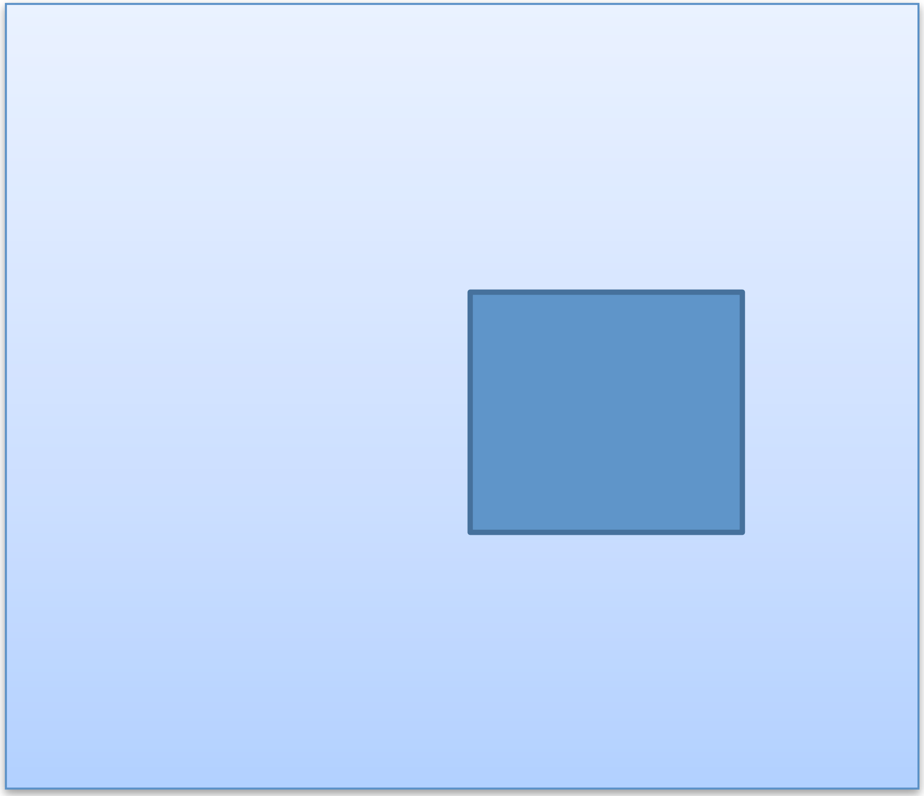
Text Algorithms (6EAP)
2-D pattern matching

Jaak Vilo

2012 fall

2-D

- P=  S= 

- Why?
  - Images
  - ...

Brute force

Input: Pattern P of shape m1 x m2
Text S of shape n1 x n2

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?

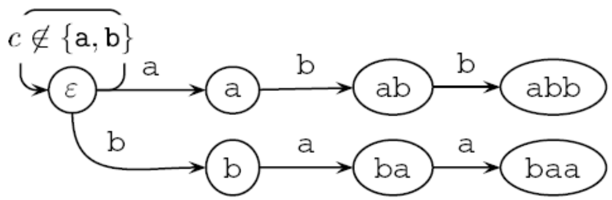
- What did we learn from 1-D matching?

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

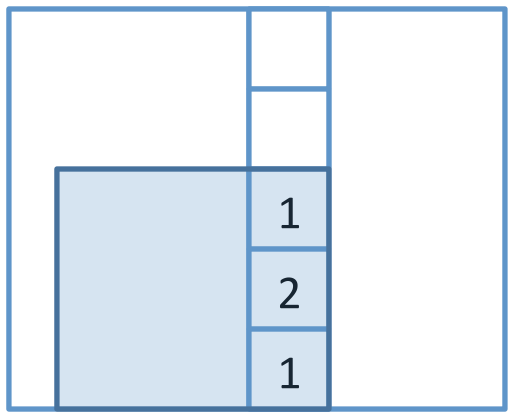
$x =$

b	a	a
a	b	b
b	a	a



2

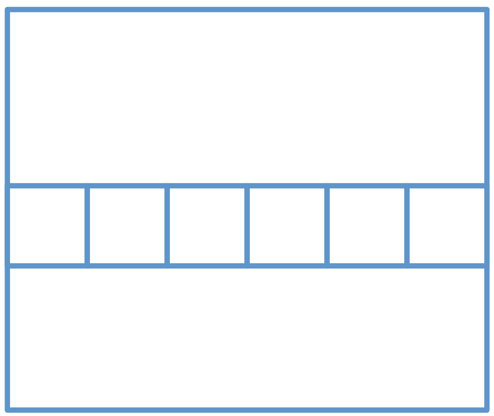
1., 3.=1



- 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

- 1xn 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, n1, n2, X, m1, m2)
    /* Preprocessing */
    for i ← 0 to n2 − 1
        do a[i] ← 0
    root ← PRE-AC (set of lines of X, m1)
    PRE-KMP-FOR-B (X, m1, next)
    /* Searching */
    for row ← 0 to n1 − 1
        do r ← root
            for column ← 0 to n2 − 1
                do while child(r, Y[row, column]) = UNDEFINED
                        do r ← fail(r)
                    r ← child(r, Y[row, column])
                    if out(r) ≠ ∅
                        then k ← a[column]
                             while k > 0 and X[k, 0 . . . m2 − 1] = out(r)
                                 do k ← next[k]
                             a[column] ← k + 1
                             if a[column] = m1
                                 then OUTPUT (row − m1 + 1, column − m2 + 1)
                        else a[column] ← 0
```

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

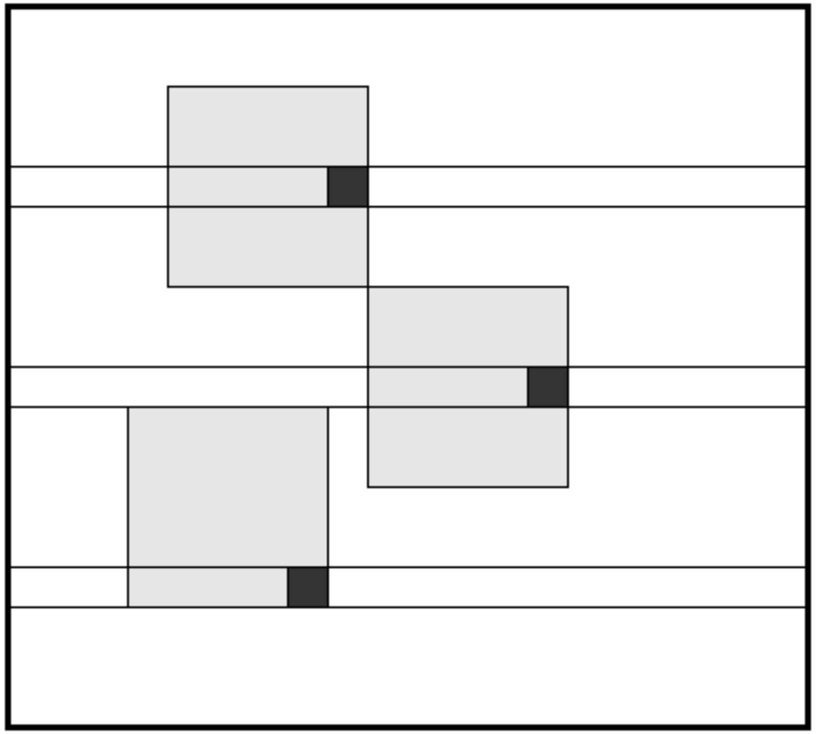
http://biit.cs.ut.ee/~vilo/edu/2005-06/Text_Algorithms/handbook-of-computer-science.pdf

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



http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.581

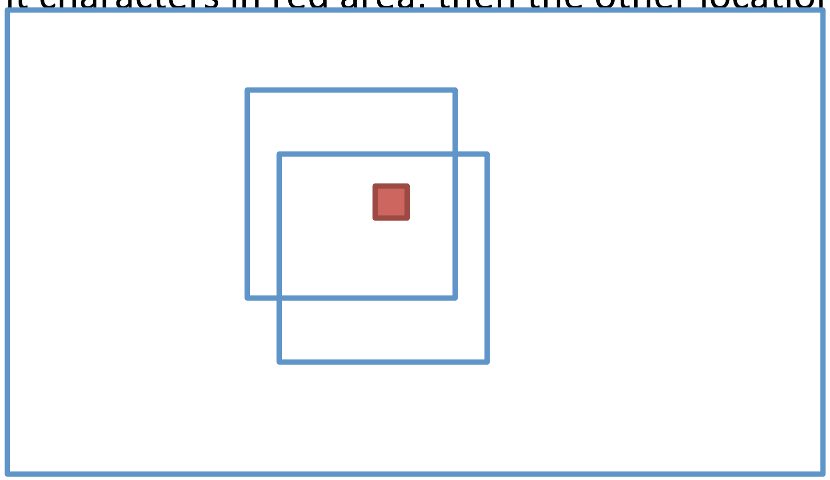
Elimination

- 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 cadidates 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

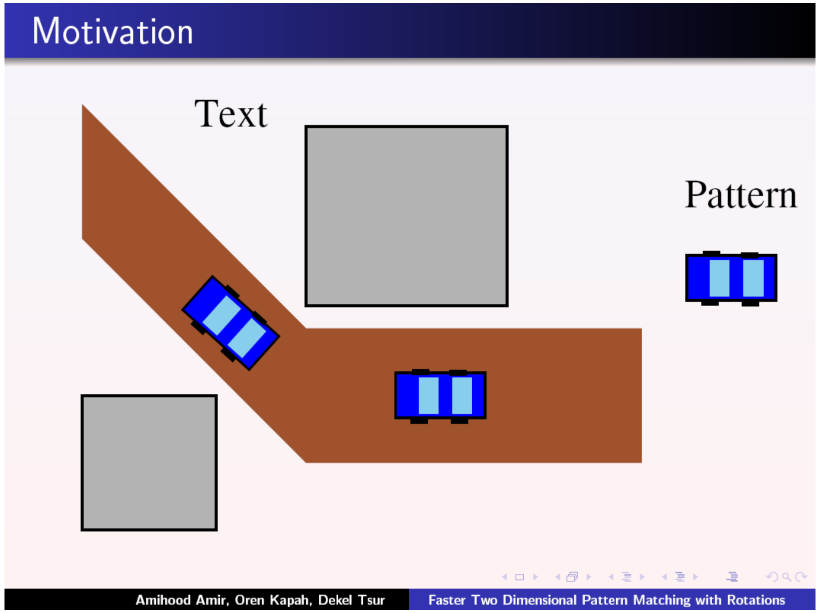
- Duel and witnesses : eliminate impossible locations!
- If P has two different characters in red area, then the other location can be eliminated!



- **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
- 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