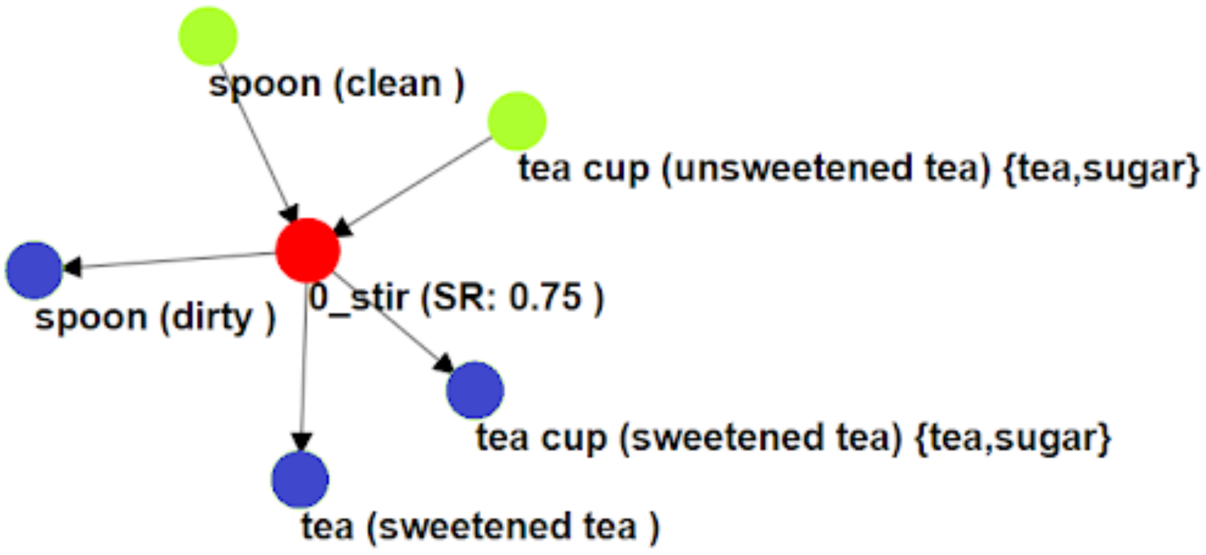
Project 1: Knowledge Retrieval

Part 1

In this project, you will be introduced to a knowledge representation for robots called FOON (Functional Object-Oriented Network). This representation is a graph that contains information about how objects can be used in certain tasks or manipulations to do things. Each action (represented by the connections to a motion object in red) is contained within a structure called a functional unit.

Here is a functional unit for example:


spoon (clean)
tea cup (unsweetened tea) {tea,sugar}
spoon (dirty)
0_stir (SR: 0.75)
tea cup (sweetened tea) {tea,sugar}
tea (sweetened tea)

This unit shows that two items, a clean spoon and a cup of unsweetened tea and sugar, can be used in a stirring action to create a cup of sweetened tea. The textual equivalent for this is the following:

```
O	tea cup	0
S	unsweetened tea		{tea,sugar}
O	spoon	1
S	clean
M	stir	Assumed		Assumed
O	tea	0
S	sweetened tea
O	tea cup	0
S	sweetened tea	{tea,sugar}
O	spoon	1
S	dirty
```

Note the format of the file.

1. Each component is tab-separated (i.e. \t). If the object name has multiple parts, separate it using space.
2. You need to list input objects BEFORE the motion line and output objects AFTER the motion line.
3. The object line is formatted as:

   O\t<NAME OF OBJECT>\t<0 or 1, describing if this object is moving or not>

4. The state line is formatted as:

   S\t<OBJECT'S STATE>\t{LIST OF INGREDIENTS, COMMA SEPARATED}

   The state is identified by the phrase or word within the parentheses. Certain items can be containers. This is why we need to list the items contained within them by identifying them in curly brackets.

5. The motion line is formatted as:

   M\t<NAME OF MOTION>\t<STARTING TIME>\t<ENDING TIME>

   When labelling motions, you need to watch the videos and note the starting and end times if the action is indeed in the video. If the action is not there, then you can just put **Assumed**.

6. After finishing a functional unit, you need to put a line with "\\" in it to separate each functional unit from one another.

   For example:

   O….
   S…. **(FUNCTIONAL UNIT *T*)**
   \\
   O…
   S… **(FUNCTIONAL UNIT *T+1*)**

For this project, you will first have to annotate videos as FOON graphs. You will work individually. We have provided some annotated files that you can use as an example. While you perform the annotation, you can use the following page to visualize the graph. Your graph must be in the correct format, or else, there will not be anything shown. The tool can be found here: http://www.foonets.com/FOON_view/visualizer.html. You need to select the file using the file chooser button, select level 3 and then it will display the graph.
Your Tasks:

1. Understand FOON graphs:

- check the sample annotations from this link: https://github.com/sadman3/task-tree-generation/tree/master/subgraphs/TXT
- You will find the video link at the top of each .txt file.
- Visualize the graphs using the interface provided in http://www.foonets.com/FOON_view_v2/visualizer.html
  a. In this interface, object nodes are denoted in a lime green color, while motion nodes are denoted in a red color.
  b. You can drag nodes around to make them easier to view, or you can zoom in/out to visualize the graphs better.
- Read the
  https://arxiv.org/pdf/1902.01537.pdf
  https://arxiv.org/pdf/2207.03693.pdf

2. Go through the instruction video project1.mp4. Write a simple function that

- can load the FOON file
- can find a goal node and output the functional units that have the goal node
- Find all nodes in the list of ingredients and utensils available in the kitchen from a text file, if they are available and output their functional units

- Your program should have the following input:

  1. FOON (A .txt file)
  2. A goal object to search (An object name and its state)
  3. List of ingredients and utensils available in the kitchen (A .txt file)

FOON, sample goal objects and a kitchen file will be found in the provided starter code.

Submission:

You should use Python for this project. Create a zip file including everything that are required to run your program. That is,

- FOON (.txt file)
- Your code
- A test script that can demonstrate the function in your task 2.
- A readme file with the instructions about how to run your program.

Name the zip file with your name (e.g. james.zip). Submit it on Canvas.

This first part is worth 20% of your the project's grade.