# Construct DFA For KMP

# KMP

- whenever we detect a mismatch, we already know some of the characters in the text (since they matched the pattern characters prior to the mismatch).
- We can take advantage of this information to avoid backing up the text pointer over all those known characters.
- How?
- Using DFA to record how far to back up the pattern pointer

# DFA for Pattern

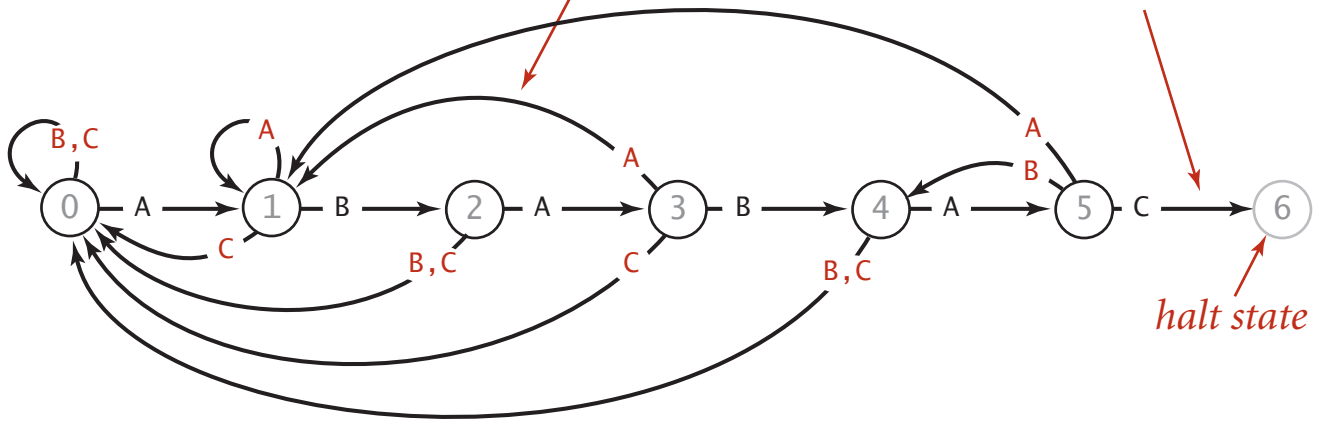
**internal representation**

| | j | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| pat.charAt(j) | | A | B | A | B | A | C |
| dfa[][j] | A | 1 | 1 | 3 | 1 | 5 | 1 |
| | B | 0 | 2 | 0 | 4 | 0 | 4 |
| | C | 0 | 0 | 0 | 0 | 0 | 6 |

*mismatch transition (back up)*

*match transition (increment)*

**graphical representation**




**DFA corresponding to the string A B A B A C**

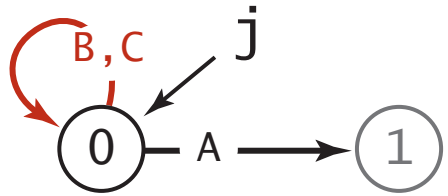
# Implementation

```
public KMP(String pat) {
    this.R = 256;
    this.pat = pat;

    // build DFA from pattern
    int m = pat.length();
    dfa = new int[R][m];
    dfa[pat.charAt(0)][0] = 1;
    for (int x = 0, j = 1; j < m; j++) {
        for (int c = 0; c < R; c++)
            dfa[c][j] = dfa[c][x];     // Copy mismatch cases.
        dfa[pat.charAt(j)][j] = j+1;   // Set match case.
        x = dfa[pat.charAt(j)][x];     // Update restart state.
    }
}
```

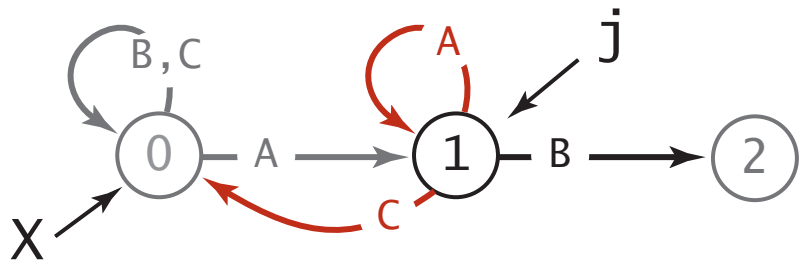
# Procedure

| | | |
|---|---|---|
| | j | 0 |
| pat.charAt(j) | | A |
| dfa[][j] | A | 1 |
| | B | 0 |
| | C | 0 |



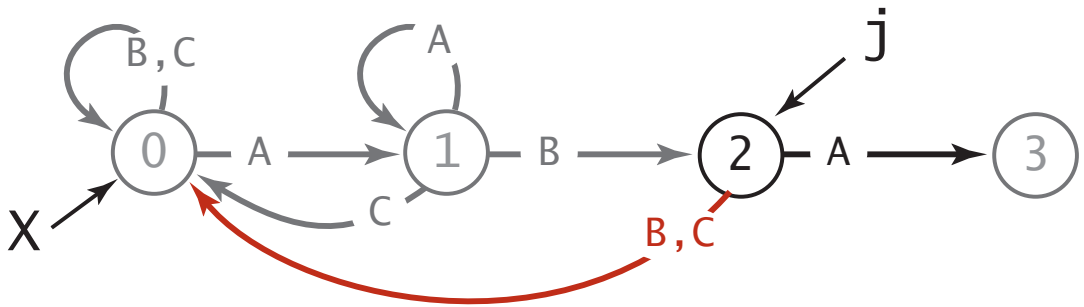
# Procedure

| | | X | |
|---|---|---|---|
| | j | 0 | 1 |
| pat.charAt(j) | | A | B |
| dfa[][j] | A | 1 | 1 |
| | B | 0 | 2 |
| | C | 0 | 0 |



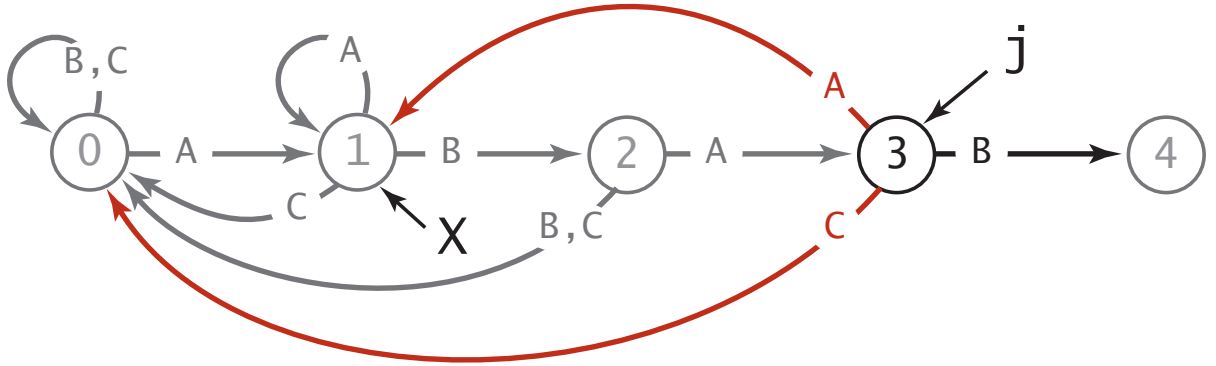
# Procedure

| | | X | | |
|---|---|---|---|---|
| | j | 0 | 1 | 2 |
| pat.charAt(j) | | A | B | A |
| dfa[][j] | A | 1 | 1 | 3 |
| | B | 0 | 2 | 0 |
| | C | 0 | 0 | 0 |



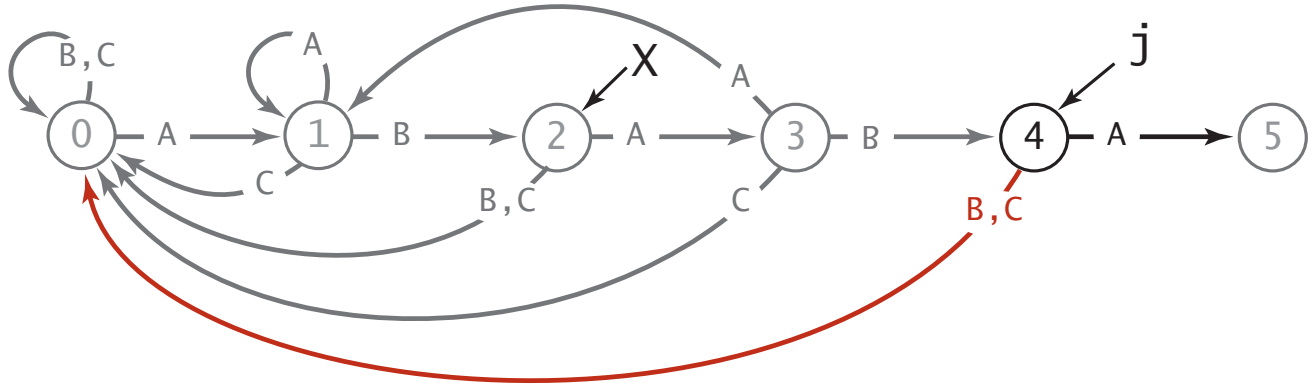
# Procedure

| | | X | | |
|---|---|---|---|---|
| j | 0 | 1 | 2 | 3 |
| pat.charAt(j) | A | B | A | B |
| dfa[][j] A | 1 | 1 | 3 | 1 |
| dfa[][j] B | 0 | 2 | 0 | 4 |
| dfa[][j] C | 0 | 0 | 0 | 0 |



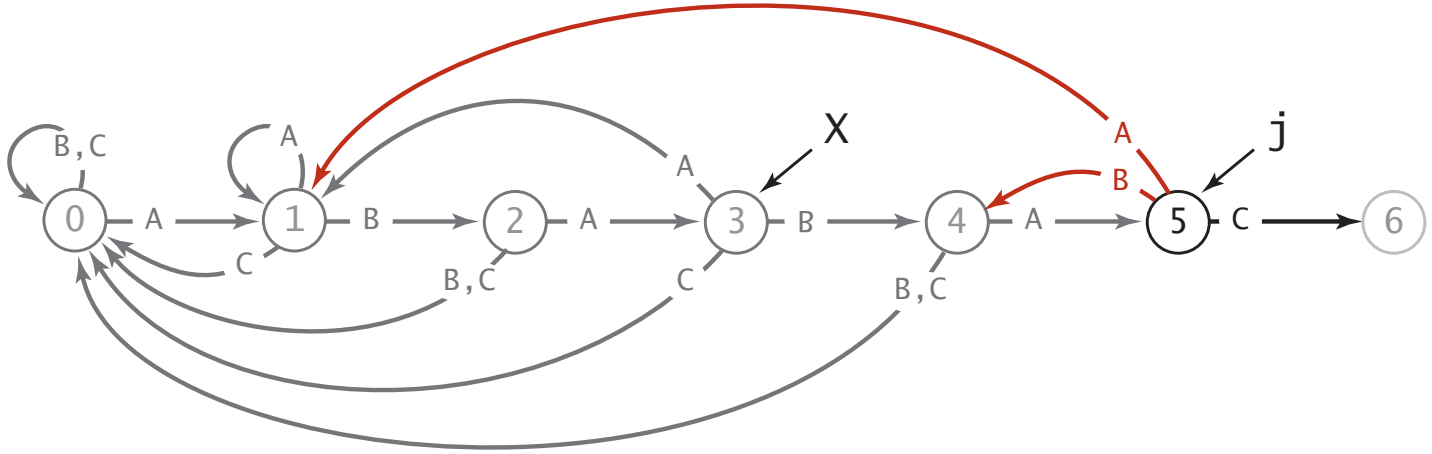
# Procedure

| | j | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | pat.charAt(j) | A | B | A | B | A |
| dfa[][j] | A | 1 | 1 | 3 | 1 | 5 |
| | B | 0 | 2 | 0 | 4 | 0 |
| | C | 0 | 0 | 0 | 0 | 0 |



# Procedure

| | j | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| | pat.charAt(j) | A | B | A | B | A | C |
| dfa[][j] | A | 1 | 1 | 3 | 1 | 5 | 1 |
| | B | 0 | 2 | 0 | 4 | 0 | 4 |
| | C | 0 | 0 | 0 | 0 | 0 | 6 |



# Another Example

- AAABAAC

1 2 3 3 5 6 3
0 0 0 4 0 0 0
0 0 0 0 0 0 7