CHAPTER 4

String Matching Algorithm

string-matching algorithms, sometimes called **string-searching algorithms**, are an important class of string algorithms that try to find a place where one or several strings (also called patterns) are found within a larger string or text.
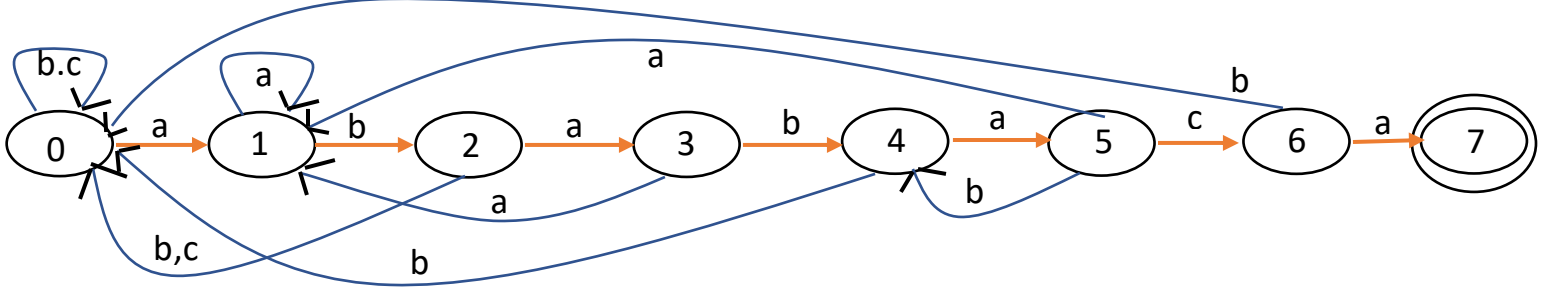
T → Text

    aabbcdabbcay

P → bca   : Pattern

Algorithm

    n ← length(T)

    m ← length(P)

    for (s = 0; s <= (n-m); s++)

        if (p[1..m] = T[s+1 … s+m])

            print " pattern found"

        end if

    end for

p = a b a b a c a

Transition Table

State	a	b	c
0	1	0	0
1	1	2	0
2	3	0	0
3	1	4	0
4	5	0	0
5	1	4	6
6	7	0	0
7	1/1	2/0	0/0

With Overlap   without Overlap

Find_Shift_Key (T, R, P)

    // T: text,   R: Transition Table,   P: Pattern

    n ← length(T)

    q ← 0   // state number

    for (i = 0; i <= n; i++)

        q = R [q , T [i+1]);

        if (q == m)   // m: Final state

            s = i – m ;

            print " Pattern is at position ",s

        end if

    end for

Example:

T = a b a c a b a a c b a b a b a c a b a …

P = a b a b a c a

q = 0

q = R [0, a] = 1

q = R [1, b] = 2

q = R [2, a] = 3

q = R [3, c] = 0

q = R [0, a] = 1

q = R [1, b] = 2

q = R [2, a] = 3

q = R [3, a] = 1

q = R [1, c] = 0

q = R [0, b] = 0

q = R [0, a] = 1

q = R [1, b] = 2

q = R [2, a] = 3

q = R [3, b] = 4

q = R [4, a] = 5

q = R [5, c] = 6

q = R [6, a] = 7

q = R [7, b] = 2

q = R [2, a] = 3

State	a	b	c
0	1	0	0
1	1	2	0
2	3	0	0
3	1	4	0
4	5	0	0
5	1	4	6
6	7	0	0
7	1/1	2/0	0/0