

Link to Github repo for Lab4: <https://github.com/Catalin-David/FLCD/tree/main/Lab4>

932/2 David Catalin-Ioan

Lab4

The Symbol Table was implemented using a hash table.

PifEntry represents an entry inside the Program Internal Form. The class has 3 fields: one is the type of the entry: "identifier", "constant" otherwise the reserved word (e.g. "%", "val", "int" etc.). The hashValue and index are the position inside the Symbol Table if it is an identifier or a constant, otherwise those are null.

The Pif class contains a list of PifEntries, on which we can perform basic operations such as addition, and it represents the Program Internal Form.

The FileReader and FileWriter are classes that I defined in order to read and write to files, line by line.

The Scanner contains the most important method, scanProgram, and multiple helper functions.

- * scanProgram - receives the path of a file containing source code of the toy language and prints whether the program is lexically correct or not. In case it is not, it prints the line and token where the lexical error was found.
- * readTokens - reads and stores the tokens of the toy language (reserved words, separators and operators).
- * isToken - returns whether a word is a token inside the toy language or not.
- * splitWords - receives one line of source code and splits it into words (identifiers, constants and tokens).
- * printData - prints the information found inside the Symbol Table and Program Internal Form.

The FiniteAutomata class:

- * readFiniteAutomata - reads from FA.in and creates the components of a FiniteAutomata: states, alphabet, initial state, final states and transitions
- * checkDeterministic - checks whether a FA is deterministic
- * verifySequence - verify whether a sequence is accepted by a FA

* `splitIntoString` splits a line into strings, `splitIntoInts` splits a line into integers

The `Transition` class represents a transition inside the FA.