**Cover Page**

**CS323 Programming Assignments**

Names [ 1. Chance Noonan ] Section [ 7 ]

[ 2. Brian Alvarez ] Section [ 7 ]

[ 3. Vance Tran ] Section [ 7 ]

2. Assignment Number [ 2 ]

3. Due Date [ 7 April, 2024 ]

4. Submission date [ 7 April, 2024 ]

5. Executable File name [ compiler.py ]

6. Names of the test case files - input test file output test file

test 1. [demo.rat24s] [tokenized\_demo1.txt]

test 2. [] []

test 3. [] []

7. Operating System [Windows ]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**To be filled out by the Instructor:**

Comments and Grade:

**CS323 Documentation**

1. **Problem Statement**

The objective of this project is to develop a syntax analyzer (top-down parser) that can process the token stream generated by the lexical analyzer created in assignment 1 for the Rat24S language.The analyzer shall check each token against the Rat24S grammar, and the Parser class shall output which grammar rules are being applied at each step of the parsing process. The main file, which is now compiler.py, invokes the lexical analyzer to break down the source code into a token stream that is fed into syntaxer.py where the Parser class resides.

1. **How to use your program**

To execute the program, run compiler.py with python3 and a test case in the terminal.

EX:

python3 compiler.py demo.rat24s

1. **Design of your program**

The design of the syntax analyzer is centered around the *Parser* class, which is designed to iteratively process a stream of tokens and conform to the Rat24S BNF specifications. The class is initialized with a list of tokens generated by the lexical analyzer that take the form of tuples holding data pertaining to the type of token, its corresponding lexeme, and the line number from the .Rat24S file source code.

The *next\_token* method advances the parser token by token, ensuring that the analyzer interprets each one sequentially. The *match* method works together with the *next\_token* method by comparing the current token to the expected token or symbol. This method is responsible for generating syntax error messages when the actual and expected tokens do not match. It also plays a key role in processing (*$*) delimiters by recognizing the start and end of sections of Rat24S code. The *parse* function is what begins the parsing process by invoking the *Rat24S* method which represents the top-level rule of the grammar. Once called, the parsing process employs a recursive descent strategy where it goes through a series of function calls, with each individual function representing one of the twenty-nine BNF rules. This structure is what allows the analyzer to produce a hierarchical output of the grammar rule applications associated with each step of parsing the input code.

**Left Recursion Removals:**

R25.

<Expression> ::= <Term> <Expression’>   
<Expression Prime> ::= (+ <Term> <Expression’>) | (- <Term> <Expression’>) | <Empty>

R26.   
<Term> ::= <Factor> <Term’>  
<Term Prime> ::= (\*<Factor> <Term’>) | (/ <Factor> <Term’>) | <Empty>

**Left Factorization**

R3.

<Function Definitions> ::= <Function><Function Definitions’>

<Function Definitions’> ::= <Empty> | <Function Definitions’>

R6.

<Parameter List> ::= <Parameter> <Parameter List’>

<Parameter List’> :: = ,<Parameter List’> | <Empty>

R11.

<Declaration List> ::= <Declaration><Declaration List’> |

**<**Declaration List’> ::= ;<Declaration List’> | <Empty>

R13.

<IDs> ::= <Identifier><IDs’>

<IDs’> ::= ,<IDs> | <Empty>

R14.

<Statement List> ::= <Statement> <Statement List’>

<Statement List’> ::= <Statement> <Statement List’> | <Empty>

R19

<Return> ::= return <Return’>;

<Return’> ::= <Expression> | <Empty>

1. **Any Limitation**

None; every rule associated with the grammar is represented by a method in the *Parser* class

1. **Any Shortcomings**

* Expressions must follow strict identifier, operator, identifier format in order to be recognized by the analyzer
* Parser fails when it does not find another identifier where it expects one
* Error messages lack specificity