Computer Science 4140
Final Exam

This exam is take-home. You may consult your handouts, notes, and online class resources. Do not reference any other resources. Do not discuss it with anyone but the instructor. It is due Monday May 1, at 10:30 A.M. Work the problems on your own paper. **Be sure your name is on every page.**

1. Consider the following dialog excerpt from an interaction with the Circuit Fix-It Shop (C denotes computer utterances and U denotes user utterances).

    1. C: What is the voltage between connector 41 and connector 110?
    2. U: I do not know.
    3. C: Locate the voltmeter.
    4. U: Okay.
    5. C: Add a wire between the minus com hole on the voltmeter and connector 41.
    6. U: How do I do that?
    7. C: Connect the end of the black wire with the small plug to the minus com hole on the voltmeter.
    8. U: Okay.
    9. C: Connect the end of the black wire with the large plug to connector 41.
    10. U: Where is connector 41?
    11. C: Near the left boundary and toward the middle is a green region. Now in the middle of this green region is connector 41.
        (user makes all connections and measures voltage)
    12. U: The voltage is zero.

    Draw the proof tree associated with this dialog excerpt. Use a presentation method analogous to the method used in the paper, "Spoken Variable Initiative Dialog: An Adaptable Natural-Language Interface." Be sure to clearly indicate which utterances are attempts to get the user to supply a missing axiom and which utterances actually satisfy missing axioms as done in the paper (15 points).

2. In the CPOF system design, what mechanisms are used to enable a module to focus only on what information it needs without having to specify where the information resides (5 points)?

3. For DSTC 2, list at least three different types of information available from the logged dialog data that are available for use by a dialog state tracker (5 points).

Note: Questions 4 and 5 require you to specify Python code. You should submit your code electronically in a file named *examcode.py* using a submission ID of *final*. In addition, you should write your answers on paper and submit with the rest of the take home exam.

4. Define a function *supergloss(s)* that takes a synset *s* as its argument and returns a string consisting of the concatenation of the definition of *s*, and the definitions of all the hypernyms and hyponyms of *s* (15 points). When you produce your concatenation, be sure to include the newline character at the end of each definition so that when the return value is printed, the results will look like the following.

```
>>> print(supergloss('tree.v.03'))
chase an animal up a tree
HYPONYMS:
HYPERNYMS:
go after with the intent to catch

>>> print(supergloss('play.n.03'))
a preset plan of action in team sports
HYPONYMS:
a play executed by a basketball team
(American football) a play by the offensive team
(rugby) knocking the ball forward while trying to catch it (a foul)
(ice hockey) a play in which one team has a numerical advantage over the other as a result of penalties
any exciting and complex play intended to confuse (dazzle) the opponent
HYPERNYMS:
a plan for actively doing something
```

5. **(Corrected: Tuesday April 24 at 8:15 P.M.)** Given the execution of the following code:

```
wsj = nltk.corpus.treebank.tagged_words()
cfd2 = nltk.ConditionalFreqDist((tag, word) for (word, tag) in wsj)
targets = list(cfd2['VBN'])
```

write a piece of code that produces the list of all the word-tag pairs that immediately precede items from *targets*. In addition, write down the size of the list that your code produces as well as the size of the set that it produces. In other words, if you store the result in a variable named *ans*, please submit the value you get for *list(ans)* and *list(set(ans))*. In your electronic submission, be sure to include the three lines provided above (10 points).