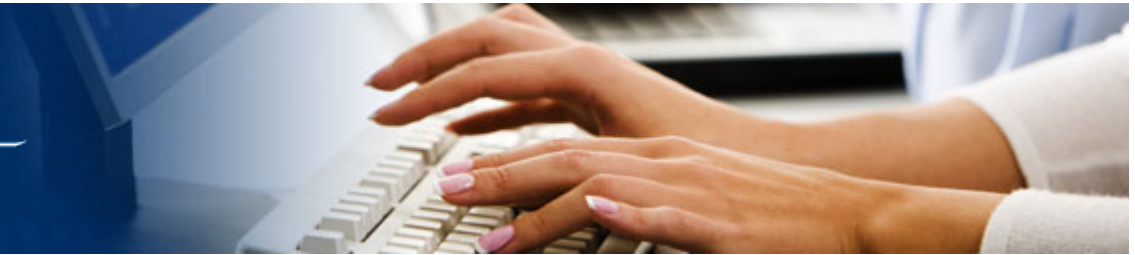




TARTU ÜLIKOOOL

ARVUTITEADUSE INSTITUUT



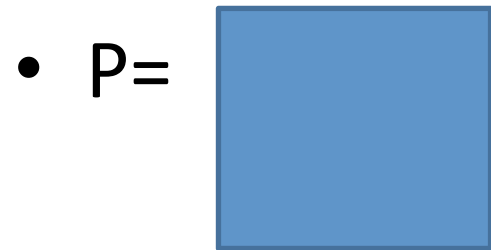
# Text Algorithms (6EAP)

## 2-D pattern matching

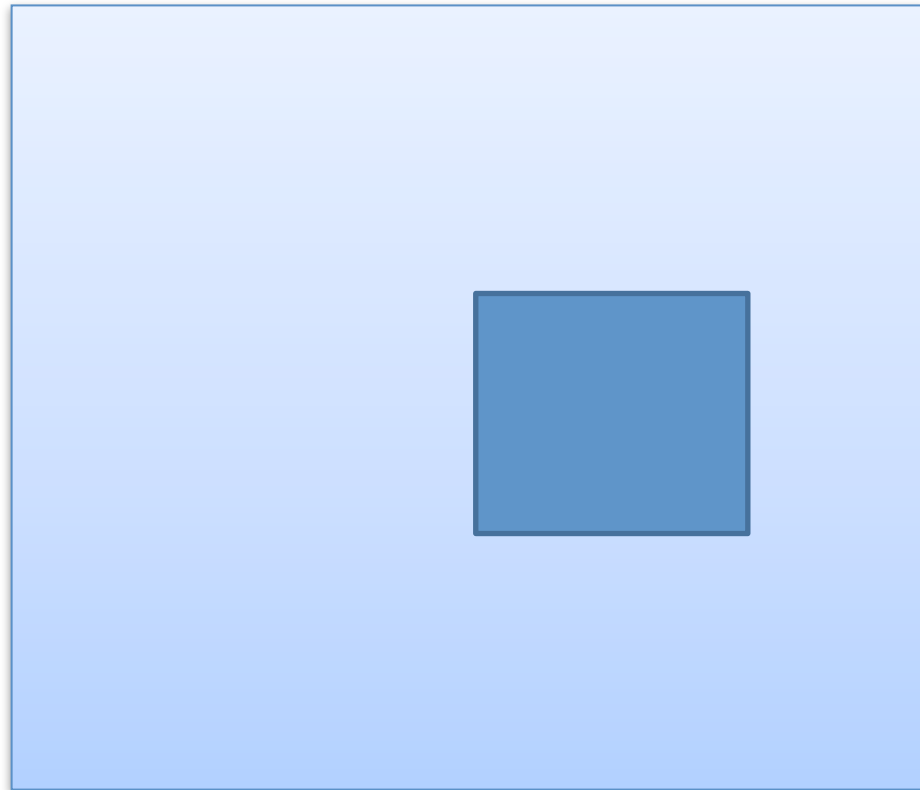
Jaak Vilo

2010 fall

# 2-D



$S =$



- Why?
  - Images
  - ...

# Brute force

**Input:** Pattern P of shape  $m_1 \times m_2$

Text S of shape  $n_1 \times n_2$

**Output:** Occurrences of P in S

```
for i=1..(n1-m1+1) // X-axis in S
```

```
Sp: for j=1..(n2-m2+1) // Y-axis in S
```

```
    for pi=1..m1
```

```
        for pj=1..m2
```

```
            goto next Sp if S[i+pi-1,j+pj-1] != P[pi,pj]
```

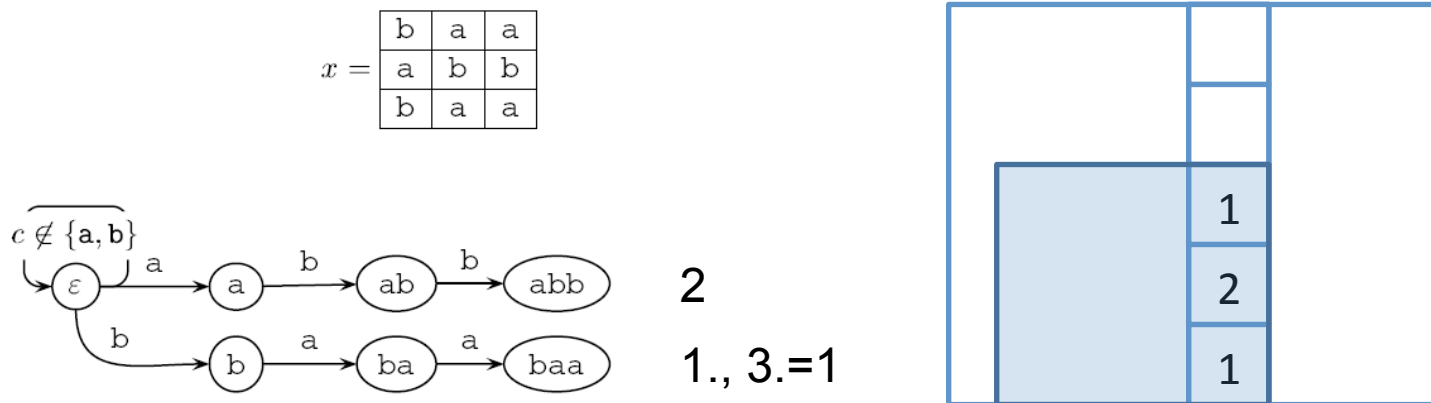
```
            output (i,j) // All positions of P have matched!
```

# How can we do better?

- What did we learn from 1-D matching?

# Bird (1977) and Baker (1978)

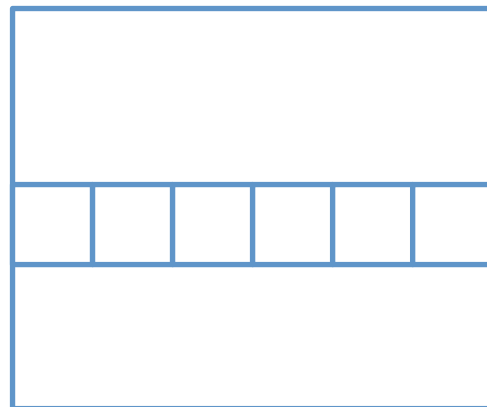
- Search for all rows of P using Aho-Corasick.
- Mark all end positions in the nxn table
- Search with KMP if some column matches the right rows



- Bird, R.S. 1977. Two-dimensional pattern matching. Inf. Process. Lett. 6(5):168 170.
- Baker, T.P. 1978. A technique for extending rapid exact-match string matching to arrays of more than one dimension. SIAM J. Comput. 7(4):533 541.
- [HANDBOOK OF COMPUTER SCIENCE AND ENGINEERING Chapter 6 Pattern matching and text compression algorithms](#), pp. 22-24.

# Less space

- $1 \times n$  table for matching in columns is enough
- for every column the vertical (from top to bottom) matching with KMP is done.
- Memorize  $n$  states of the same KMP



```

B ( $Y, n_1, n_2, X, m_1, m_2$ )
  /* Preprocessing */
1  for  $i \leftarrow 0$  to  $n_2 - 1$ 
2    do  $a[i] \leftarrow 0$ 
3   $root \leftarrow$  PRE-AC (set of lines of  $X, m_1$ )
4  PRE-KMP-FOR-B ( $X, m_1, next$ )
  /* Searching */
5  for  $row \leftarrow 0$  to  $n_1 - 1$ 
6    do  $r \leftarrow root$ 
7      for  $column \leftarrow 0$  to  $n_2 - 1$ 
8        do while  $child(r, Y[row, column]) = \text{UNDEFINED}$ 
9          do  $r \leftarrow fail(r)$ 
10          $r \leftarrow child(r, Y[row, column])$ 
11         if  $out(r) \neq \emptyset$ 
12           then  $k \leftarrow a[column]$ 
13             while  $k > 0$  and  $X[k, 0 \dots m_2 - 1] = out(r)$ 
14               do  $k \leftarrow next[k]$ 
15              $a[column] \leftarrow k + 1$ 
16             if  $a[column] = m_1$ 
17               then OUTPUT ( $row - m_1 + 1, column - m_2 + 1$ )
18             else  $a[column] \leftarrow 0$ 

```

Figure 6.25: The Bird/Baker two-dimensional pattern matching algorithm.

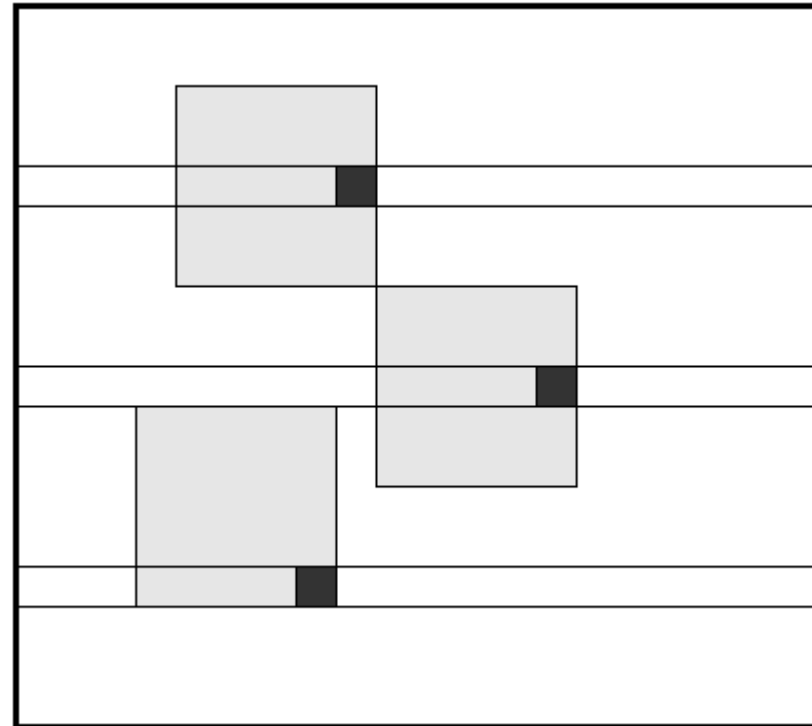
# Zhu & Takaoka (1989)

- Karp-Rabin by rows
- KMP or Boyer-Moore by columns
- Zhu, R.F., Takaoka, T. 1989. A technique for two-dimensional pattern matching. Comm. ACM. 32(9):1110 1120.
- <http://knight.cis.temple.edu/~vasilis/Courses/CIS750/Papers/p1110-zhu.pdf>



# Baeza-Yates and Regnier (1992)

- Similar to Boyer-Moore
- Match every m'th row using Commentz-Walter
- If a potential hit, then check

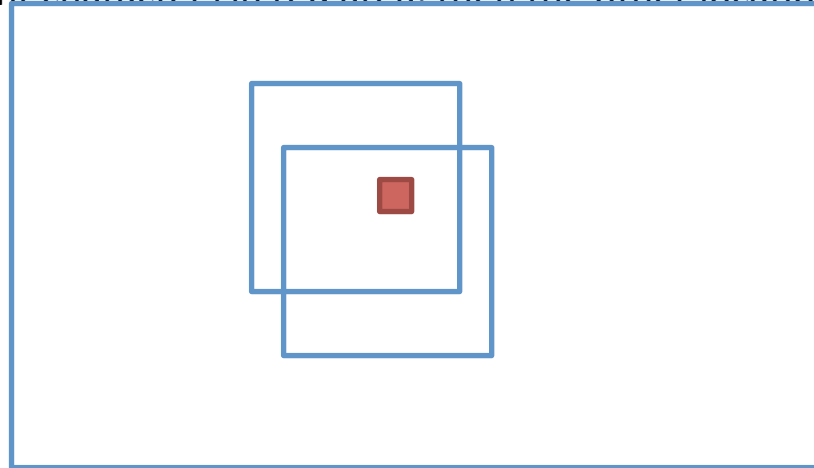


# Elimination

- At first all positions are possible for P (candidates)
- Compare (some positions) with S. If mismatch, one can eliminate that candidate
- Comparisons are made as a duel - at least one of the candidates will be eliminated
- For remaining candidates compare exactly
- $O(m^2 + n^2)$  independent of alphabet size
- Dueling was first used by Vishkin for efficient parallel string matching algorithms. The idea is to provide, in constant time, a method that eliminates one of two competing candidates for pattern occurrence. **This elimination is based on identifying locations where the two candidates expect conflicting symbols.** Vishkin used string periodicity properties to guarantee that such locations exist for every two overlapping candidates.

# Duels and witnesses

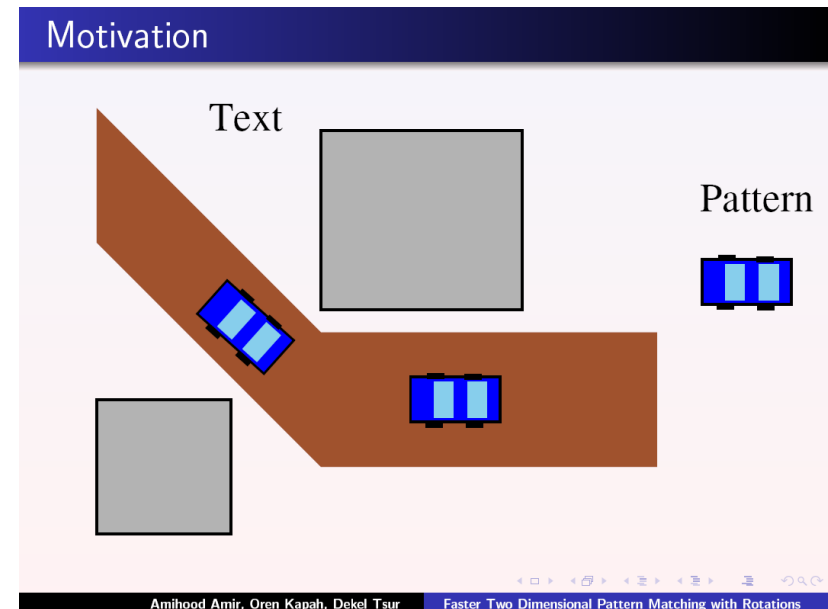
- Truly alphabet-independent two-dimensional pattern matching  
Galil, Z. Park, K.
- This paper appears in:  
[Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on](#) 24-27  
Oct 1992, page(s): 247-256  
doi: 10.1109/SFCS.1992.267767
- Duel and witnesses : eliminate impossible locations!
- If P has two different characters in red area, then the other location can be eliminated!



- **Juha Kärkkäinen and Esko Ukkonen:** Two and higher dimensional pattern matching in optimal expected time. In Daniel D. Sleator, editor, Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 715--723, Arlington, VA, January 1994. ACM Press.  
<http://citeseer.ist.psu.edu/article/karkkainen94two.html>  
[L3\\_2D\\_matching/Karkkainen\\_Ukkonen\\_p715-karkkainen.pdf](http://citeseer.ist.psu.edu/article/karkkainen94two.html)
- **Jorma Tarhio 1993:** Boyer-Moore-Horspooli generalization for 2D

# Variants

- Free form patterns
- Rotated patterns
- Scaled patterns



- **Amihood Amir, Oren Kapah, Dekel Tsur:** Faster two dimensional pattern matching with rotations. Proc. 15th Symposium on Combinatorial Pattern Matching (CPM '04), LNCS 3109, 409-419, 2004.
- <http://cs.haifa.ac.il/~dekelts/publications/rotation.ps> local
- [PDF slide show](#) local