Project 3: Task Tree Generation

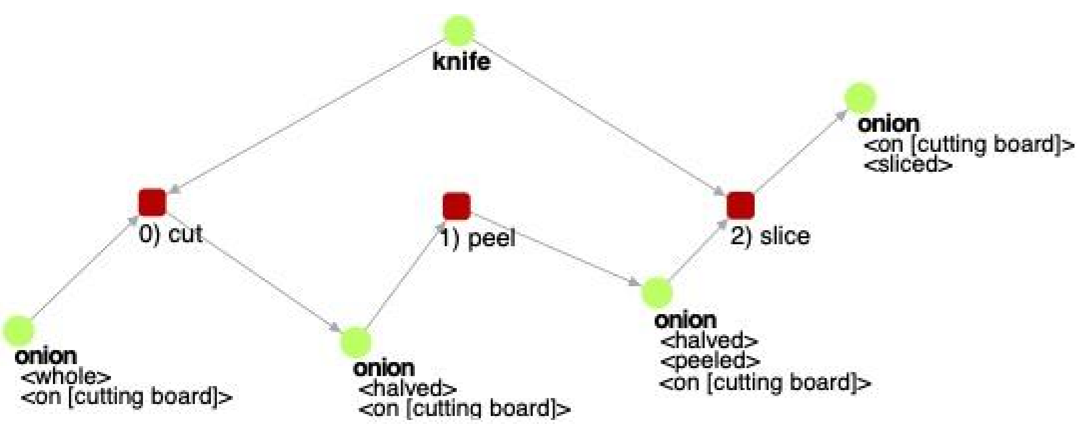
Often, natural language instructions are challenging for robots to grasp due to their unstructured nature. Thus, creating a robust knowledge representation for robots that can make sense of these unorganized instructions is crucial. This project targets the generation of this knowledge representation, more specifically, creating a 'task tree'.

A 'task tree' stands as an ordered sequence of actions that a robot is required to carry out to achieve an intended outcome, providing a systematic breakdown of information. Each step enclosed within this task tree is depicted by what we classify as a 'functional unit'.

A 'functional unit' is made up of three fundamental elements -

- **Input Nodes:** These define the initial state of the objects involved in the action (e.g., whole potato vs. pre-sliced potato).
- **Output Nodes:** These represent the anticipated state of the objects after the action (e.g., sliced potatoes).
- **Motion Node:** This specifies the specific physical movement the robot needs to perform (e.g., grasp the potato).

The following is an example of a task tree of 'slicing a whole onion':



Part 1: Prompt Engineering with Gemini

Here, you'll be working with a large language model called Gemini to design effective prompts. These prompts will guide Gemini towards generating precise task trees, outlining the steps needed to cook specific dishes.

Here is the starter code discussed in the class:
https://github.com/sadman3/project3-starter-code.git

Input:

- Dish name and preferred ingredients
- Available ingredients and utensils in the kitchen (including their states - e.g., chopped onions, clean pot)

The cafeteria menu assigned to you will be found here:
https://docs.google.com/spreadsheets/d/15JxTDUJfP4MGCCYqbBJbU2O7OxMtpEocRnlx-CONh0g/edit?usp=sharing

Task:
Your challenge is to design at least three different prompting approaches that incorporate several examples of task tree generation for cooking meals. This will equip Gemini with the necessary knowledge to create accurate task trees for any given dish. You should compare those approaches, figure out their pros and cons, and report their performances.

Output:
Task trees for each dish in the cafeteria menu.

What to submit:

1. Source code written in python. The code will read from a input file and output each task tree in separate json files. You need to follow the specific input and output format provided in the starter code.
2. The input file. This input file has to be created from the cafeteria menu following the specific format provided in the starter code.

3. A short report on performances. You only need to submit the code of the best prompting approach. But you should also submit a PDF file describing your prompting approaches and analyzing their performances of all your approaches and what leads to them.

Part 2: Peer Review

Every participant is required to evaluate two other participants' Part 1 submissions during this phase. You will execute the Python script and produce task trees for the dishes listed in the input file as part of your evaluation. Subsequently, each generated task tree should be evaluated and accuracy estimation computed. The accuracy can be deduced as follows:
(Correctly generated task trees/ Total task trees)

Part 3: Writing a Report

Based on your evaluation experience, you should revisit your solutions and perform more analysis. You should compare all the approaches including your own prompting approaches and the ones you have reviewed. Then, write a project paper on your work in the previous two phases of this project using IEEE conference template. The templates can be found at:
https://www.ieee.org/conferences/publishing/templates.html

You should have the following sections in your paper (more detail will be provided later):

- Introduction
- Related work
- Methodology
- Experiments/Discussion
- References

For your reference, here are some related works you should read:
LLM with FOON:

- Sakib, Md Sadman, and Yu Sun. "Consolidating Trees of Robotic Plans Generated Using Large Language Models to Improve Reliability."

International Journal of Artificial Intelligence and Robotics Research (2024): 2450002. (https://www.worldscientific.com/doi/abs/10.1142/S2972335324500029 or https://arxiv.org/abs/2401.07868

- Sakib, Md Sadman, and Yu Sun. "From Cooking Recipes to Robot Task Trees--Improving Planning Correctness and Task Efficiency by Leveraging LLMs with a Knowledge Network." arXiv preprint arXiv:2309.09181 (2023). (https://arxiv.org/abs/2309.09181)

Language model with FOON:

- M. S. Sakib, D. Paulius and Y. Sun, "Approximate Task Tree Retrieval in a Knowledge Network for Robotic Cooking," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 11492-11499, Oct. 2022, (https://ieeexplore.ieee.org/abstract/document/9830875)

Background of FOON:

- Paulius, David, Yongqiang Huang, Roger Milton, William D. Buchanan, Jeanine Sam, and Yu Sun. "Functional object-oriented network for manipulation learning." In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2655-2662. IEEE, 2016. (https://ieeexplore.ieee.org/abstract/document/7759413, https://arxiv.org/abs/1902.01537)
- Paulius, David, Ahmad B. Jelodar, and Yu Sun. "Functional object-oriented network: Construction & expansion." In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 5935-5941. IEEE, 2018. (https://ieeexplore.ieee.org/abstract/document/8460200, https://www.kavrakilab.org/2017-rss-workshop/paulius.pdf)
- Paulius, David, Kelvin Sheng Pei Dong, and Yu Sun. "Task planning with a weighted functional object-oriented network." In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 3904-3910. IEEE, 2021. (https://ieeexplore.ieee.org/abstract/document/9561680, https://arxiv.org/abs/1905.00502)
- Sakib, Md Sadman, Hailey Baez, David Paulius, and Yu Sun. "Evaluating recipes generated from functional object-oriented network." arXiv preprint arXiv:2106.00728 (2021). (https://arxiv.org/abs/2106.00728)

- Sun, Yu, Shaogang Ren, and Yun Lin. "Object–object interaction affordance learning." Robotics and Autonomous Systems 62, no. 4 (2014): 487-496. (https://www.sciencedirect.com/science/article/pii/S0921889013002339)

Submission:
Submit a pdf file on Canvas.

Bonus Points:
If you submit your paper to arXiv and it gets accepted, email me (mdsadman@usf.edu)
the link of your paper to get 10 bonus points (10% of this project grade).
Here is a tutorial on how to submit a paper to arXiv:
https://www.youtube.com/watch?v=0i4C8yxbs48