Advanced Algorithm

String Matching

Horspool's Algorithm

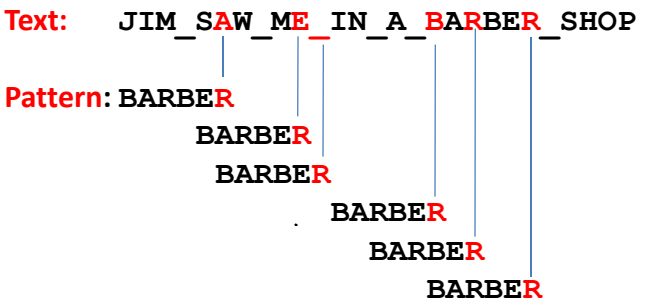
Consider, as an example, searching for the pattern BARBER in some text:

s_0 ... c ... s_{n-1}

B A R B E R

Starting with the last R of the pattern and moving right to left, we compare the corresponding pairs of characters in the pattern and the text. If all the pattern's characters match successfully, a matching substring is found. (Then the search can be either stopped altogether or continued if another occurrence of the same pattern is desired.) If, however, we encounter a mismatch, we need to shift the pattern to the right. Clearly, we would like to make as large a shift as possible without risking the possibility of missing a matching substring in the text. Horspool's algorithm determines the size of such a shift by looking at the character c of the text that was aligned against the last character of the pattern.

Try Yourself (Solution)



Horspool's Algorithm

Algorithm 2.11: Horspool
Input: text $T = T[0 \ldots n)$, pattern $P = P[0 \ldots m)$
Output: position of the first occurrence of P in T
Preprocess:
(1) for $c \in \Sigma$ do $shift[c] \leftarrow m$
(2) for $i \leftarrow 0$ to $m-2$ do $shift[P[i]] \leftarrow m-1-i$
Search:
(3) $j \leftarrow 0$
(4) while $j + m \leq n$ do
(5) if $P[m-1] = T[j+m-1]$ then
(6) $i \leftarrow m-2$
(7) while $i \geq 0$ and $P[i] = T[j+i]$ do $i \leftarrow i-1$
(8) if $i = -1$ then return j
(9) $j \leftarrow j + shift[T[j+m-1]]$
(10) return n

String Matching Using Finite Automata

Example: String ending with ‘0’ (One Accepting State)

- Accepted

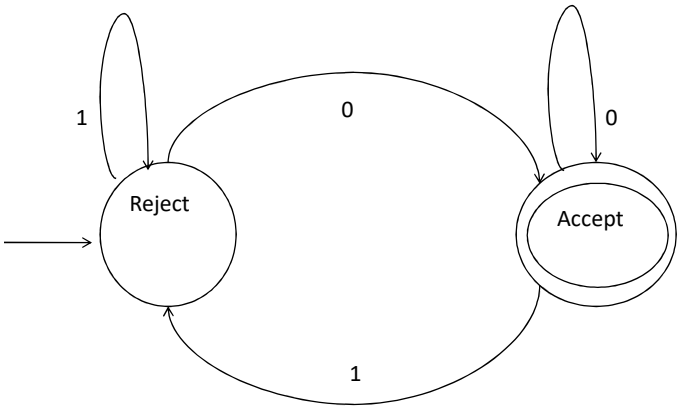
0
00
10
000
010
100
110

- Rejected

Λ
1
01
11
001
011
101
111

How many input symbols are required to change the state from “Accept” to “Reject” / “Reject” to “accept” ?

Example: String ending with ‘0’ (One Accepting State)



Example: String ending with ‘11’ (One Accepting State)

- Accepted

11
011
111

- Rejected

Λ
0
1
00
01
10
000
001
010
100
101
110

How many input symbols are required to change the state from “Accept” to “Reject” / “Reject” to “accept” ?

Example: String ending with ‘11’ (One Accepting State)

- Accepted

11
011
111

- Rejected

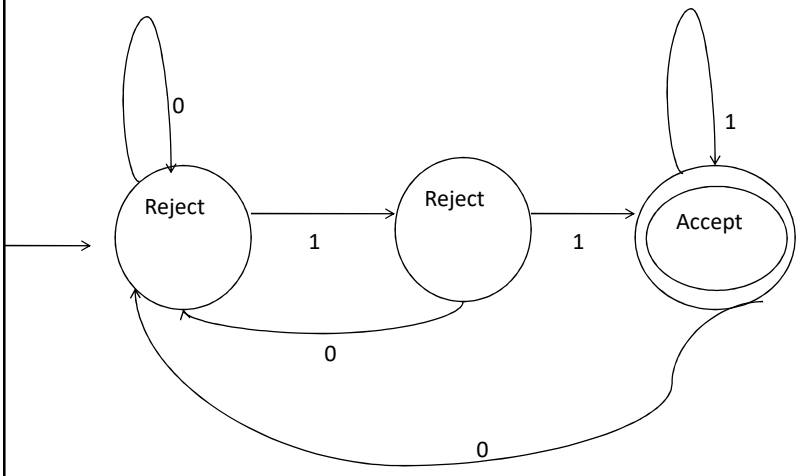
1
01
001
101

- Rejected

Λ
0
00
10
000
010
100
110

How many input symbols are required to change the state from “Accept” to “Reject” / “Reject” to “accept” ?

Example: String ending with ‘11’ (One Accepting State)



Example: ’00’ is not a substring

- Accepted

^
0
1
01
10
11
011
010
101
110
111

- Rejected

00
000
001
100

How many input symbols are required to change the state from “Accept” to “Reject” / “Reject” to “accept” ?

Example: ’00’ is not a substring (Two Accepting States)

- Accepted

^
1
01
11
011
101
111

- Accepted

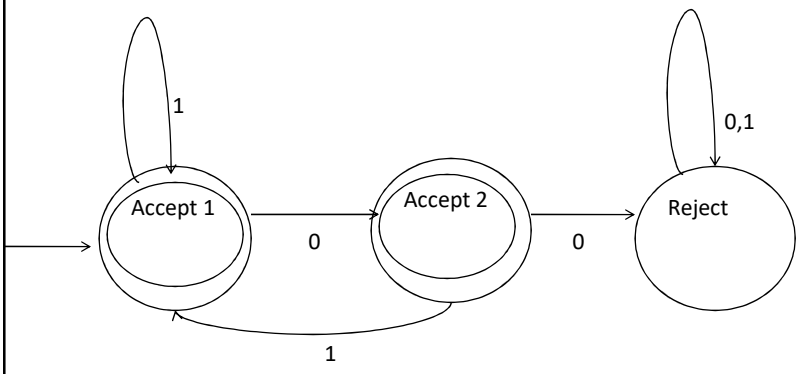
0
10
010
110

- Rejected

00
000
001
100

How many input symbols are required to change the state from “Accept” to “Reject” / “Reject” to “accept” ?

Example: ’00’ is not a substring (Two Accepting States)



Example: String ending with ‘1’ and doesn’t contain 00 (One Accepting State)

- Accepted

1
01
11
011
101
111

- Rejected

Λ
0
00
10
000
001
010
100
110

How many input symbols are required to change the state from “Accept” to “Reject” / “Reject” to “accept” ?

Example: String ending with '1' and doesn't contain 00 (One Accepting State)

- Accept

1
01
11
011
101
111

- Reject

Λ
0
10
010
110

- Reject

100

- Never Accepted

00
000
001

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: $\{\alpha \in \Sigma^* | \alpha$ is a binary number divisible by $4\}$ (One Accepting State)

(0+1)*00

Try Yourself

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?