**CS 171, Winter, 2021  
Artificial Intelligence  
Project, part 1**

Below is part 1 of the quarter-long chatbot project. It is due around the end of the third week of the quarter; precise date and time to be determined.

You must work on the project individually. You can talk with other people in the class about your and their general approaches, but do not copy any code or algorithm details.

We will supply some starter code in Python, which is a very simple weather chatbot, but you can do your own coding in another programming language. Java and C++ are fine. If you want to use another language, get approval first from Marzieh Ashrafiamiri, *mashrafi at uci dot edu*. Here are links to the three starter Python files (as .txt files):

* [Proj1.py](https://frost.ics.uci.edu/cs171W21/Proj1.py.txt)
* [CYKParse.py](https://frost.ics.uci.edu/cs171W21/CYKParse.py.txt)
* [Tree.py](https://frost.ics.uci.edu/cs171W21/Tree.py.txt)

Text-based chatbots typically have several components:

1. Code to convert the input string into a parse tree. (CYKParse, getSentenceParse)
2. Code to convert a parse tree into some convenient internal representation. (updateRequestInfo)
3. Code to retrieve information from a database, using the convenient internal representation. (getTemperature)
4. Code to format a reply to the user, based on the retrieved information. (reply)

You'll notice that in the starter code most of these components are implemented in a rudimentary way. As you proceed through the project, you'll need to design and code more robust and interesting replacements for the code supplied here.

Your assignment for part 1 of the project is to complete the following steps. You should get started soon, although it is likely that some details of the steps may change over the next week.

**Steps**

1. Become familiar with the supplied code, and translate it into another programming language if you choose. It would be good to run CYKParse as the main module, comment out and un-comment out various lines in the main() function, and carefully look at the output. Nothing to turn in for this step.
2. The CYK-Parse in the textbook (Fig. 23.5) and in the code require that the syntax be in Chomsky Normal Form. Sometimes this is inconvenient, and it would be helpful to also accept syntax rules of the form *X ⟶ Y [p]* — that is, with a single category on the right hand side. For instance, one of the rules in Fig. 23.4 has such a form. Let's call the modified CNF, relaxed to also permit single category right hand sides, "Chomsky 171 Form". Your assignment in this step is to modify the CYKParse implementation to accept C1F.
3. Now that your code accepts C1F, take advantage of it. Modify the getGrammarWeather function to use C1F rules, with no change in functionality. This should make the new grammar somewhat more compact.
4. Here's a bold claim: "The probabilities associated with CNF and C1F lexicons, as seen in Fig. 23.3, are completely unnecessary and have no effect on the output of CYK-Parse." Is this statement true or false? Support your position, ideally by reporting on an experiment with your code.
5. Improve and extend the weather-bot in several ways:
   1. Include the cities Tustin and Pasadena, and the time yesterday, in the system's capabilities.
   2. Extend the grammar to understand and respond to questions of the sort "Will tomorrow be hotter than today in Irvine", "Will yesterday be hotter than tomorrow in Pasadena?" (Yes, the verb tense doesn't quite work in that one, but you can ignore that.)