

• 关于 DFA 表达式和串 X 的匹配.



X 是匹配的 因为第 j 个字符相等.

当 j = i 时失败. 实际上 pat[0:i] 与 string[i:j] 匹配了 pat[0, j-1] 位置.

brute-force 的做法是先从(0,0)尝试后移一位. (int).

如果使用 pat[0] 与 string 的 i-1 位. 然后将 pat[1, j-1].

逆序 == pat[1]. 然而当 j-1 位都相等时. 即 pat[1, j-1] == DFA[pat[j-1]] 时. 从第 j 位到第 0 位.

此时从 j-1 位开始重走 pat[1, j-1] 位 - 位 (对称算法 - lecture note p-29). 这个重走有 j-1 位. 其时. 可以将 pat X 逆序. 由左到右走 DFA 从 0 到 j-1 位的重走. 从而逆向走 DFA 从 j-1 到 0 位.

即 $X = pat[1, j-1]$ 和 DFA 从 0 到 j-1 位的重走. 从而逆向走 DFA 从 j-1 到 0 位.

X° 重走. 2 $X^{\circ} = pat[1, j-2]$. $X = DFA[pat[j-1]](X)$.

X° 重走. 2 $X^{\circ} = pat[1, j-2]$. $X = DFA[pat[j-1]](X)$

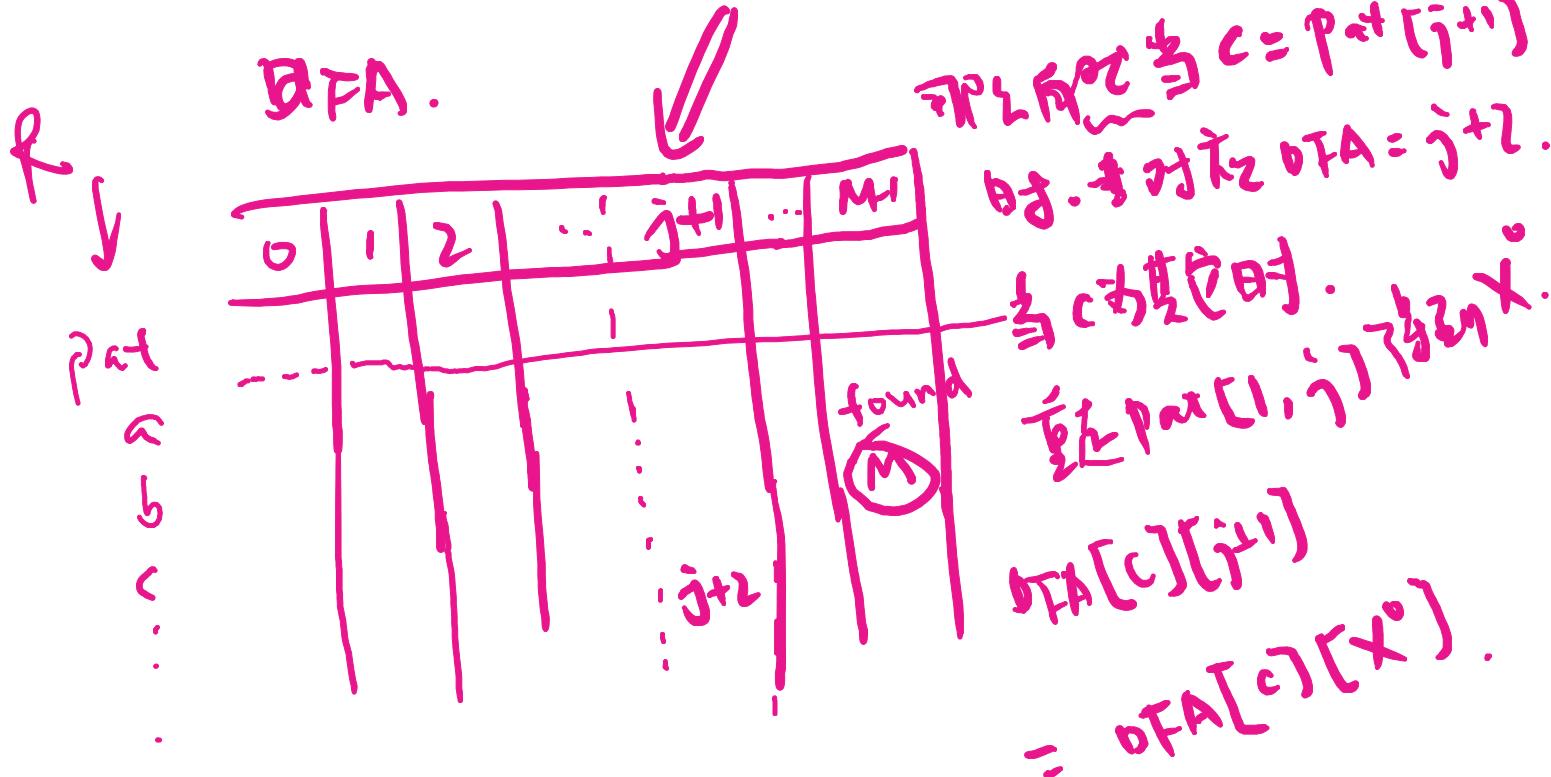
同理. 第 j 位的重走 $X^{\circ} = DFA[pat[j]](X)$

该思路又取到了字符串 search b.

X 是 string 的重走. 从 j-1 位到 0 位的重走.

先从中间. 即 j 位开始重走. 然后从 j+1 位开始重走.

注意. j+1 位下是 looking for pat[j+1].



\Rightarrow 這是：
不是說前 $j+1$ 個字元是合法的，而是說第 $j+1$ 個字元
是非法的。這就是 $\text{pat}[1, j+1]$ 被判斷為非法。(根據前面
已經判斷出 $\text{pat}[0, j]$ 是合法的)。
DFA // the DFA is good {
 match($\text{pat}[0, j]$)
 dismatch($\text{pat}[j+1]$)
 + char C
}

\star DFA 亂的，不是 string 的亂。有區別！
到 M 時 for 會退出，不是 found。
二階搜尋思想