Home Work 3

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

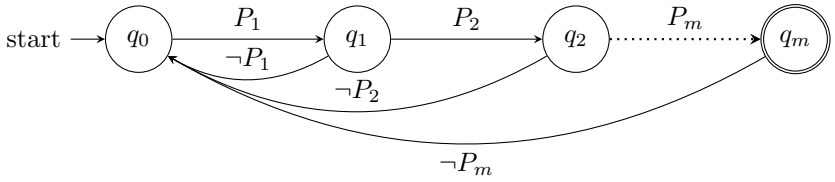
September 24, 2018

```
1: procedure NAIVE(On input S[1,..m])
2:     n ← |T|
3:     m ← |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

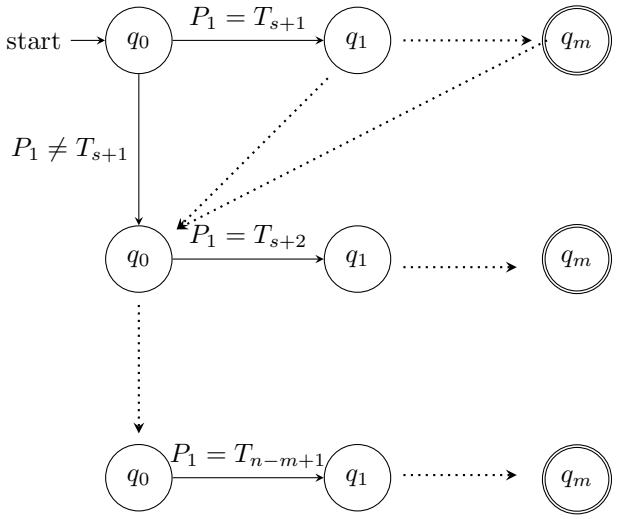
The string matching problem is defined as: "Given a text $T = T_1...T_n$ which is stored as array $T = T[1, ..., n]$, and a pattern $P = P_1...P_m = P[1...m]$ with $m < n$, where both are strings over the same alphabet Σ; decide whether S is a substring of T.

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.



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.


start
q_0
$P_1 = T_{s+1}$
q_1
q_m
$P_1 \neq T_{s+1}$
q_0
$P_1 = T_{s+2}$
q_1
q_m
q_0
$P_1 = T_{n-m+1}$
q_1
q_m