Assignment 4 – CS 4536/536: Programming Language Design
Instructor: Rose Bohrer (pronouns: she/her)
Instructor Office: Fuller Labs 139

Overview

The primary goal of the programming assignment is to develop a type-checker for a simple programming language, i.e., a program that inspects a program **without running** and either (1) reports the static type of the program or (2) reports that no such type exists. In the case that the program has a type, it should satisfy **type safety**, meaning that running the program would give back a value of the correct type.

How Long Will This Take?

My goal is that most students will spend at most 13 hours/week on this course and that few will spend more than 15 hours/week, in total.

If you have spent more than eight hours on this assignment and are not done, you are **expected** to reach out to the course staff for help (office hours, chat, or email). It is our job to help you and your job to seek help when appropriate.

Grading

This assignment is worth 100 points. 75 points are for the programming part and 25 points for the written part. Your grade for the programming part is the sum of scores for each passing test case. Your grade for the written part is 20 * the fraction of problems for which you submit an honest attempt at a solution plus 5 * the fraction of assigned peer reviews you complete.

What to Read While Working on This Assignment

See recommendations in the asgn4.rs file. Crates.io is good for Rust documentation. The textbook guides you through an algorithm for most but not all of this assignment. The asgn4.rs file explains what additions you will make to the textbook material. You are also welcome to refer to other resources about typing rules and type checkers.

Programming Assignment

Download and extract the programming assignment from Canvas. Your assignment is to implement all functions (and parser rules) to meet their specifications as given in the comments. Point breakdown:

- 12pts – numeric operations
- 3pts – strings
- 30pts – Boolean operations
- 30pts – functions

Written Assignment

20pts Design a user study which you will perform on your classmates in class. You should expect to receive feedback from course staff before the studies are performed, and feedback from classmates during or after the study.

1. What question about a programming language do you want to answer? If you are having trouble identifying a question, feel free to ask classmates and course staff for inspiration.
2. Which of the following methods do you want to use to answer the question, either because it's the best method for the job or because you want to practice the method? Due to time constraints that you will have during the in-class experiment, we recommend only using one of the following methods in your study, not mixing them:
   a. Survey
   b. Semi-Structured Interview
   c. Think-Aloud Activity
3. Follow the instructions for your chosen method. **Hand in the materials you create as part of your homework submission**:
   a. Survey: Create a list of questions for the participants to answer.
      i. Make sure that the given questions are answerable in a maximum of 10 minutes.
      ii. Write questions that gather any demographic information that is needed for your study. These questions should follow the best practices discussed in class for survey design: they should be minimized to those needed for the study, they should respect privacy, and they should be based on an inclusive treatment of participant identity.
      iii. Write at least 4 questions in Likert format to assess your research question. Answer the question: are these questions better interpreted as a Likert scale or Likert-type data?
      iv. Write at least one question with an open-ended text response to collect qualitative data.
   b. Semi-Structured Interview: Create a list of interview questions.
      i. Make sure that the given questions are answerable in a maximum of 10 minutes.
      ii. Write questions that gather any demographic information that is needed for your study. These questions should follow the best practices discussed in class for survey design: they should be minimized to those needed for the study, they should respect privacy, and they should be based on an inclusive treatment of participant identity.
      iii. Write at least 4 interview questions that address your research question. At least 2 questions should invite open-ended answers, for example, using questions like "Why" or "How".
      iv. Prepare yourself to conduct the study by writing down (a) where and how you plan to use probes to elicit follow-up responses from participants and (b) where and how you anticipate that some participants might deviate from your script
   c. Think-Aloud Activity: Prepare a handout for a single activity.
      i. Decide the form that the activity will take. Common forms for an activity are (1) showing the participant one snippet of code and asking them to answer a question about the code; the difference between this and a survey is that you ask them to talk through their reasoning while working through the original prompt (2) showing them two snippets of code which solve the same problem in

two languages and doing the above (3) asking them to step through the type-checking or evaluation process for a code snippet, or (4) asking them to write a program to perform a given task

ii. Write any instructions they need to carry out the task, which may include explaining how a given programming language feature works, telling them topics you wish them to keep in mind during the think-aloud, or telling them the parameters of interacting with the experimenter (e.g. "you can give me code and, if it does not compile, I will tell you the problem in a style similar to compiler error messages").

iii. Optionally, write down any fixed questions you plan to ask to augment the think-aloud questions, such as demographics. If you follow this step, follow the corresponding instructions from the Survey or Interview headings.

4. Carry out a first test-run of your study on one of your classmates. Time it (submit the timing information with your homework). Have that classmate give you feedback, which should follow the course rules for providing constructive feedback

5. Revise your study based on the feedback.

6. Repeat steps 4+5 with a second classmate.

Peer Reviews

Instead of peer reviews on last week's assignment, you will receive credit for giving each other feedback on your user study drafts.