Documentation for IPPeCode Parser

Task 1 - parser.py

Alessandro Amella
xamella00

March 26, 2024

1 Introduction

The script `parser.py` is designed to tokenize and parse IPPeCode code, converting it into an XML format. It utilizes the PLY (Python Lex-Yacc) library for lexical analysis (tokenization) and syntax analysis (parsing), and uses the Python Standard Library's XML modules for generating and managing XML output.

2 Lexical Analysis

Lexical analysis is performed using PLY's `lex` module. The script defines a set of tokens which includes the language operands, along with regular expressions for each.

Special tokens and actions are defined for handling whitespace, newline characters to track line numbers, and illegal characters to flag errors.

3 Syntax Analysis

For syntax analysis, the script uses PLY's `yacc` module. Production rules in the script define the structure of valid IPPeCode syntax, beginning with a top-level `program` rule that consists of a list of three-address codes (TAC) elements.

An abstract syntax tree is built by grouping these into actions with specific types and values.

4 XML Output Generation

The script generates XML output based on the parsed data. It constructs an XML document using the ElementTree API, creating elements for the program and each TAC with their operands. XML attributes and text content are properly escaped.

5 Debugging and Error Handling

The script offers debug mode for detailed process logging, including tokenization and parsing steps, via command-line arguments. Additionally, it implements error handling mechanisms to report illegal characters during lexing and unexpected tokens during parsing.

6 Conclusion

This script is a robust implementation of the IPPeCode parser which, by specifying a context-free grammar (CFG) and making an abstract syntax tree (AST), allows easy future expansion with an expanded set of instructions.