

# GLASS

(Generalized Language Abstraction and Scripting System)

Tommy Galletta, Alexander Lockard

Faculty Advisor: Dr. Ryan Stansifer, College of Engineering and Science, Florida Institute of Technology

## Goal

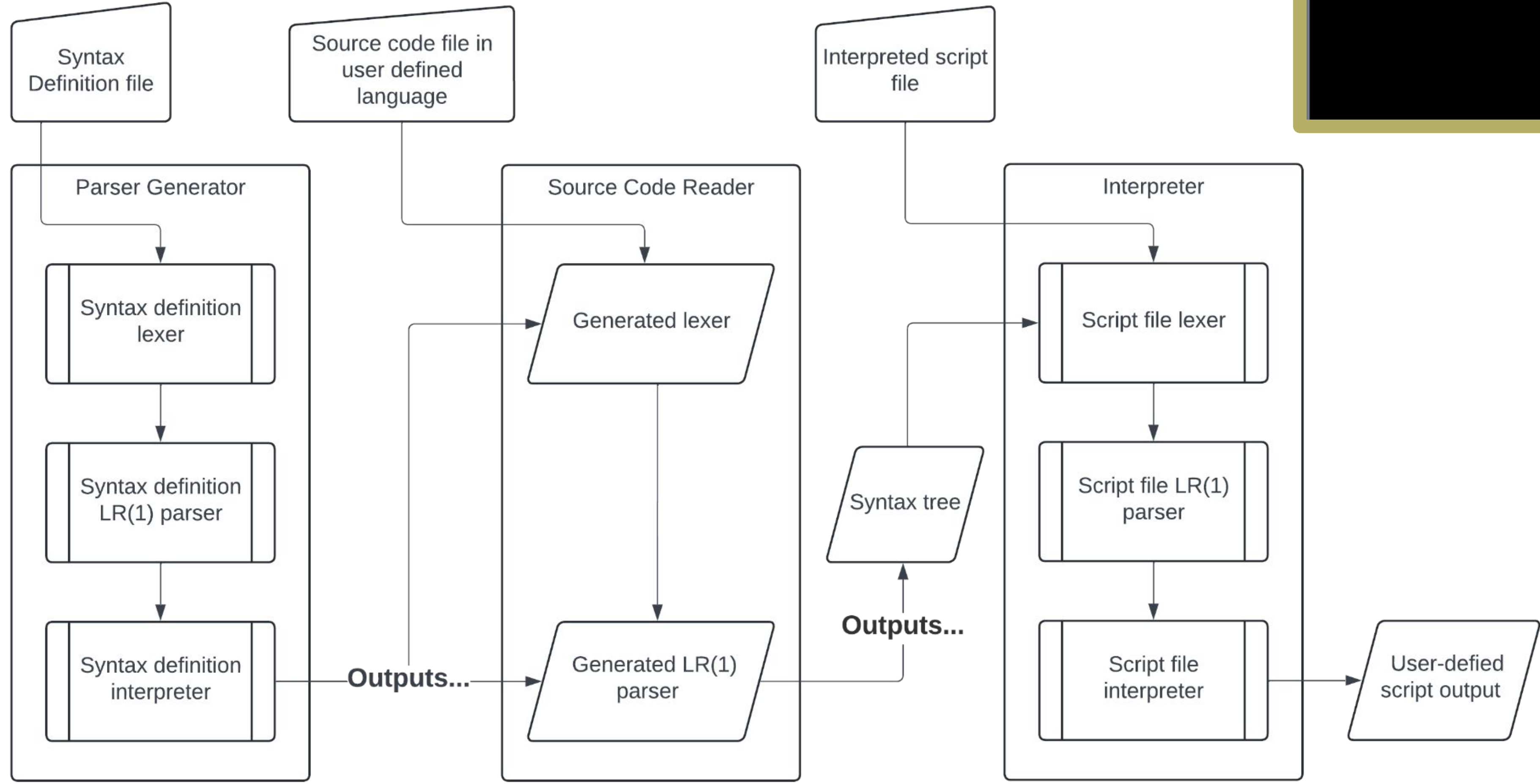
To create a parser generator tool that attempts to tackle some of the difficulties that exist in the current leading parser generator tools.

## Motivation

To address the pains users have had when trying to use other parser generator tools, and related tools. This list includes:

- Overwhelming documentation with lots of assumptions of knowledge.
- Confusing syntax that interweave existing programming languages with other syntax.
- High amounts of boilerplate needed for simple tasks.
- Pipeline makes use of multiple different tools (no “all in one” solution).

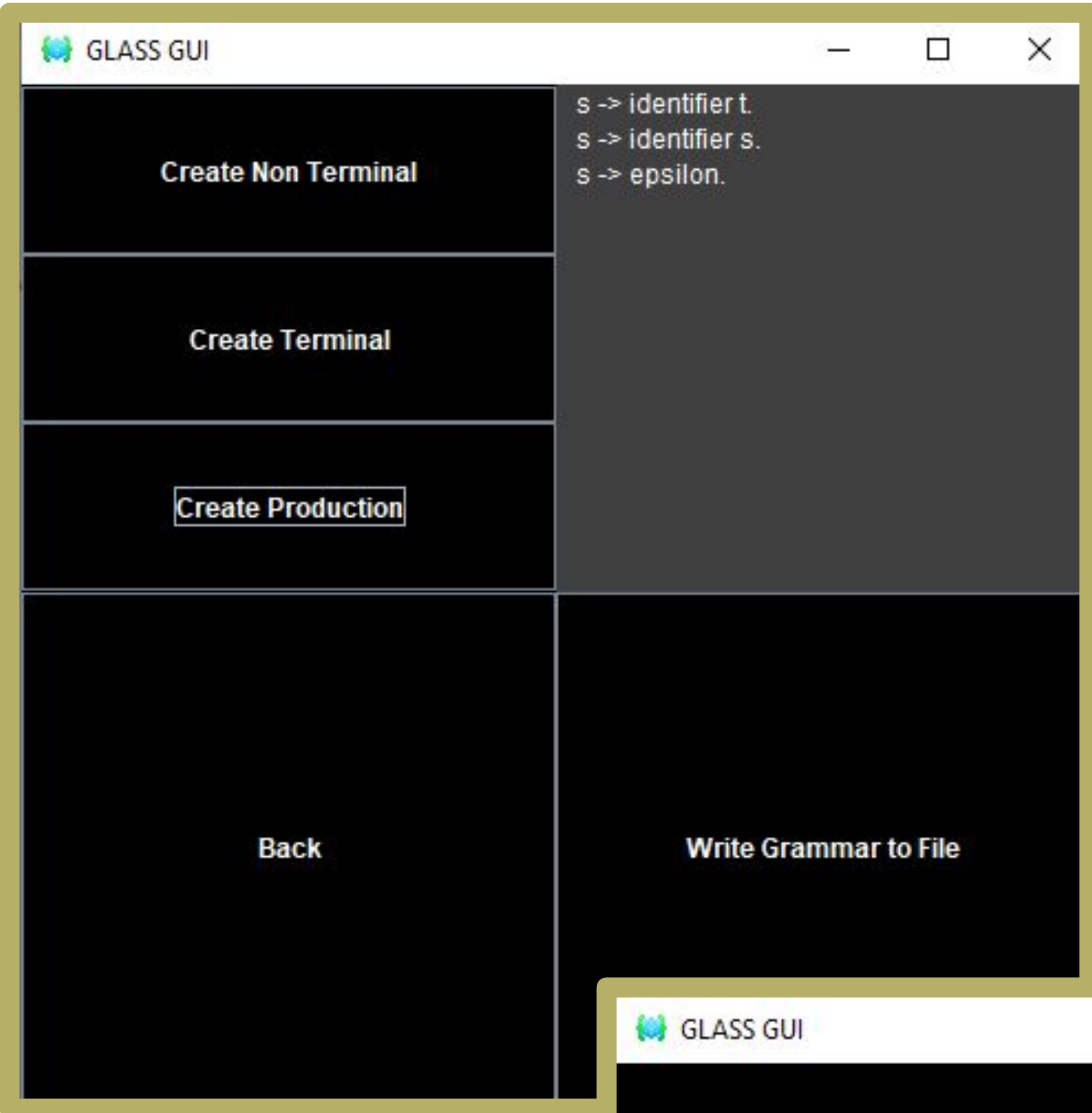
## System Architecture



## Features

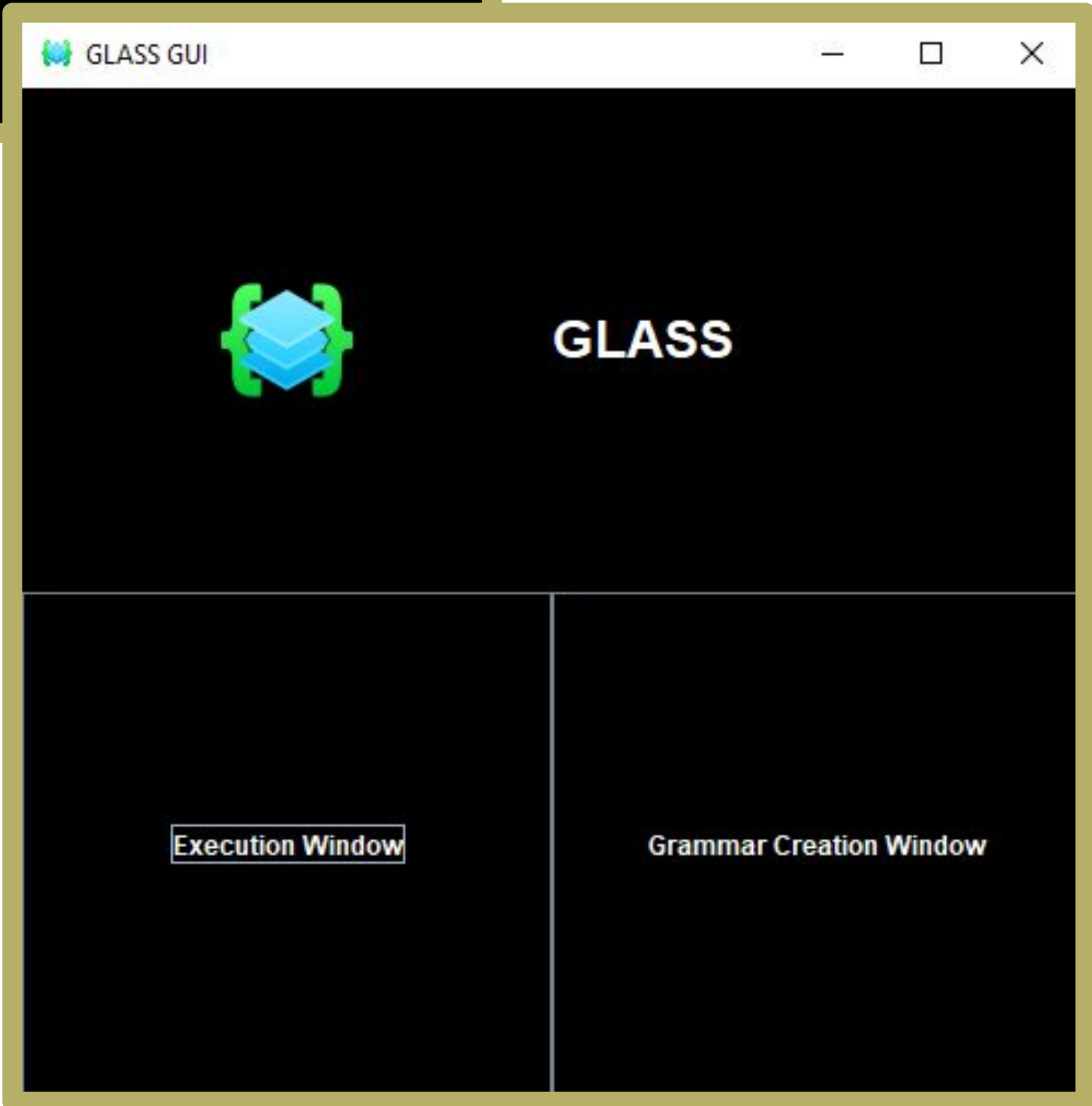
- File tokens can be defined using regex.
- Certain tokens (such as whitespace and comments) can be defined to be accepted in lexing but ignored during parsing.
- File structure can be defined using BNF grammar productions.
- Script with interpretation instructions written in a simplified Java-like scripting language.
- Debug options to address issues when creating grammars and scripts.

## GUI



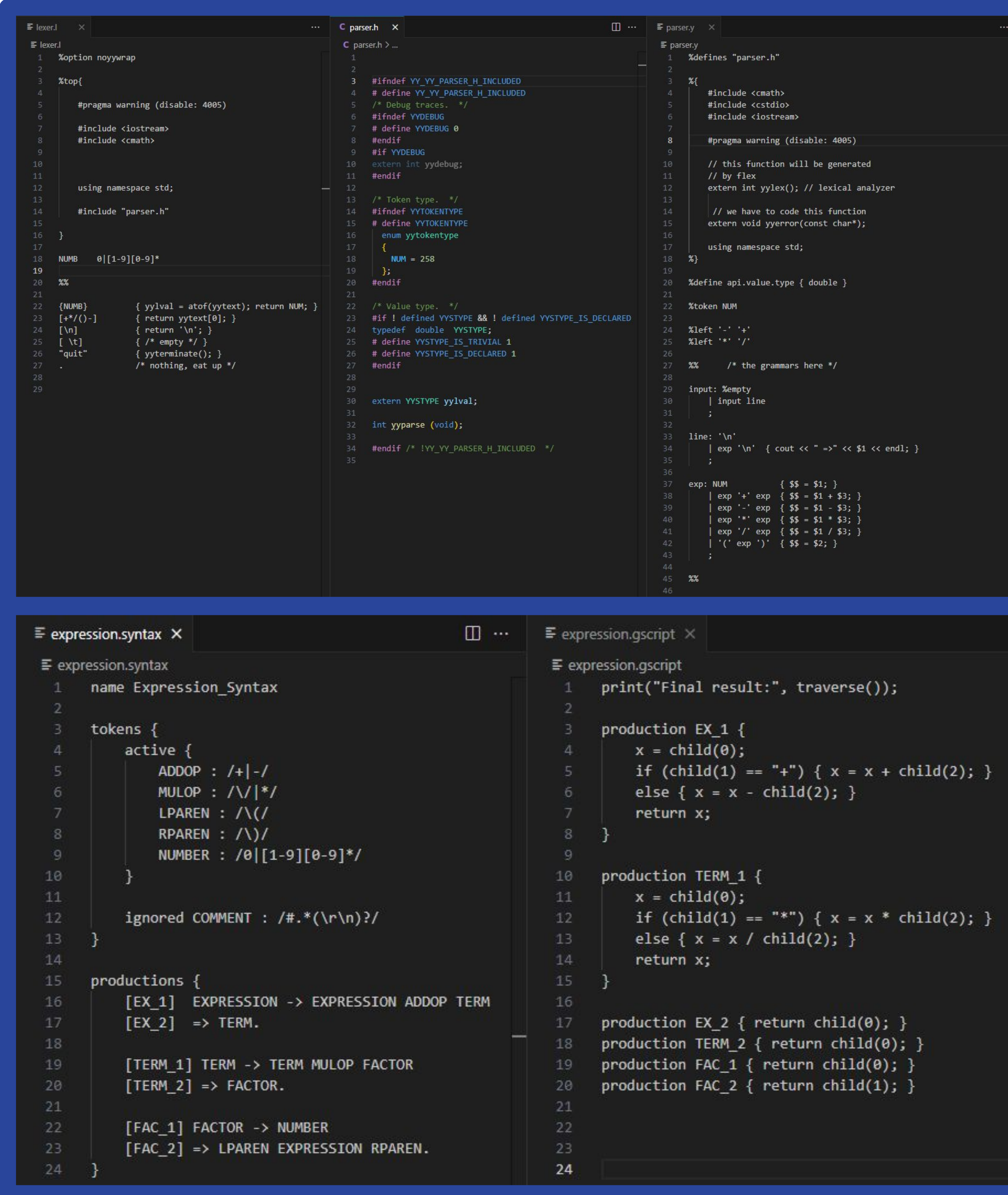
GUI for creating syntax definition file and executing the tool.

Grammar creation window (left)  
Main menu window (below).



## Comparison: GLASS vs. flex/bison

Below is an example of the code needed to make an expression evaluator in flex/bison (above) and in GLASS (below).



Scan this QR code to view our documentation!

