GLASS

Project Plan - Semester Two

Team: Tommy Galletta, Alexander Lockard

Faculty Advisor: Dr. Stansifer

Goals and Motivation

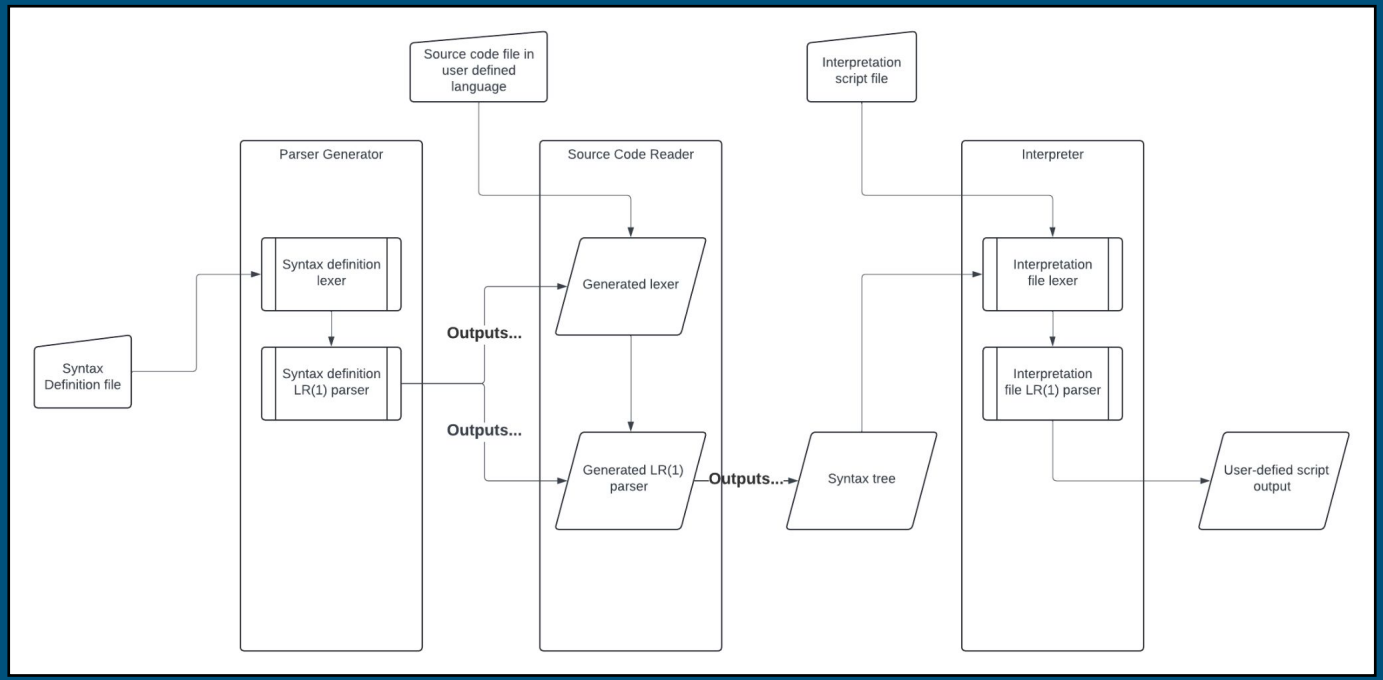
Goals:

- To examine and understand the inner workings of parser generators, both from a theory standpoint as well as by examining pre-existing tools.
- To build our own parser generator tool that attempts to resolve some of the issues of pre-existing tools in an elegant manner.
- To implement a GUI system and a script system around the parser generator that gives the tool additional flexibility while maintaining ease of use.

Motivation:

- We hope to learn about the inner workings of parser generators, as well as the theory that goes along with it.
- We hope to develop a tool that addresses the complexities of other parser generators by having a simpler way to specify syntax and semantics.
- We hope to build a small extension for the parser generator, that being the script system, that allows for parsed code to be interpreted to a certain degree.

Design


Source code file in user defined language
Interpretation script file
Parser Generator
Source Code Reader
Interpreter
Syntax definition lexer
Generated lexer
Interpretation file lexer
Syntax Definition file
Outputs...
Syntax definition LR(1) parser
Interpretation file LR(1) parser
Outputs...
Generated LR(1) parser
Outputs...
Syntax tree
User-defied script output

Approach

- Grammar specification and parser generation
- Language binding utility APIs
- GUI based grammar specification tool
- Interpretatio script parsing tool
- Extensive documentation

Technical Challenges

Defining the structure of the syntax specification file

- With the goal of our tool being simplicity, we will have to closely examine the different specification languages used within parser generator tools to determine which choices we should implement in our language and which we should avoid.

Making the parser generator

- We will have to construct the parser generator to allow our tool to parse a specified language. This will require a dive into how parsers work, as well as the theory that goes along with it.

Designing the script interpretation tool

- We will have to decide which built-in functions we should include in our application, and then determine how to implement them.

Novel Features

Integration of grammar definition file and language interpretation file

- Our tool aims to separate grammar definition and a language's functionalities into two separate files, while internally integrating them to perform the same operations that other parser generator tools would allow.
- The end goal of this approach is to make the input files both more readable and easier for the user to write themselves.

Algorithms and Tools

LR(1) Parsing Algorithm

- Allows us to parse through a large number of grammars and generate a parse tree.

Parser Generation

- Allows us to construct an LR(1) parse table from a series of grammar productions. These productions may be user-defined, allowing the user to generate parse tables for their own grammars.

React

- We will use React as a means of providing our documentation in a well-organized manner for users to access via the Internet.

Researched Tools

- NTLR, bison, tree-sitter, lemon, JQ, and Visual BNF have been incredibly helpful in providing insight into how we should go about developing our project.

Evaluation

Ease of use

- We will have several computer science students to “test drive” our tool. They will first fill out a small survey about their knowledge related to our tool. Then, the user will be asked to perform a series of predefined tasks, having full access to the documentation we provide for the tool. We will note how long it takes each user to complete each task.

User survey

- After all tasks have been completed and completion times have been recorded, there will be a short user survey asking about their experience using the tool.

Satisfying different use cases

- Can we use our tools to satisfy the use cases we originally set out to develop it for?

Progress Summary

Feature	Completion %	Todo
Grammar specification and parser generation	90%	Debugging and code cleaning, potential room for options and ease of use features
Language binding utility APIs	0%	Fully implement Java and Python bindings
GUI based grammar specification tool	0%	Create and integrate the GUI tool
Script interpretation tool	10%	Fully implement script interpretation
Extensive documentation	20%	Complete documentation for existing features, transfer documentation to project website

Milestone 4

Documentation

- Continue writing documentation
- Host documentation on project website

GUI

- Create GUI based tool for creating syntax definitions
- Integrate GUI such that generated syntax definitions can be immediately used to parse a source code file

Debugging / Code cleaning

- Debug and clean code for pre-existing systems
- Ensure GitHub repo is well-structured

Ease of use features

- Implement features that allow for easier debugging of “broken” grammars
- Implement default values and settings that can be overridden for syntax definitions

Milestone 5

Main system GUI

- Implement GUI for main system interactions (selecting syntax definition, source code, and interpretation script files)

Script interpretation

- Implement a system to read, parse, and process a user-defined script to be applied to a parse tree. The actions defined in the script should be executed at the appropriate time during the traversal of the parse tree.

Documentation

- Update and extend documentation on website as appropriate

Poster

- Create presentation poster

Milestone 5

Main system GUI

Implement GUI for main system interactions (selecting syntax definition, source code, and interpretation script files)

Script interpretation

Implement a system to read, parse, and process a user-defined script to be applied to a parse tree. The actions defined in the script should be executed at the appropriate time during the traversal of the parse tree.

Documentation

Update and extend documentation on website as appropriate

Poster

Create presentation poster

Milestone 6

Evaluation

- Conduct evaluation and analyze results

Finalized Project Items

- Test/demo of the entire system
- Create user/developer manual
- Create demo video

Milestone 4 Task Matrix

Task	Tommy	Xander
Continue writing documentation and host it on project website	75%	25%
GUI-based syntax specification tool	10%	90%
Debug / clean currently implemented systems	75%	25%
Additional ease of use features	75%	25%

Questions?