**Pattern Matching and Replacing**

**Abstract**

**Pattern matching** is the act of checking a given **sequence** of tokens for the presence of the constituents of some pattern. There are different algorithms. The main goal to design these types of algorithms is to reduce the time complexity. The traditional approach may take a lot of time to complete the pattern searching task for a longer text but with the help of finite automata, it can be achieved in a much shorter time.

**What I aim to do:**

I would like to develop a program which takes and produces the following input and outputs.

**Input:**

* Text (in which pattern has to be searched)
* Pattern (that has to be searched in text)

**Output:**

* Number of occurrences of pattern
* Locations of occurrences of patterns

**Further, program will also allow some** **operations to be performed**:

* Replace the pattern with some other string in the original text.
* Delete the pattern from the original text.

These operations can be performed on all the patterns matched or just on a pattern at a specific location as inputted by the user.

**Flow of program:**

* Firstly, the user will input a pattern.
* A Finite Automata will be constructed which accepts the pattern as a substring.
* Now, User will input some text in which the pattern needs to be searched.
* The text will be passed through the Finite Automata to check for the substring (pattern) presence.
* All the occurrences of the pattern will be displayed to the user
* Now, user will be able to perform operations on the matched patterns (operations such as replace or delete)
* User will also be able to choose whether to perform operation on all the occurrences of the pattern or only on a specific pattern.
* Finally, the modified text would be displayed.