The PixARLang compiler

CPS2000 Compiler Theory and Practice

Mark Mizzi

Last edited: March 25, 2023

Contents

1 Lexing

1.1 Code organization

The implementation of the lexer is contained in the two source files src/lexer.hh and src/lexer.cc. The code is contained in a lexer namespace.

Tokens are represented by an instance of struct Token. Each token has a tag field of type TokenType which identifies the kind of token (identifier, integer literal, etc.), and a value field which contains the substring of input which the lexer consumed to produce the token. In addition, each token carries location information in four fields, giving the line and column of the start and end of the token.

The lexer implementation is encapsulated in a Lexer class. Instances of this class are passed the program input as a std::string on construction.

The class provides GetNextToken() as an interface method to produce the next token from the input.

1.2 Lexing approach

There are several approaches available to the compiler implementer when it comes to writing a hand-coded lexer.

This implementation uses a table-driven approach to lexing. The core functionality of the lexer is implemented in the method Lexer::nextToken().

2 Parsing