CS323 SPL Parser - Phase 1

陈康睿	樊斯特	肖佳辰
12110524	12111624	12112012

Lexical Part

In the file `lex.l`, we detect valid and invalid tokens and build nodes of the parse tree for them, and invalid tokens are used for error recovery for subsequent syntax analysis.

In this section, we define a `has_error` variable to record whether or not an error occurred during compilation. `has_error` is initialized to 0 and is set to 1 after either a TYPE A or TYPE B error, and after the analysis is complete, the parse tree is not output if there is an error in the process.

Syntactical Part

In the file `syntax.y`, we formulate the matching rules used to analyze the syntax and use the node pointers of the parse tree as nonterminal values, so that a complete parse tree can be constructed by Bison's syntax analysis.

When performing the matching, we specify some common errors.

Parse Tree

`treeNode.c` and `treeNode.h` define the structure `treeNode` of the syntactic analysis tree, as well as a set of functions necessary to build the tree. Each tree node holds the following variables:

- name: the name of the node of the (non)terminal
- val: the content of the terminal
- lineno: the line number in the source file that corresponds to this (non)terminal
- child: head of the list of child nodes
- nxt: pointer to a sibling node