

## Home Work 3

Francisco J. Díaz Riollano  
Student ID: 802-15-2172

September 25, 2018

---

```

1: procedure NAIVE(On input S[1,..m])
2:    $n \leftarrow |T|$ 
3:    $m \leftarrow |S|$ 
4:   for s=0 to n-m do
5:     if P[1,..m] = T[s+1...s+m] then
6:       print("pattern found")

```

---

## Question 1

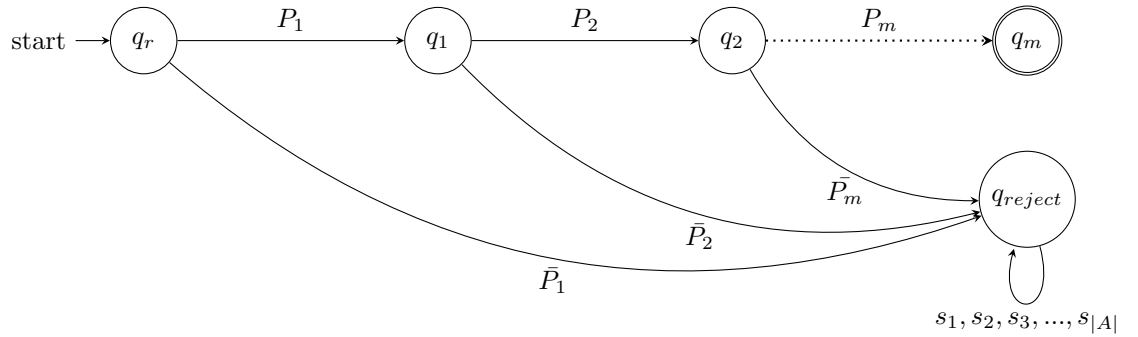
The string matching problem is defined as: "Given a text  $T = T_1...T_n$  which is stored as array  $T = T[1, ..., n]$ , and a pattern  $P = P_1...P_m = P[1...m]$  with  $m < n$ , where both are strings over the same alphabet  $\Sigma$ ; decide whether S is a substring of T.

Algorithm 1 is the so-called naive-pattern finding algorithm. Use Algorithm 1 to construct a Finite State Automata(deterministic or non-deterministic) for solving the matching problem.

Let  $M = (Q, \Sigma, \delta, q_0, F)$  be 5-tuple NFSA where:  
 $Q$  is a finite set of states  
 $\Sigma$  is a finite set of input symbols  
 $\delta$  is a state transition function from  $Q \times \Sigma \rightarrow Q$   
 $q_0 \in Q$  is the initial state  
 $F \subseteq Q$   $Q$  is the set of final states  
Let  $A = \{s_1, s_2, ..., s_{|A|}\}$  and  $\bar{p} = A - p$ .

High Level Description of the automata:

M =: "On input S[1,...,m]"



## Question 2

Algorithm 1 returns a result in the time proportional to  $O(|T||S|)$ . Discuss the computation time of your automaton.

In the worst case the automata will have  $O(|T||S|)$  complexity. We know this because it iterates through the whole text of size  $|T|$  and for each of those

iterations it compares at  $|S|$ , which is the size of the string, individual items. Thus the automata has the same time complexity as the algorithm.

