Lecture 6

Two-Dimensional Pattern Matching

Problem: Given pattern $P[1..m, 1..m]$ and text $T[1..n, 1..n]$, find all occurrences of P in T.

- Symbols from alphabet Σ.
- Can be generalized to rectangular arrays.
- Applications: computer vision, multimedia systems where 2D images are stored in a database.

Baker-Bird Algorithm

Text position (i, j) is an occurrence if
$P[1..m, 1..m] = T[i - m + 1..i, j - m + 1..j]$.

1. Row-matching step

Consider P as a set of its rows.

Example

p_1	`aabba`
p_2	`aaabb`
p_3	`ababa`
p_4	`aabba`
p_5	`aaabb`

Give a number to each distinct row:

- $r(p_1) = r(p_4) = 1$
- $r(p_2) = r(p_5) = 2$
- $r(p_3) = 3$

Build an $n \times n$ array R: $R[i, j]$ is $r(p_k)$ such that $T[i, j - m + 1..j] = p_k$; 0 if there is no such p_k.

Example: $R[i, j]$ is one of $0, 1, 2, 3$.

How to compute R: Aho-Corasick

- Build the data structure (tree + failure function) with the rows of P. Time $O(|\Sigma|m^2)$.
- For each row of T, run the AC search algorithm. Time $O(n)$ for each row; overall $O(n^2)$.

2. Column-matching step

Given R, find all occurrence positions.
We need to check if all rows of P appear vertically.

1. Let $P' = r(p_1)r(p_2)\cdots r(p_m)$. Example: $P' = 12312$.
2. For each column of R, run the KMP algorithm with P' as the pattern.

- Time: KMP preprocessing $O(m)$. $O(n)$ for each column of R; overall $O(n^2)$. The total time: $O(|\Sigma|m^2 + n^2)$. (my paper – $O(m^2 + n^2)$ like KMP)
- Extra space: $O(|\Sigma|m^2)$ for AC, $O(m)$ for KMP. $O(n^2)$ for R, but we can use only $O(n)$: Compute R row by row. At the same time, run n KMP algorithms, one for each column of R. Interleave computing R and running KMP (if $R[i, j]$ is computed, run j-th KMP one more step). Thus extra space is $O(|\Sigma|m^2 + n)$.

Example text:

```
T: aabbaaabba        R: 0000100021
   aaabbaaabb           0000210002
   ababaababa           0000300003
   aabbaaabba           0000100021
   aaabbaaabb           0000210002
   baaabbabab           0000021000
   aababaabba           0000030001
   aaabbaaabb           0000210002
   baaabbaaab           0000021000
```