[keywords: web automation, program synthesis, potentially also human-computer interaction and interface design]

[an incomplete set of technical skills: familiar with HTML/JS/XPATH/CSS, like and good at using software tools]

Our web automation project will bear some similarities to the Helena project:

<http://helena-lang.org/>

So in this “warm-up” exercise, try your best to learn about Helena, 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 Helena papers (available here: <http://helena-lang.org/publications>)

In terms of relevance, “Rousillon: Scraping Distributed Hierarchical Web Data” is the most relevant one but it is also the latest one meaning it builds upon all prior work. So if you want to read this paper first, you probably want to quickly skim through the other papers too.

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 kinds of tasks can their techniques automate? Are there any other tasks also related to web automation which cannot be handled by their techniques?
* What are the key important ideas in their papers? What assumptions do they make in order for these ideas to work? Are these assumptions reasonable? Are there cases that these assumptions don’t hold?
* How do their techniques work? For instance, how do they generate (nested) loops? How do they generate loop bodies?
* What are some limitations of their techniques? Can you give some concrete examples to illustrate these limitations?

Task 2. Use the Helena tool (available here: <http://helena-lang.org/install>)

You may choose to try the tool first if you want, but generally I’d suggest you at least take a quick look at the papers before you use the tool. You may want to revisit the papers again once you try their tool.

Try to use their tool to solve some real-world tasks. You may try to repeat whatever tasks they considered in their papers. You could also try to find some real tasks that are not covered in their paper. For instance, here are some good sources to find some interesting tasks:

* <https://forum.imacros.net/index.php?sid=a62ce79fdaf80044a2f2fce8a7ab87e6>

This is a help forum for iMacros which is a commercial tool for web automation. People post questions regarding web-related tasks they want to do. One issue is that some posts do not contain enough information for you to “reconstruct” the task they want to perform, e.g., the post doesn’t provide a URL or a website. You probably want to look at multiple posts in order to find some interesting tasks with enough details.

* StackOverflow: <https://stackoverflow.com/>

SO is a general help forum with questions on many many topics. You may try to search for interesting web related tasks using whatever keywords you can come up with. You may also need to filter out posts that don’t contain enough details.

* Helena also has a button “Saved Scripts” where you can see all scripts people have created in the past. You can try to reverse-engineer the tasks other people were trying to solve.

Feel free to use other online resources or papers to find interesting tasks.

Once you collected a good number of tasks, note them down in your report, and try to use the Helena tool to solve those problems. In general, you may encounter several possibilities:

* Helena works well. This is a good indicator that you managed to learn how to use Helena.
* Helena doesn’t work. There may be cases where Helena doesn’t work well. Why? You may need to revisit their papers, or you may need to dig into their code to see how the algorithms actually work, or another idea is to construct a “smaller” task so the tool works in order to understand which part of the original task that makes the tool not working. You should note down the reasons for each case.

For tasks that Helena can solve, what are the programs generated by it? Try to understand those programs.

Task 3. 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 real-world tasks did you find? Give the source (e.g., post links). Explain the task.
* How does Helena work on these tasks? Does it work? Why does it work? Does it not work? Why does it not work?
* What tasks is Helena good at? For the tasks that Helena can solve, what programs does it generate? What do those programs mean?
* Try to summarize at a high level the limitations of Helena. What kinds of tasks Helena in general cannot handle? Or what kinds of programs in general Helena cannot generate?
* If you were to improve Helena and overcome some of its limitations, what would you do? What’s your idea? Why would that solve the problem?