Compiling `ParL` to `PArIR` in Rust

A report on the Rust compiler for the `ParL` programming language.

Giorgio Grigolo - 0418803L

Contents

Abstract

In this report, we discuss the implementation details of a compiler for `ParL`, an expression-based strongly typed programming langauge. Code written in `ParL` is compiled to `PArIR`, which is the proprietary assembly-like language that is used to drive the programmable pixel art displays designed by the company `PArDis`. The `ParL` compiler was written in Rust, due to its strong type system and performance characteristics. It nwas implemented incrementally, as to ensure each component can be run in isolation, and is working correctly before moving on to the next one.

1   Lexical Analysis

The first step in the compilation process is the lexical analysis. A recurring theme in the implementation of this compiler is the use of *abstraction* to achieve *modularity*.