[keywords: data visualization, program synthesis, human-computer interaction, user interface design]

[an incomplete set of technical skills: familiar with Python, like designing user interfaces and algorithms,]

This summer project is about how to automatically generate data visualizations from high-level specifications. It is somewhat similar to the following papers:

<https://www.cs.utexas.edu/users/isil/popl20.pdf>

<https://arxiv.org/abs/2102.01024>

<https://falx.cs.washington.edu/>

So in this “warm-up” exercise, try your best to learn about Falx, including their papers and the tool. The general format of this exercise is to finish as many tasks as you can and write a short report to demonstrate your understanding. You can choose to work on the tasks in any order you want. In case your time is limited, you should prioritize quality over quantity. That is, feel free to finish a subset of these tasks and write a report summarizing your findings with high quality.

Task 1. Read papers.

The main Falx paper is this one: <https://arxiv.org/abs/2102.01024>

It is built on top of their prior work: <https://www.cs.utexas.edu/users/isil/popl20.pdf>

Falx has a page: <https://falx.cs.washington.edu/>

Read these papers and the tutorial. You are not expected to understand everything about these papers, however, you do want to focus your attention and try to answer a few important questions, such as:

* What are the key ideas in their techniques?
* How do the algorithms work? Try it on some concrete examples.
* What is the language they use to generate visualizations?
* How do users specify their desired visualizations? Any limitations?

Task 2. Try the Falx tool.

<https://github.com/Mestway/falx> they have a Github repo. Try to use their tool to solve their benchmarks: <https://github.com/Mestway/falx/tree/master/benchmarks>

Try to understand the format of each benchmark. Feel free to also play with their online demo: <https://falx.cs.washington.edu/tool>

Task 3. Identify limitations of Falx.

Try to identify a few benchmarks, or any other visualizations that you may want to create, that Falx doesn’t work so well for. There may be different reasons why it doesn’t work so well. For instance, maybe Falx is too slow. Or, it may be that Falx requires a lot of input from the user (i.e., the user needs to provide too many examples). Or, it’s possible that Falx never finds the desired visualization you want. These are just some examples, and you may find other reasons to show Falx is not good enough.

One thing that we particularly want to improve Flax upon is how it allows users to specify their intent. Currently, it only allows users to provide “partial examples”, however, oftentimes users find that not expressive enough. In this summer project, we want to identify other specification modalities that users can easily provide and that would be useful to help the system generate better visualizations. So you may want to be creative and think about other possible ways to specify the user intent along these lines. It would be great if you could demonstrate what you mean on a few concrete examples/benchmarks.

Task 4. Write a report.

Put only important things in the report. Be concise. Avoid copy-pasting or rephrasing things from their papers to your report.

You should consider putting what follows in the report, but feel free to add whatever you think is important.

* What are some benchmarks that Falx doesn’t work well for? For what reasons?
* How would you improve Falx for those benchmarks? Would that require creating a new user interface? Would that require redesigning the back-end synthesis algorithm?