START OF QUIZ
Student ID:
97713317,Song,Shawn

Question 1

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 2

Topic: Lecture 2
Source: Lecture 2

Conceptually, obliques and nmods are very similar. How do they differ, and what does this actually mean, from a linguistic perspective (ie, when would we use one over the other)? (1)

Question 3

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

Question 4

Topic: Lecture 3
Source: Lecture 3

Explain how phrasal attachment errors produce ambiguity. (1)

Question 5

Topic: Lecture 4
Source: Lecture 4

Briefly describe how underspecification works in a feature grammar. (1)

Question 6

Basque is an "ergative-absolutive" language - instead of defining NPs with respect to labels such as "subject" and "direct object", NPs are defined with respect to "subject of a transitive verb" (ergative) or "subject of an intransitive verb OR object of a transitive verb" (absolutive). Explain what features would need to be defined in such a grammar, and how they would interact (you can assume a similar SVO order as English). (2)

Question 7

Imagine, if you will, a "mildly-context-sensitive" grammar, that only allows for one non-terminal to appear as a contextual marker (let's call it "CON"). Anything not involving CON has to satisfy CFG rules. Do you think that this would be restrictive enough to satisfy the small number of cases that don't satisfy context-freedom, without just being a CSG in disguise? (2)

Question 8

We use trees to represent the structure of a parse, but that doesn't necessarily mean we have to use a Python Tree to represent them. Can you think of an alternative way of representing a syntax tree, preserving the hierarchy and traversal features inherent in a tree (no, you can't just create a "Shrub" class). Write some pseudocode that shows how this structure works. (2)

Question 9

Topic: Long
Source: Lecture 4

Imagine you're building a tool to help second language learners of language X. You have a grammar of their first language (L1), and a grammar of the language they are trying to learn (X). How might you build a tool that learns how to translate a production from L1 into X? Describe any additional data or tools you might need, and the process you would use to learn a "production-translation grammar". Also explain how you could use this to create illustrative examples of how the syntax of language X works.

END OF QUIZ