Robotics Project Report

Astrid Gamoneda

***Abstract*—The aim of this project is to structure knowledge representation for robot learning and manipulation using directed graph with object and motions where the edges denote a sequencing of actions called foon (Functional Object-Oriented Network).**

I. Introduction

THIS project has several the following parts: Video annotation, Merging results and Searching task trees. This is all done to try to have a working foon system. The goals is to try to achieve a universal FOON. Stracting steps from cooking videos to try to bring this knowledge so the robots can achieve this movements. Cooking is the focus of this project but this same technique would work with any other theme as long as there are objects moving and interacting with each other. The result that this project produces is a file that then can be parse through a program that visualizes it. [1].

II. Experiment - Part 1

The aim of this step is to parse through a video and create a text file with the specific format that can be used to generate Functional Units that together then form a FOON. FOON (Functional Object-Oriented Network) is a graph that contains information about how objects can be used in certain tasks or manipulations to do things. Each action is contained within a structure called a functional unit. They consist of objects states and movements. Every object is followed by a state and every functional unit has only one motion. The format of the file we are trying to archive is like the one shown in figure 1.

III. Experiment - Part 2

For this part a pseudo code was given and the goal was to merge as many FOONs as possible to obtain the universal FOON everytime a video to be parsed is added. The universal FOON is initially empty and everytime is merged with another FOON every functional unit is checked to find out if it is part of the Universal Foon. If it is not we add it to the Universal FOON and if it is not there is no need to add it just reference it. To do all of this there is a need to implement how to compare FU(functional units). The approach taken is to compare the Strings received in the toString() method overridden to return a relevant and unique piece of information for every FU. As Java is a object oriented programming language a class for FU was created while the universal FOON was an ArrayList of FU. The Functional Units were created parsing the file and creating new objects for every delimeter (//). The result that was obtain using this method can be represented by the following graph shown in figure 2 which represents the Universal Foon.

```
O238  tea cup          0
S83   unsweetened tea       {tea,sugar}
O22   spoon 1
S7    clean
M13   stir     Assumed       Assumed
O62   tea      0
S73   sweetened tea
O238  tea cup          0
S73   sweetened tea       {tea,sugar}
O22   spoon 1
S15   dirty
```

Fig. 1. FOON representation as a text file

IV. Experiment - Part 3

The purpose of this step is to search a task tree given a set of goals in a kitchen. The goals file consist in a series of object and states that we want to find in a given tree. The kitchen is a set of objects and the states they are in which are available in to use. The aim of this step is to find out if we can reach the goal states with the ingredients (object + state) that we have in our kitchen looking in the Universal FOON for them. This is done by checking if any of the ingredients we have in our kitchen are part from the input states in our Universal FOON if there are we also add the output states to our kitchen since if we have the input ingredients we can easily create the outputs. For the algorithm to load fully it needs to loop around this process looking for output ingredients we can generate with the ones in our kitchen. This process can finish if we have accomplished all goals but sometimes it can create an infinite loop the way to avoid this is to establish a limit and once this limit is achieved it should halt the execution. Sometimes there can be no solution at all that is why adding this limitation is important now in the case that there is no solution the program will alway finish. The output of this step is a tree of FU which represent the FU needed to accomplish the goals with the ingredients available in the kitchen.

V. Discussion and Results

In the part 3 of the project the goals and kitchen files at first only produced one FU in the solution tree after some debugging it was made clear that the limit established was too low and directly affected the outcome just duplicating the limit the number of outputs increased from 1 to 20 outputs. This

Fig. 2. FOON representation as state graph -1

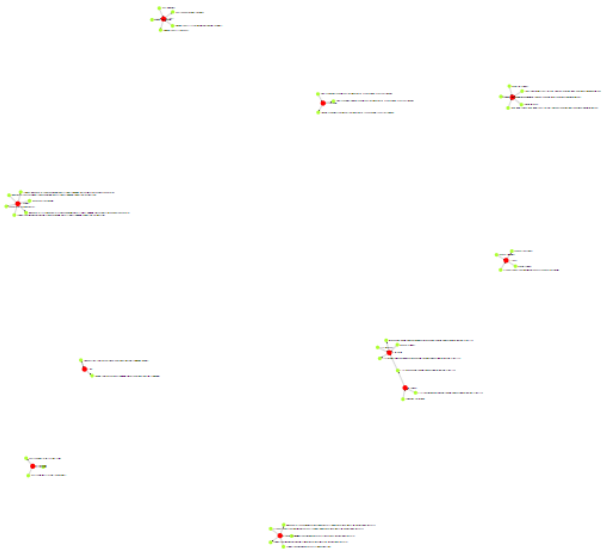

Fig. 3. FOON representation as state graph -2

makes sense because it takes a long time to search the entire tree for every possible output that can generate and from them the output that can generate and that drastically increases the complexity. The FU obtained are represented in the figure 3.

VI. Conclusion

In this report how to parse FU from cooking videos and then use them to add them to a universal FOON through a algorithm developed in java. Once a universal FOON was available we can search for FU having a goals that are available in a kitchen.

Acknowledgments

I would like to thank my professor Dr. Sun and teaching assistant David Paulius for their help and knowledge.

References

[1] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2655–2662.