INTRODUCTION

The String Validator program demonstrates how formal language rules can be applied using a graphical user interface. It validates strings based on a defined alphabet and length condition.

FORMAL LANGUAGE DEFINITION

The language is defined as:
$L = \{ w \mid w \in \{a,b\}^* \text{ and } |w| \text{ is odd} \}$

This means the string must contain only the characters a and b, and its length must be an odd number.

VALIDATION LOGIC

The program first reads the user input. It checks whether all characters belong to the alphabet {a, b}. Next, it checks if the string length is odd. If both conditions are true, the string is accepted; otherwise, it is rejected.

VALIDATION EXAMPLES

Accepted strings include: a, aba, bbbbb
Rejected strings include: ab, abab, abc, empty string

USER INTERFACE STRUCTURE

The interface is built using Tkinter. It contains an input field, a validate button, rule indicators, and a result display that shows acceptance or rejection.

KEY COMPONENTS

The program uses Tk for the main window, Entry for text input, Button for triggering validation, Label for displaying rules and results, and StringVar for dynamic updates.

DESIGN AND FEEDBACK

Green colors indicate valid conditions while red colors indicate errors. This helps users easily understand which rule failed.

EVENT HANDLING

User interaction is handled through button click events and real-time UI updates using callbacks.

SUMMARY

The String Validator demonstrates formal language validation, event-driven programming, and simple GUI design using Python and Tkinter.