CSCI468 Compilers Portfolio

Logan Shy*
Montana State University
Bozeman, Montana
loganshy740@gmail.com

Chris Erickson
Montana State University
Bozeman, Montana
erickson.christopher2@gmail.com

James Jacobs
Montana State University
Bozeman, Montana
x444428@yahoo.com

ABSTRACT

A compiler is a complex, and necessary part of life in the modern technological world. Virtually every language utilizes one before anything can be executed, and many of it's functions are hidden to the developer. This paper's goal is to provide insight and show the implementation and design of a compiler.

ACM Reference Format:
Logan Shy, Chris Erickson, and James Jacobs. 2020. CSCI468 Compilers Portfolio. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROGRAM

2 TEAMWORK

3 DESIGN PATTERN

4 TECHNICAL REPORT

4.1 Introduction

4.2 Background

4.3 Methods and Discussion

4.3.a Tools and Setup. To generate a parser and other necessary tools, the software that would become our main focus would be Antlr, which stands for ANother Tool for Language Recognition. The setup for Antlr comes in the form of a jar file, executable on almost any machine that can run the JVM. Once the jar file is installed, it was necessary to add the file's parent directory to my local machines CLASSPATH environment variable. For convenience, quick CLI command alias' were created for both Antlr and Grun, the testing framework. The steps taken on a Linux-based environment are similar if not identical, however most of the grunt work is done over the command line. To use the software, the command shortcut is used on the g4 grammar file. Afterwards, javac is then used to compile all the different .java's into an executable. To test this, Grun is used on the executable with specific strings used to test the parse tree.

As we continued to talk over email, we quickly realized that for the convenience of all the group members involved, the ability to communicate needed to be streamlined. Our solution was to set up a free workplace/messaging forum on the tool known as Slack. The ability to download an app to your phone and never miss a notification was truly a selling point that we didn't want to miss out on, aside from the fact that it was free. The functionality and look-and-feel of the client are easy to use and clean, plus certain members already had prior experience with the tool. The setup was relatively simple, and allows the user to decide how much they should install. The bare minimum is to sign up on the Slack.com website, and create an account. After choosing a payment plan (our group chose the free option) and naming a workspace, all that's left is to invite group members via their primary email address. If no account is associated with said email, they will be prompted to create one upon receiving and opening the email. To take the setup another step further, one could install the desktop application onto their home computer to negate the use of a web browser. Additionally, an application can also be installed onto one's smartphone.

Next on the list was to utilize a versioning system for our semester-long project. We settled on Git, given it's ubiquitous nature and members' prior experience with the tool. Not only does Git allow versioning, but it also keeps the working master code separate from whichever section of code a developer decides to work on. With a built in peer review system, buggy or flawed code rarely makes it to the production side of the working codebase. The setup for Git is a bit more extensive than our choice for communication. However, the documentation on Git's webpage is easy to follow, and holds your hand every step of the way. After downloading from a browser, among the files will be the Git Bash, which is a CLI with built in Git commands to be used right out of the box. After setting the users email and personal editor, one is free to initialize and clone repositories as they wish. Our group initialized the repo on Github.com, and then cloned from it and committed all basic files (i.e. README, .gitignore, .g4).

With all of these tools combined, our workflow pattern is as optimized as nearly possible under the given circumstances. These first steps are some of the most critical in terms of group teamwork and general survival, and any group effort would do well to take similar approaches. More often than not, miscommunication and inefficiency play devil's advocate against a team of developers. It can be easy to get bogged down in the mire of extraneous details and logistics, but these problems can be avoided if the proper steps are taken at the proper time. By putting our best foot forward to prevent these hindrances, our hope is that we can relieve ourselves of the time wasters and focus solely on the coding and problem-solving near at hand and on the horizon. As problems arise, which always do, the tools at hand and the perseverance of the team will be the key things that allow these problems to make or break us.

4.3.b Scanner.

4.3.c Parser.

*All authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

4.3.d Symbol Table.

4.3.e Code Generation.

4.3.f Full-fledged compiler.

4.4 Conclusion and Future Work

5 UML

6 DESIGN TRADE-OFFS

7 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

REFERENCES

A RESEARCH METHODS

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B ONLINE RESOURCES

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.