Activity based

Project Report on

System Programming

Submitted to Vishwakarma University, Pune

Under the Initiative of

Contemporary Curriculum, Pedagogy, and Practice (C2P2)

By

Yashashree Sagar Bedmutha

SRN No : 202100177

Roll No : 02

Div : E

Third Year Engineering

Department of Computer Engineering

Faculty of Science and Technology

Academic Year

2023-2024

1.1. Abstract

The main goal of the project is to develop a flexible lexical analyzer for the English language. Tokenization, classification, design and implementation, and its wide range of applications in language processing are all included in this comprehensive project. English text is effectively broken up into structured tokens by the analyzer, providing an essential basis for more complicated natural language processing tasks. A crucial aspect is its capacity to recognize and classify tokens for a range of applications, including text categorization, sentiment analysis, and information retrieval. The goal of this project is to produce a reliable and versatile tool that can be used in a variety of language processing contexts. The method simplifies the process of obtaining insightful information from unstructured English text data by use of rigorous design and execution procedures.

1.2.Introduction

Within the field of programming and natural language processing, the main objective of this project is to develop a basic, yet flexible, lexical analyzer that is specifically designed for the English language. A thorough examination of the analyzer's lifetime, including crucial stages like design and implementation, tokenization, classification, and its extensive application in the field of language processing, is required for this project.

Careful design considerations are prioritized from the beginning. The project begins with a strategic blueprint that carefully shapes the data structures, algorithms, and architectural elements of the lexical analyzer. This design step converts abstract design principles into tangible code, laying the groundwork for the implementation that follows. The goal is to give the analyzer the capacity to parse input text, recognize and classify tokens quickly, and follow pre-established lexical rules.

To improve analyzability, tokenization—the division of English text into meaningful units like words, numbers, punctuation, and operators—is a crucial component. Classification and recognition play equally important roles in allowing the analyzer to recognize and classify different kinds of tokens according to pre-established guidelines. Due to its adaptability, the tool can be easily integrated into a wide range of language processing applications, such as information retrieval, text categorization, and sentiment analysis.

The analyzer serves as the cornerstone for more complex natural language processing activities as well. It supports tasks like named entity recognition, sentiment analysis, and part-of-

speech tagging, which enable the extraction of insightful information from unstructured English text data.

1.3. Literature Summary

- Fundamental understanding of the concepts of tokenization and text segmentation a vital consideration in the project's design and execution can be gained from lexical analysis and natural language processing literature.
- In line with the project's methodology, regular expressions are extensively studied in the literature as crucial tools for tokenization. Researchers describe how to utilize them to identify and extract different sorts of tokens from text in an efficient manner.
- The literature highlights the flexibility and adaptability of lexical analyzers and discusses how they serve as the basis for more complex natural language processing tasks including sentiment analysis, named entity recognition, and part-of-speech tagging.
- Researchers stress the importance of precise and meaningful tokenization, which is a fundamental aspect of the project's goals, and the need for well-defined lexical rules.
- The literature emphasizes how crucial lexical analyzers are to making it possible to extract useful information from unstructured text data, which is consistent with the project's goal of developing a fundamental and adaptable tool.

1.4. Methodology

1. Lexical Rules:
    a. Nouns:
        i. Lexical Guidelines: Typically, nouns in English texts are words that are used to refer to concepts, persons, locations, or things. Common nouns like "cat" and "car" as well as proper nouns like "London" and "Microsoft" may be among them. Both plural nouns (NNS) and singular nouns (NN) are acknowledged.
        ii. Examples of Regular Expression Patterns: r'\b\w+\b'
        iii. Example: "cat," "car," "London," and "Microsoft"
    b. Pronouns:
        i. Lexical Rules: In English, pronouns take the role of nouns and are frequently used to refer to people, things, or concepts that have already been discussed

in the text. Personal pronouns (like "he," "she"), possessive pronouns (like "his," "its"), and pronouns (like "who," "which") fall under this category.

ii. Examples of the Regular Expression Pattern "r'\b\w+\b'" include "he," "she," "his," and "who

c. Verbs:

i. Lexical Rules: Verbs are action words in English, representing actions, occurrences, or states. They encompass various forms, such as the base form (VB), past tense (VBD), gerund or present participle (VBG), past participle (VBN), base form for non-3rd person singular (VBP), and 3rd person singular present tense (VBZ).

ii. Regular Expression Pattern: r'\b\w+\b'

iii. Examples: "run," "ran," "running," "eaten."

d. Adverbs:

i. Lexical Rules: Adverbs modify verbs, adjectives, or other adverbs to provide more information about how, when, or to what extent an action occurred. Adverbs include simple adverbs (RB), comparative adverbs (RBR), and superlative adverbs (RBS).

ii. Regular Expression Pattern: r'\b\w+\b'

iii. Examples: "quickly," "faster," "fastest."

e. Adjectives:

i. Lexical Rules: Adjectives are words used to describe or modify nouns. They encompass regular adjectives (JJ), comparative adjectives (JJR), and superlative adjectives (JJS).

ii. Regular Expression Pattern: r'\b\w+\b'

iii. Examples: "red," "redder," "reddest."

f. Numbers:

i. Lexical Rules: Numbers in English text include cardinal numbers (CD) representing quantities, counts, or numerical values.

ii. Regular Expression Pattern: r'\b\d+\b'

iii. Examples: "123," "5," "1,000."

g. Punctuation:

i. Lexical Rules: Punctuation marks in English text encompass a wide range of characters used for various purposes, including separating words, indicating pauses, and providing structure. These characters include common punctuation marks like periods, commas, semicolons, colons, exclamation marks, question marks, quotation marks, parentheses, and more.
ii. Regular Expression Pattern: r'\b[.,;:!?()"'-]+\b'
iii. Examples: ".", ",", ";", ":", "!", "?", "(", ")", """, "-"

h. Operators:
   i. Lexical Rules: Operators are a subset of symbols used in programming or mathematical operations. These symbols are often customized to suit specific project needs and requirements, and they are identified with the POS tag "SYM."
   ii. Regular Expression Pattern: r"[+,||,\,&&]"
   iii. Examples: "+," "&&," "<=" (customized as needed)

i. Special Characters:
   i. Lexical Rules: Special tokens are distinctive characters or symbols defined by a specific regex pattern, allowing for customization to suit the project's unique requirements. These tokens are identified as "special."
   ii. Regular Expression Pattern: **^[@#$%&*()]$**
   iii. Examples: "@", "#", "$", "%", "&", "*", "(", ")".

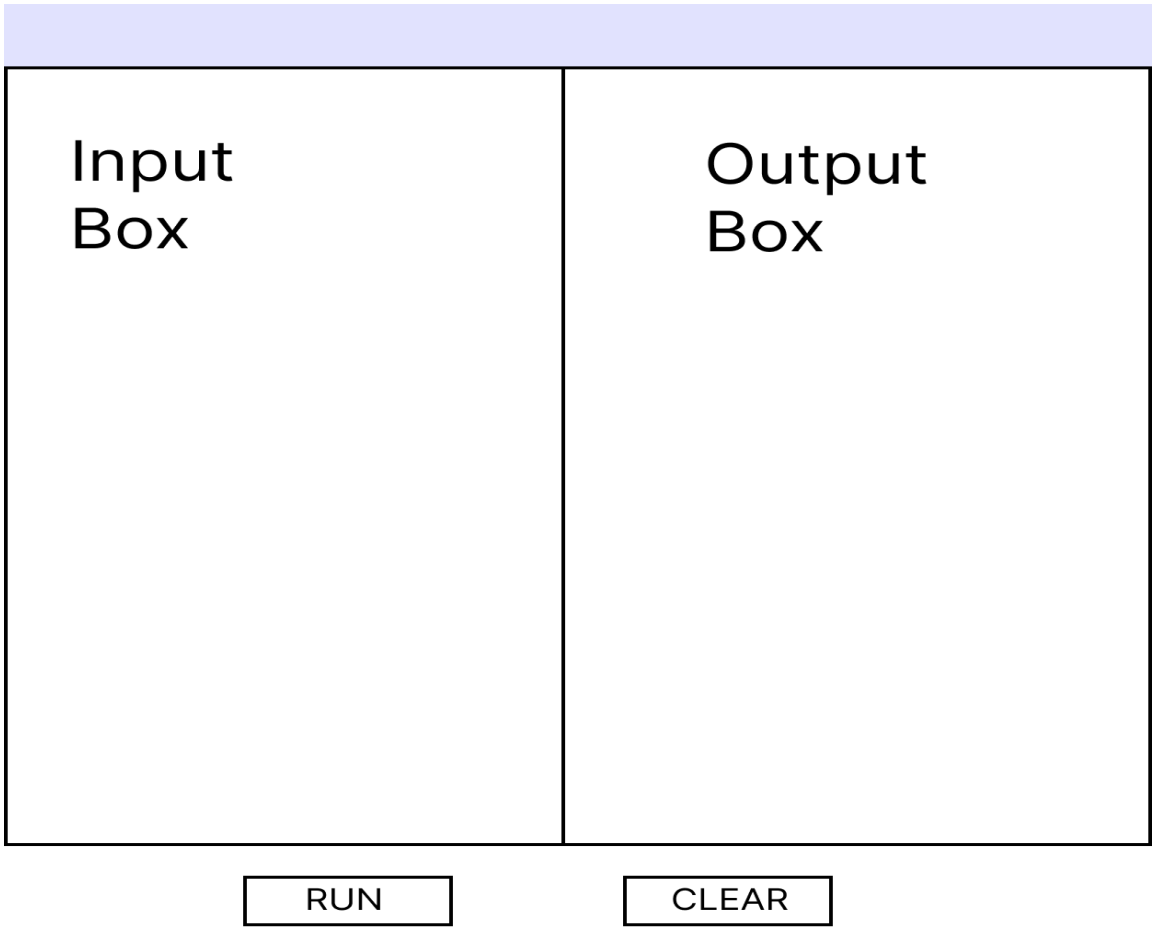
1.5. Result

Token types, including keywords, identifiers, numerals, and operators, are defined in Phase 1 of the project together with their unique lexical rules for processing English text. This critical stage provides a thorough and organized framework for locating and classifying tokens within the text, setting the foundation for the lexical analysis project's next phases.

Token types can be identified and categorized using the well-defined lexical rules, which makes it possible for the next stages to tokenize and classify English language structures in an efficient and organized manner. This careful preparation improves the text's analyzability, making it possible for the lexical analyzer to quickly and precisely identify and classify tokens—a crucial step for a variety of language processing tasks. These guidelines serve as the foundation for the project's

tokenization and classification success, which in turn helps to extract insightful information from unstructured English text data.

1.6. GUI Design



1.7. References

1) Pustejovsky, James, and Branimir Boguraev. "Lexical knowledge representation and natural language processing."
2) Spositto, Osvaldo Mario, Julio César Bossero, Edgardo Javier Moreno, Viviana Alejandra Ledesma, and Lorena Romina Matteo. "Lexical Analysis Using Regular Expressions for Information Retrieval from a Legal Corpus."
3) Contreras, Jennifer O., Melvin A. Ballera, Ace C. Lagman, and Jennalyn G. Raviz. "Lexicon-based Sentiment Analysis with Pattern Matching Application using Regular Expression in Automata."