• Modern Information Retrieval (1999)
  Ricardo-Baeza Yates and Berthier Ribeiro-Neto

• Flexible Pattern Matching in Strings (2002)
  Gonzalo Navarro and Mathieu Raffinot

• Algorithms on strings (2001)
  M. Crochemore, C. Hancart and T. Lecroq

• http://www-igm.univ-mlv.fr/~lecroq/string/index.html
String Matching

String matching: definition of the problem (text, pattern)

- **Exact matching**: depends on what we have: text or patterns
  - The patterns ---> Data structures for the patterns
    - 1 pattern ---> The algorithm depends on |p| and |Σ|
    - k patterns ---> The algorithm depends on k, |p| and |Σ|
  - Extensions
  - Regular Expressions
  - The text ---> Data structure for the text (suffix tree, ...)

- **Approximate matching**:
  - Dynamic programming
  - Sequence alignment (pairwise and multiple)
  - Sequence assembly: hash algorithm

- **Probabilistic search**: Hidden Markov Models
How does the matching algorithms made the search?

There is a sliding window along the text against which the pattern is compared:

Text:

Pattern:

At each step the comparison is made and the window is shifted to the right.

Which are the facts that differentiate the algorithms?

1. How the comparison is made.
2. The length of the shift.
Experimental efficiency (Navarro & Raffinot)

BNDM : Backward Nondeterministic Dawg Matching
BOM : Backward Oracle Matching
Automata Factor Oracle

Factor Oracle of GTA

All states are final ==> Recognizes all factors .... and more!

Hip: recognizes all the factors of GTA

This state recognizes all the factors that
• end in this position
• have not been recognized before as GTAT, TAT, AT.

All factors have been recognized
Automata Factor Oracle: insertion algorithm

Factor Oracle of $G T A T G T A$

What is the meaning of a suffix link?

The suffix link points to the state that recognizes the following factors:
- GT
- ATGT
- TGT

And this fact is pictured as follows ...
Automata Factor Oracle: insertion algorithm

... or more compacted ...
**Insertion algorithm**: Given a new character $T$ we add the next state $i$;

Is there a transition labeled with $T$ from state $j$?

If this transition labeled with $T$ exists ...

then the remaining factors are recognized recursively by the automata and a new suffix ling is inserted

Otherwise this transition labeled with $T$ doesn’t exist, then ...


Automata Factor Oracle: insertion algorithm

... if there is no transition link labeled with T then we add it...

... and recursively we continue the insertion of the character T following the suffix link of ...

... until the character T appears as a valid transition or the initial node is reached...

... and a new suffix link is inserted from the node i to the last visited node.
BOM algorithm

The automata

recognizes all the factors of GTGTA…

…and also recognizes strings that are not factors …

But, the main property is:
If a string is not recognized ==> it is not a factor!

This is the basis of the BOM search algorithm
BOM algorithm (Backward Oracle Matching)

• How the comparison is made?

Text:

Pattern: Automata: Factor Oracle

Check if the suffix is a factor

• Which is the next position of the window?

The position determined by the last character of the text with a transition in the automata
• We construct the automaton of the inverse pattern. Given ATGTATG

• And the search phase: G T A C T A G A A T G T G T A G A C A T G T A T G G T G A...
BOM algorithm: example

• We construct the automaton of the inverse pattern. Given ATGTATG

• And the search phase: G T A C T A G A A T G T G T A G A C A T G T A T G G T G

  ATG TATG

  ATG TATG
We construct the automaton of the inverse pattern. Given ATGTATG

And the search phase:

G T A C T A G A A T G T G T A G A C A T G T A T G G T G

ATGTATG

ATGTATG

ATGTATG
BOM algorithm: example

- We construct the automaton of the inverse pattern. Given ATGTATG

- And the search phase:

  GTACTAGA ATGTGTAGA CATG TATGGTG
BOM algorithm: example

• We construct the automaton of the inverse pattern. Given ATGTATG

• And the search phase: G T A C T A G A A T G T G T A G A C A T G T A T G G T G ...

  AT GTATG
  AT GTATG
  AT GTATG
  AT G TATG
  AT G TATG
BOM algorithm: example

• We construct the automaton of the inverse pattern. Given ATGTATG

• And the search phase: GTACTAGAATGTGTAGACATAGTKATGTG...