String Pattern Matching with Finite Automata

Esra Eryılmaz 171044046

Tuğçe Karagöz 171044099

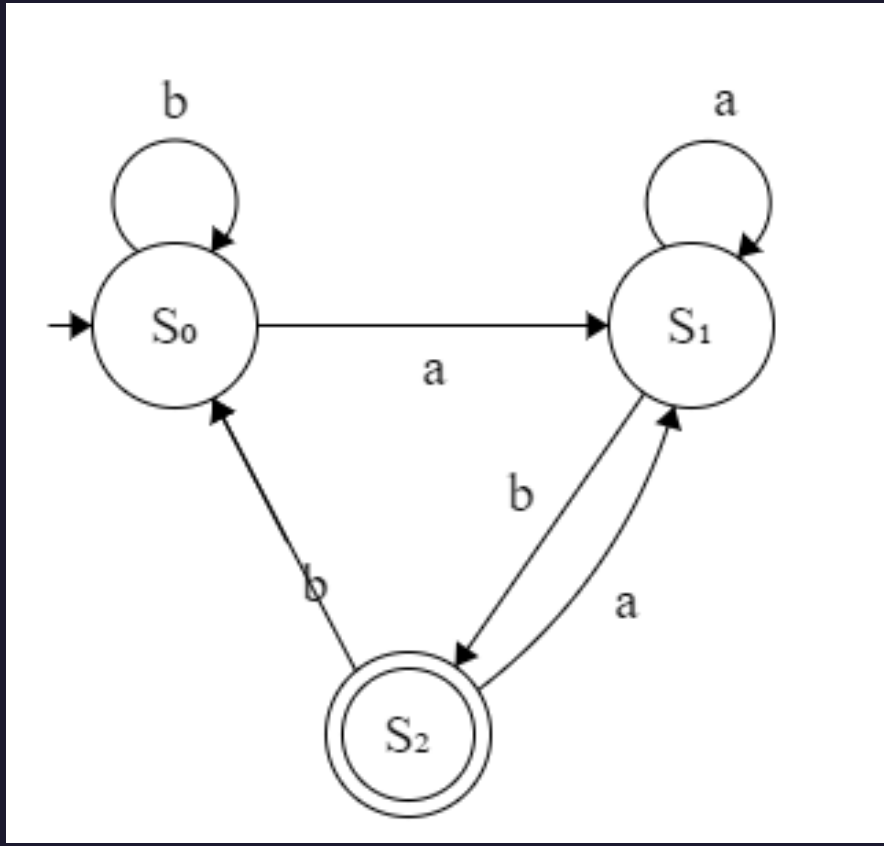
Ayşe Gül Demirbilek 1801042088

Content

- What is Finite Automata?
- What is Pattern Matching?
- How to match string using finite automata? (with example)
- How to generate finite automata for any pattern?
- User Interface
- Output Examples

What is Finite Automata?

- - Finite State Automata or Finite State Machine is the simplest model used in Automata. Finite state automata accepts regular language. In this, the term finite ***means it has a limited number of possible states, and number of alphabets in the strings are finite***.



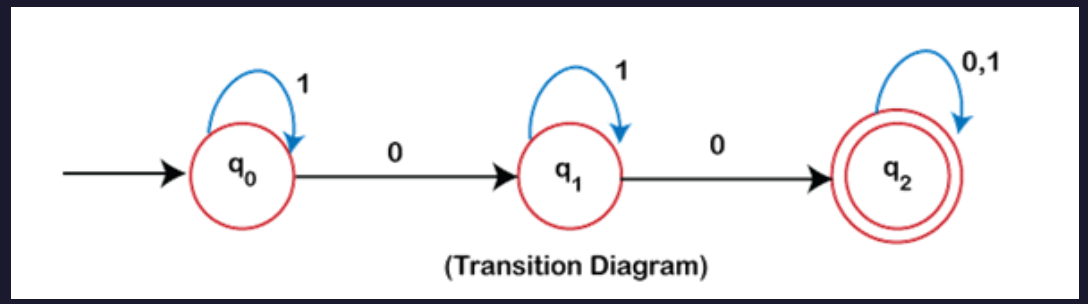
What is Finite Automata?

A finite automaton **M** is a 5-tuple **(Q, q_0, A, $\sum$, δ)** where

- Q is a finite set of **states**,
- $q_0 \in Q$ is the **start state (initial state)**,
- $A \subseteq Q$ is a notable set of **accepting states**,
- $\sum$ is a **finite input alphabet**,
- δ is the **transition function** that gives the next state for a given current state and input.

Representation of Finite Automata

Transition Diagram

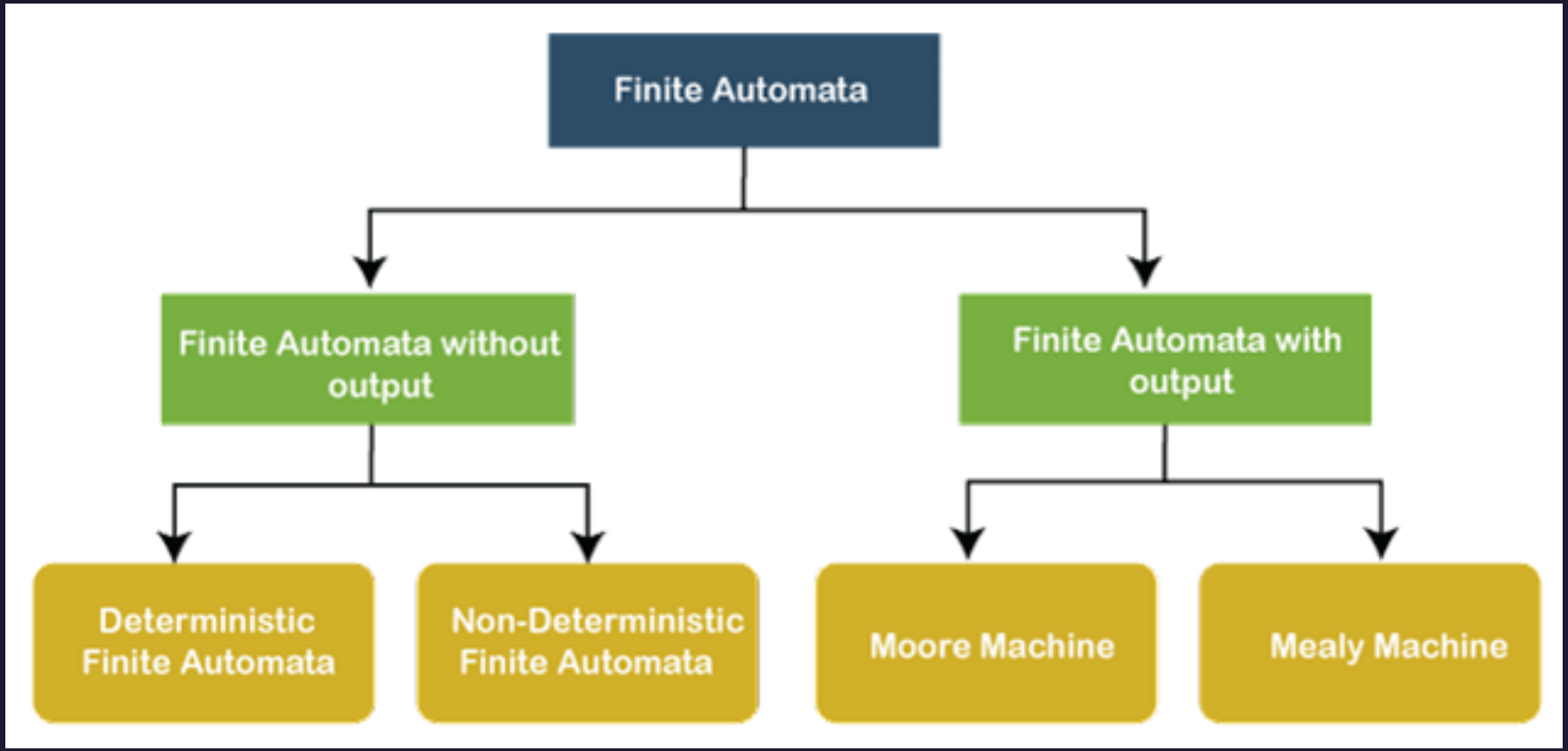


Transition Table

States	INPUT	
	0	1
→ q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_2	q_2

- The initial state is marked with :
- The final state(s) are marked with :

Types of Finite Automata



What is Pattern Matching?

Pattern :

- A collection of strings described in some formal language.

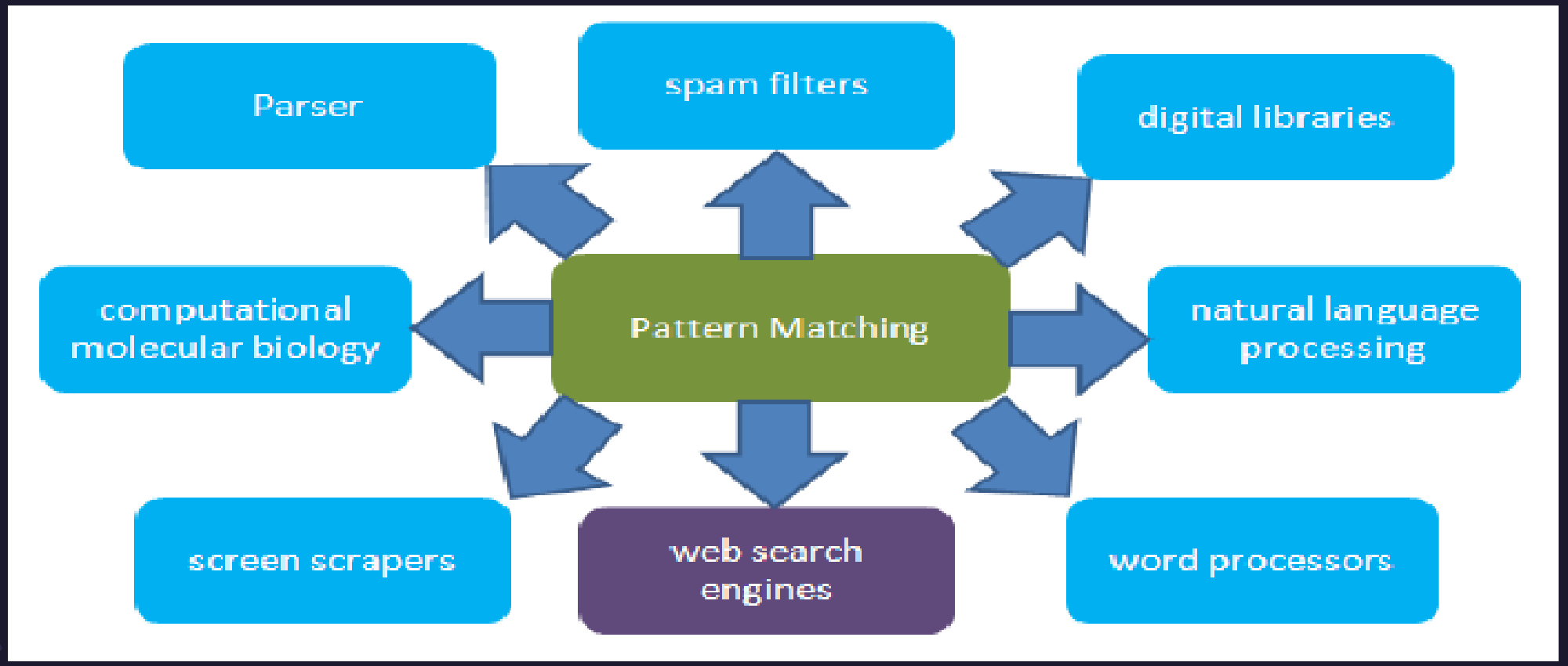
Pattern Matching :

- The problem of locating a specific pattern inside raw data.

Pattern Matching Algorithms :

1) Naive Pattern Searching
2) KMP Algorithm
3) Rabin-Karp Algorithm
4) Finite Automata
5) Boyer Moore Algorithm
6) Aho-Corasick Algorithm
7) Suffix Array
8) Kasai's Algorithm
9) Z algorithm (Linear time pattern searching Algorithm)
10) Manacher's Algorithm
11) Ukkonen's Suffix Tree Construction

Applications of Pattern Matching

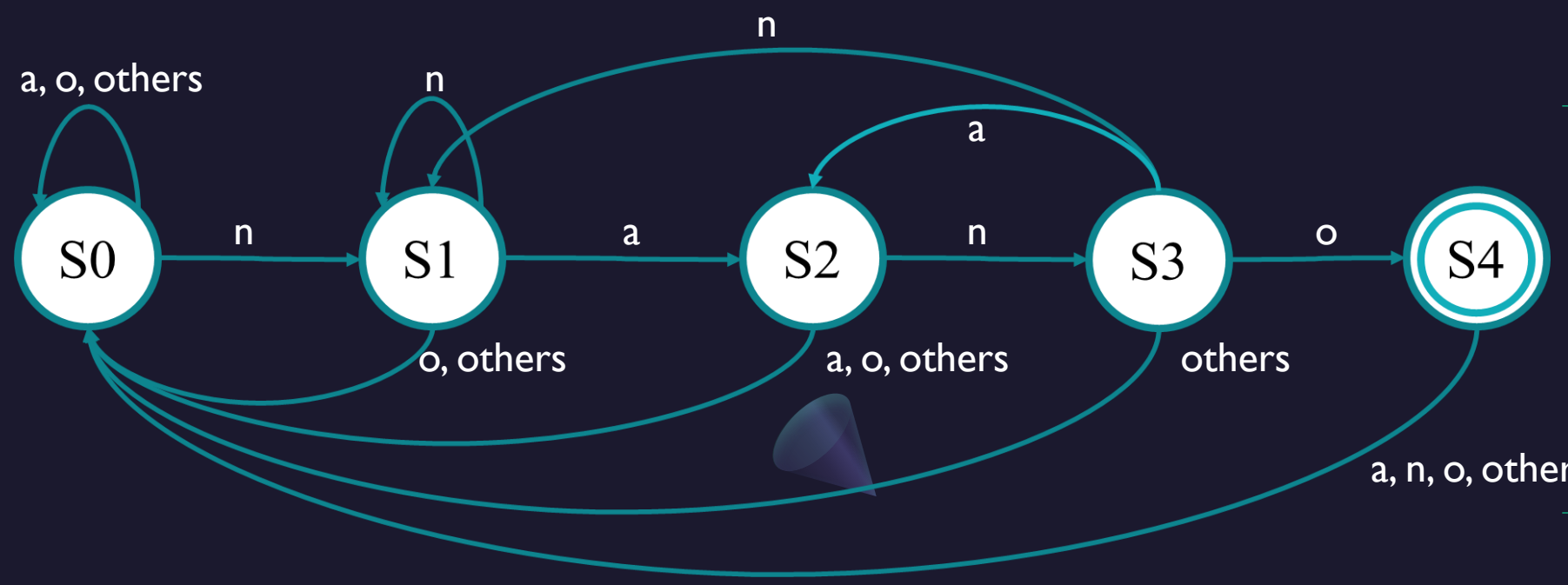


How to match strings?

Text = "bananaona"
Pattern = "nano"

States	n	a	n	o
S0	S1	S0	S0	S0
S1	S1	S2	S0	S0
S2	S3	S0	S0	S0
S3	S1	S2	S4	S0
S4	S0	S0	S0	S0

Transition Table



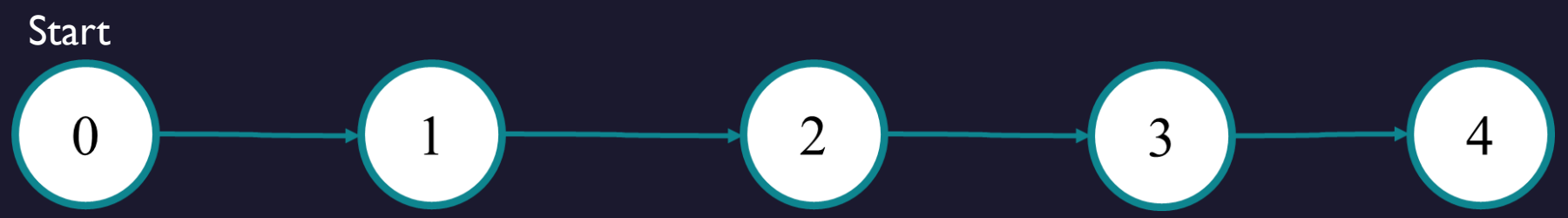
Finite Automata

How to make transition table and finite automata?

- Identify the unique characters from the pattern.

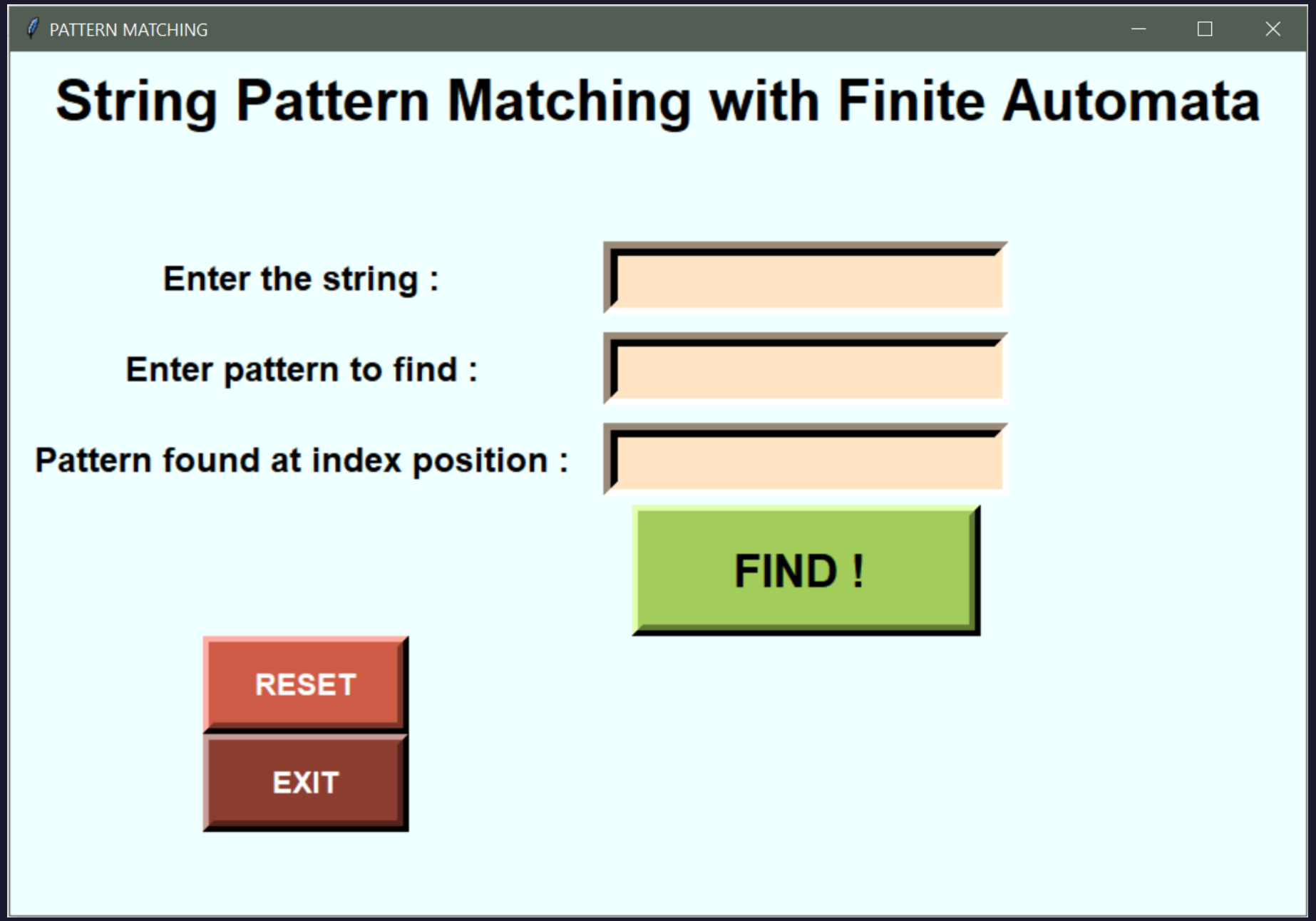
- Transitions table's columns represent those unique characters and other cases, table's rows represent the indexes of the pattern and the start state. So,

Number of columns = numberOfChar(pattern) + 1
Number of rows = length(pattern) + 1

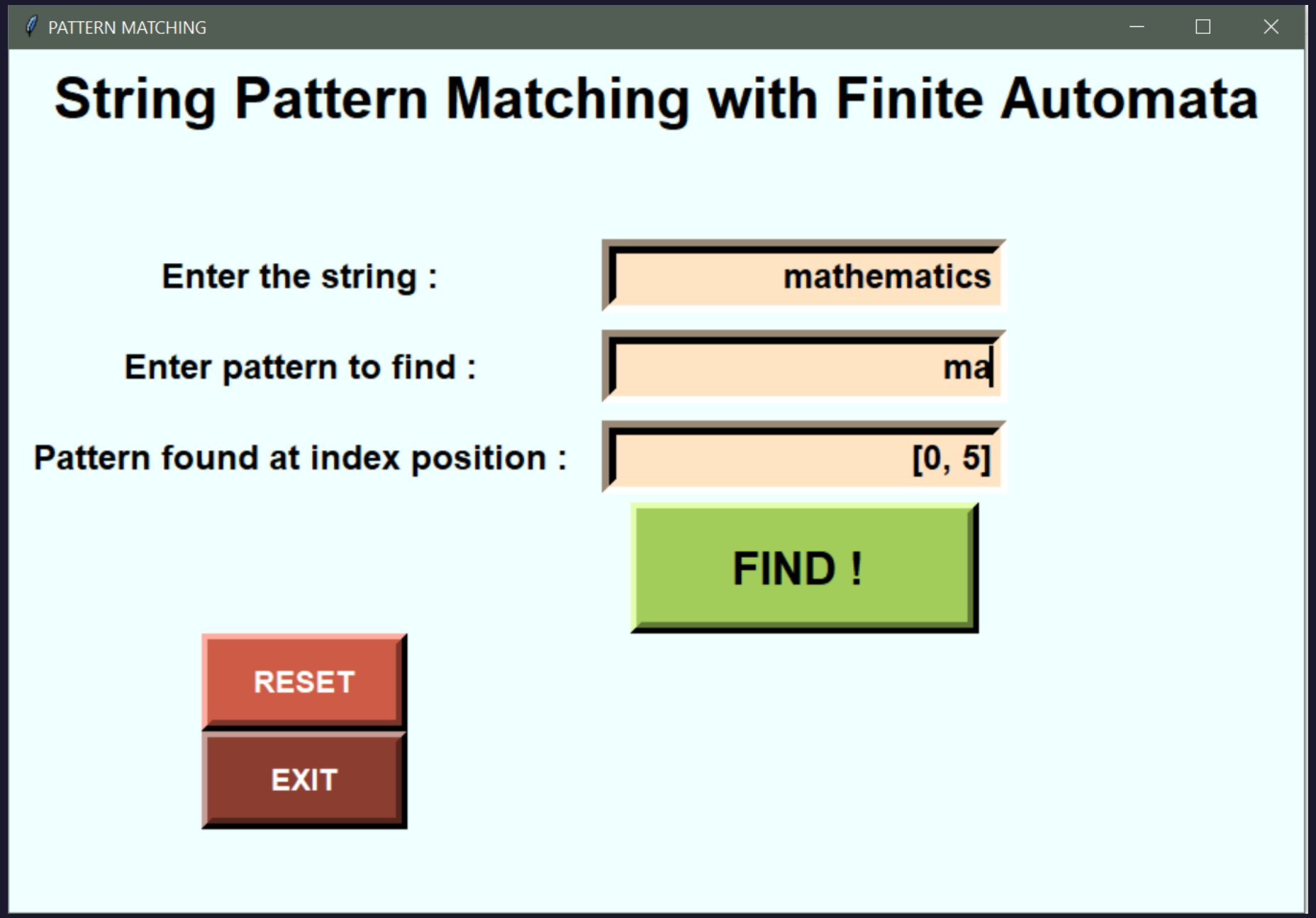


4 char is sequentially matched and reached the last character

User Interface



Output Examples



Thanks for listening