CSE594 Project Report: PDG Visual Tool

Tatheer Zahra
Advisor: Dr. Gary Tan

April 27, 2023

1 Problem Statement

Program Dependence Graph (PDG) is used to analyze and evaluate data flow in a program. However, these graphs tend to be huge, which makes it difficult to understand and evaluate the graphs. Therefore, having a visual tool that lets us see the graph efficiently is essential.

2 Objective

Modifying the previous PDG Visualization Tool, that takes PDG in a dot file and renders a visual representation from it. Following additions are ample for enhancing the capabilities of the previous PDG Visualization Tool:

- Make the user interface more user-friendly.
- Modernize the outlook of the front-end.
- Emphasizing more on the graph.
- Debug and fix the graph rendering errors.

3 Setup

The tool is a combination of front-end and back-end; therefore, having the environment setup in Anaconda makes environment management easier. Once you have a virtual python environment setup in conda, run the following commands to install dependencies:

- pip install networkx
- pip install Flask
- npm install d3-graphviz
- pip install pydot
- pip install Flask-Cors

Run "python app.py" to fire up the Flask server and follow the prompts on the screen.

4 Results

4.1 Make the user interface user-friendly

- Added a legend/graph key to the front-end, which helps the user discern different types of edges. All three edge types and what they look like is now displayed on the front-end.
- Added boundaries around the legend to add a distinction between edge types.

- Having a Select and Deselect button in the Edge drop-down menu, that saves the user some time in selecting and removing edges.
- Added vertical and horizontal scroll-bars for the div that contains the graph. This lets the graph to be partially viewable in the well, and the scroll-bar could be used to view the rest of the graph.

4.2 Modernize the outlook of the front-end

- Incorporated a new framework, Bootstrap, by re-writing the entire code for the front-end. This allows the website to be mobile friendly and responsive along with having a modern template.
- Using Bootstrap, added a navigation bar that has an Upload and Submit button for submitting a dot file and a Reset button for resetting everything.
- The navigation bar also has a drop-down menu for selecting the types of edges that should be included in the graph.
- Added wells and ViewBox for organizing the divs.
- Made each graph node visibly clickable. Now, each node becomes lighter in color, changes text color and scales a bit up whenever the mouse hovers over the node. The effect is reversed whenever the mouse is out of the node.

4.3 Emphasizing on the graph

- Edited the attributer function in JavaScript so that the graph is populated with a 100 percent height, but has a width equivalent to the inner width of the graph div.
- The graph now covers 75 percent of the screen. Everything else is moved either to the navigation bar or somewhere else on the screen.
- The zooming in and out of the graph is now adjusted according to the view box and where we click in the view box. Graph re-centering is adjusted according to the ratio of the view box and the size of graph.

4.4 Debugging and fixing errors

The previous PDG Visual Tool would end up rendering errors for some dot files. It needed minor fixes in the back-end code to make sure the files run perfectly for every dot file. The error has been corrected.

5 Conclusion and Future Work

Over the course of last few months, I worked on improving the front-end for a better user experience. The front-end was re-written to make it more responsive and have a newer look to it.
Following are some suggestions for future work:

- Add a background color to the instruction nodes. Currently they are white.
- The graph's initial size is dependent on the ratio between the view box, div and the size of the graph without expansion. This could sometimes make the graph extremely small. Using graphviz to adjust the ratio and make the graph look zoomed in, even initially would be a good addition.
- Making the select edge function faster in removing or adding edges to the graph.

6 Output

Figure 1: Front-end after opening the dot file and clicking on Select Edges

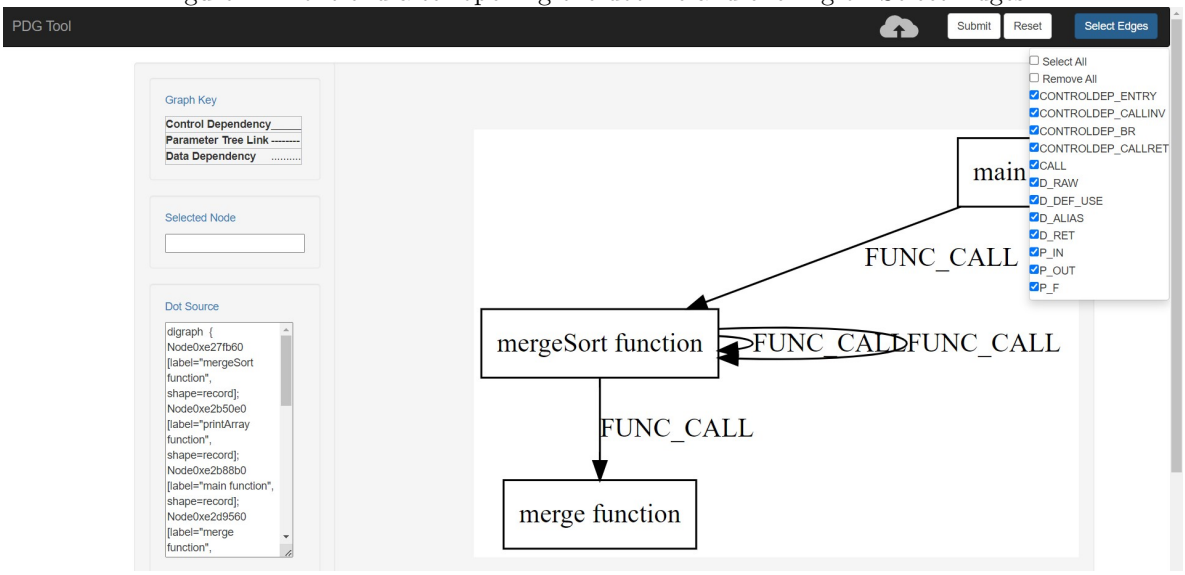

Figure 2: Close-up of the horizontal scroll-bar to view the graph