START OF QUIZ
Student ID:
38419826,Zeng,Lingsong

Question 1

Topic: Lecture 3
Source: Lecture 3

Explain why the following rule is not valid in a CFG: NP VB -> DT NN VB (1)

Question 2

Write the parenthetic parse of the following sentence: "Yertle the Turtle is king of the pond."
(1)

Question 3

Topic: Lecture 4
Source: Lecture 4

We didn't talk about it in class, but how do you think a parenthetic clause (where an explanatory or tangential clause is inserted into another) might be accounted for in a CFG or feature grammar? You can assume that it works similarly for all different types of phrases. (1)

Question 4

Topic: Lecture 4
Source: Lecture 4

Name 2 advantages of feature grammars over CFGs, and briefly explain why they are useful.
(1)

Question 5

Topic: Lecture 2
Source: Lecture 2

Why do we not use accuracy to evaluate chunkers? Can you think of any other tasks where this might be as big (or bigger) of a problem? (1)

Question 6

Topic: Lecture 1
Source: Lecture 1

Imagine someone develops a new parser that has 100

Question 7

Post-positive adjectives are adjectives that occur after the noun phrase they are modifying (such as "attorney/surgeon general", "somewhere nice", "nothing important"). Given that they tend to occur in set phrases, do you think it would be better to write a general class of PostAdj, and create PostAdj phrases in a CFG, or just list them as valid NPs (ie, NP = surgeon general)? Discuss the pros and cons of either decision. (2)

Question 8

Topic: Lecture 2
Source: Lecture 2

Imagine you're working on analysing customer feedback, and your boss wants you to identify the most common complaints. How might you use your parsing knowledge to automate and distill the most common complaints? You can assume that complaints have already been labeled with the product they are complaining about. You can also assume that just sorting the frequency of tokens is going to be insufficient. (2)

Question 9

Topic: Long
Source: Lecture 1

Imagine you're a text-to-speech (TTS) engineer. You've been asked by your boss to make your system sound more authentic by incorporating intonation into your model. Intonation is a pitch and stress pattern that differs between different pragmatic conditions. For example, English yes-no questions have a rising pitch on the end of the clause, imperative statements (ie, commands) have a falling pitch, and declarative sentences, while also falling, are not typically as sharp a fall as imperative sentences. How might you use this information, along with a parser, to modify your TTS system? Are there any complications or ambiguities that you can think of? (3)

END OF QUIZ