Course Project

17-363/17-663: Programming Language Pragmatics
Ben Titzer and Jonathan Aldrich

Due dates listed below; 300 points total

An overarching goal of this course is to provide students the expertise necessary to (A) understand, appreciate, criticize, and reuse key ideas in programming languages, and (B) put those ideas into practice. The goal of the project is to gain a greater depth of understanding in some area of programming languages and compilers, and to demonstrate that understanding through a course project. The expected scope of the project is about 20 hours per person, spread out over the last two weeks of the course and finals week (note that we have no final exam). This is the equivalent of a little over 2 weeks of homework effort.

Project options

You have many options to consider. These include:

- Extend the language you implemented over the course of this semester with some interesting features. You could implement a construct such as continuations, for example.
- Extend the compiler or the run-time you built in some interesting way. For example, you could implement a garbage collector for the language, or you could implement an optimization pass.
- Explore some area of programming languages theory in SASyLF, proving some interesting result about a language construct you are interested in.
- Design a language construct and specify its semantics (not necessarily with SASyLF, though you could).
- Extend some language or compiler other than the one you built in class. For example, you could contribute to an open-source language or compiler. Even a small pull request can be a worthy accomplishment in the context of a substantial, real system. See below for more guidance/details.
- Critique state-of-the-art research in a programming language topic of your choice by reading, analyzing, and comparing at least two research papers in that area in a detailed essay. See below for more guidance/details.

If you are not sure what to do but you have an area of interest, contact the course staff and we can help you refine your interests.

If you have an idea for something you want to do that somehow doesn't fit in any of the above options, contact us! We are happy to work with you to find something that fits with your interests.

We provide additional guidance for several of the above options in the Appendix, below.

Deliverables

The project consists of two deliverables, each with its own due date.

See the Appendix for more specific guidance, per-project-type, for each of these deliverables.

- **Proposal:** *(50 points)* Due on November 21, 2023, 11:59 pm). Turn in as a PDF.
- **Presentation Slides:** *(250 points)* 11:59 pm the day before the "final exam" slot. Nominally there are no late days permitted. Turn in as PDF. If you chose the "paper critique" option, append the PDF critique essay to your presentation slides document, and/or combine the two PDFs in a zip file.

Teamwork. Students may work in pairs, of their own selection. Pair projects will be given a single grade, with exceptions only in extraordinary circumstances. If you want to work in a group larger than a pair, reach out to the instructors. Expectations for the project will be scaled (within reason) to the size of the group.

Project scope/proposal. *(Tuesday, November 21, 50 points)* The first deliverable is description of what you intend to do for your project. This description can be brief (1–2 paragraphs). List the project type, title and the members of your group (or indicate that you are working alone). Outline what you are going to do, why it is interesting, and how you are going to evaluate it.

Presentation. *(11:59 pm Day before the Final Exam Slot; 250 points)*. Prepare a presentation of at most 10 minutes (we will enforce this!) describing your project for the benefit of your peers. Note that by default, we cannot allow late days for the presentation. Describe the problem you are solving and why it is important; give high-level background not already provided in the class; explain how you solved the problem, and show an indicative result or short demo, depending on the contribution of your project. Focus on identifying a cool thing you did or learned, and showing or explaining that cool thing to your classmates. We will structure the final exam slot as a series of presentations, and interaction/questions will be considered in final grading.

If you chose the "paper critique" option, turn in your essay as well.

Appendix

In this section, we provide some additional guidance on what we expect in the various deliverables based on the type of project you propose.

Open Source Contribution

There exists a large number of open-source analysis tools on GitHub.

If you choose this project type, your goal is to select an open-source project and complete one or more bug fixes or extensions within it. You have considerable freedom in which project and tasks you choose. In general, you are more likely to succeed if the project is active and has multiple contributors. Beyond that:

- You *must* choose a bug report or feature request from a public database or message board, following whatever protocol the project uses to communicate and track open issues. Do not invent a task; Address an actual, documented project need.
- The task must require changes to the project's source code. Pure documentation or design tasks are not appropriate.
- You may choose one large task or several smaller, related tasks.

Choosing a task of the appropriate size can be challenging; in estimating, bear in mind that it takes time to get the lay of the land with a new project.

For the proposal, ideally you will be able to list both the project and 1–3 candidate tasks you hope to take on. At minimum, you should give some options of projects/tasks so we can smell check the scope for you.

For the presentation, be sure to explain at least what the project does as well as necessary context to understand the nature of the task you took on. So long as you aren't "scooped" on the task, we do expect you to *submit your changes to the project via pull request*. This may involve adhering to project-specific submission and code quality guidelines, or communicating with members of the project community to help support acceptance. If your pull request is accepted, you will receive extra credit.

Critique[1]

If you take on a research critique, your goal will be to conduct a deeper dive into a particular area of analysis. You will read at least 2 papers in the area, and write a detailed essay (a "critique") that discusses and compares the approaches taken in the two papers. You should aim to analyze two research papers that solve the same problem in different ways, and then compare and contrast the different approaches, discussing what the papers share in common, what is different across them, and to what extent those differences are complementary, orthogonal, or mutually exclusive to each other.

You will be choosing the set of two research papers that you analyze yourself, subject to approval by the course staff, as well as these constraints:

- At least one paper from your set must have appeared within the past five years (the other can be older than that, if necessary).
- The papers must be full-length conference or journal papers, which are typically 10 or more pages long. (They cannot be "short papers" that are just a few pages long).
- None of the papers should be "survey" papers (which summarize a research area, but do not focus on presenting new approaches to solving the problems).
- Your set of papers must be approved by the course staff.

In the proposal, you should at least identify the area in which you would like to read papers, and at least one paper you will start with. If you would like to do this option but are flummoxed, reach out to us before the proposal deadline and we will help you.

Some tips:

- Perhaps the most useful starting point for finding papers that are closely related to a given paper is to look at the list of papers that it cites. A research paper is supposed to cite (and discuss) previous work that is closely related. The list of citations appears at the end of the paper (usually under the heading, "References"), and the most relevant citations are often discussed within the paper (sometimes under an explicit "Related Work" section).
- While citations are helpful in identifying related work that was published a year or more before a given paper, what about related papers that were published either concurrently with or after a given paper? One way to identify closely-related conference papers that appeared concurrently with each other is to look at the session in which they were published at the conference: conferences are usually organized thematically into sessions with particular themes, and there is a good chance that closely-related papers will end up within the same session.

[1]This material is adapted from an assignment from 15-300.

- It is also helpful to identify papers that appeared later in time that cite a given paper. Several of the major web sites that allow you to search for CS research papers (e.g., the ACM Digital Library (`http://dl.acm.org`), Google Scholar (`http://scholar.google.com`), etc.) include a "Cited By" feature.
- Finally, another very helpful resource for identifying sets of closely-related research papers is a PhD student who is working in that area. PhD students are usually quite familiar with all of the related work in their area of study, or the course staff. Feel free to reach out to such students, or to us, for pointers.

For the presentation, be sure to at least explain the high-level problem the papers are solving, and key similarities/differences in the approaches. You don't have time to go into incredible detail on both; focus on the most important or interesting features of each approach.

The bulk of the report should focus on describing and comparing the two research papers. First describe the "big picture": what problem are they trying to solve, why is this problem important and challenging, and any relevant background. Then, describe each of the two papers: What approach did they take to solve the research problem? What were their key contributions? What were their limitations? Finally, you will also contrast the twopapers: What did they share in common? What were the major differences between the papers? Regarding these differences, were they complementary, orthogonal, or mutually exclusive to each other? Conclude as you see fit, but one good approach might be to identify questions that remain open, limitations of all of the approaches, or a judgement about which approach is most promising for future application or research.