John likes plain bagels.* In this exercise, we consider that there are no labels on arcs. So, the goal is to choose a dependency structure which maximises the sum of arc scores:

$$G = \arg \max_{G \in T(G_x)} \sum_{(i,j) \in G} w_{ij} \tag{1}$$

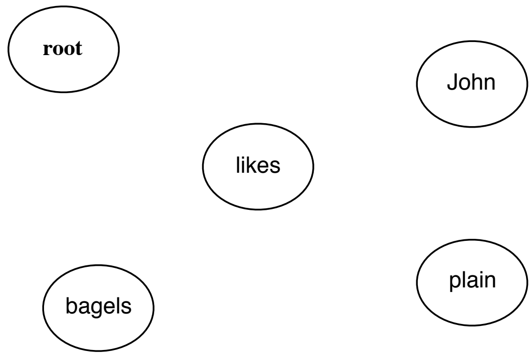
The scores for each directed arc are given in the following graph (the table is the corresponding matrix):



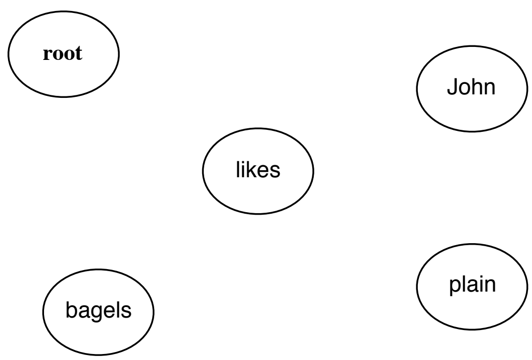
dependends \ *heads*	root	John	likes	plain	bagels
John	0	-	20	10	5
likes	15	5	-	20	10
plain	0	5	5	-	15
bagels	0	15	30	5	-

Apply the Chu-Liu-Edmonds (CLE) algorithm to the above example in order to find the MST.

(a) Apply the first step of the algorithm and indicate any cycles. 1 points

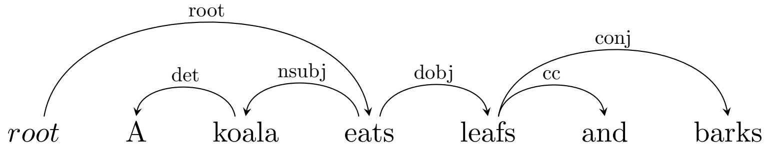


(b) Show the final result (i.e. the final graph with scores). **Do not show intermediate steps** 2 points



4. Dependency parsing / Transition-based

Consider a sentence "A koala eats leafs and barks". Assume that a correct dependency tree for this sentence is the following:



(a) Show the configurations that an arc-standard transition based dependency parser will use in order to parse this sentence. Use the following table to show the action, stack, buffer and arcs, as in the slides. Will a transition-based dependency parser using the arc-standard system be able to correctly predict the structure shown above? 3 points

Transition	Stack	Buffer	Arcs