Recuperació de la informació

- 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

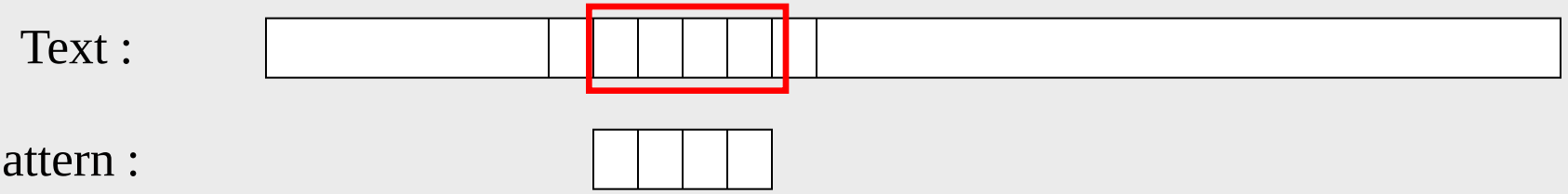
String matching: definition of the problem (text,pattern)

- 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

String matching: one pattern

How does the matching algorithms made the search?

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



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

Which are the facts that differentiate the algorithms?

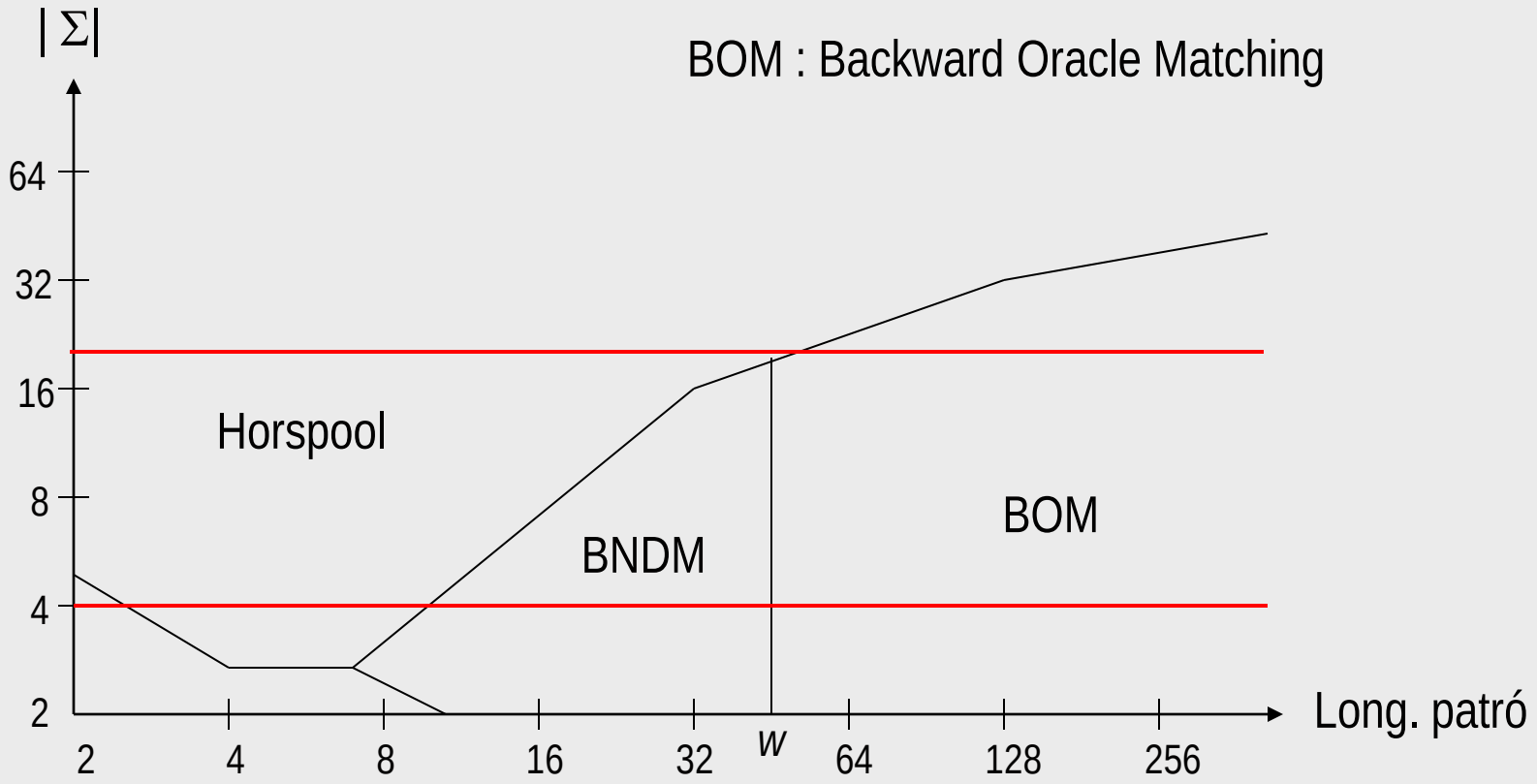
1. How the comparison is made.
2. The length of the shift.

Alg. Cerca exacta d'un patró (text on-line)

Algorismes més eficients (Navarro & Raffinot)

BNDM : Backward Nondeterministic Dawg Matching

BOM : Backward Oracle Matching



Automaton

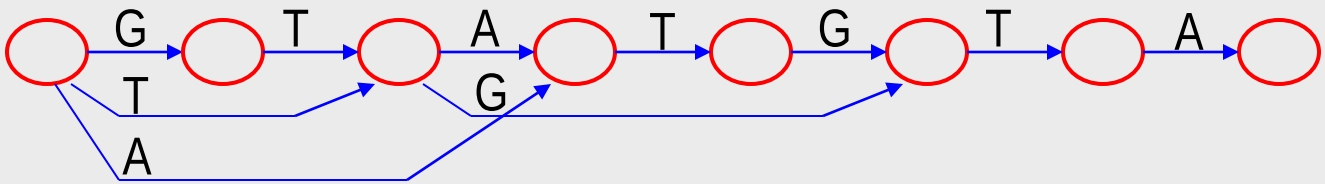
Given a pattern p, we find for an automaton A that verifies the properties:

1. *A* is acyclic.
2. *A* recognizes at least the factors of *p*.
3. *A* has the fewer states as possible.
4. *A* has a linear number of transitions according to the length of *p*.

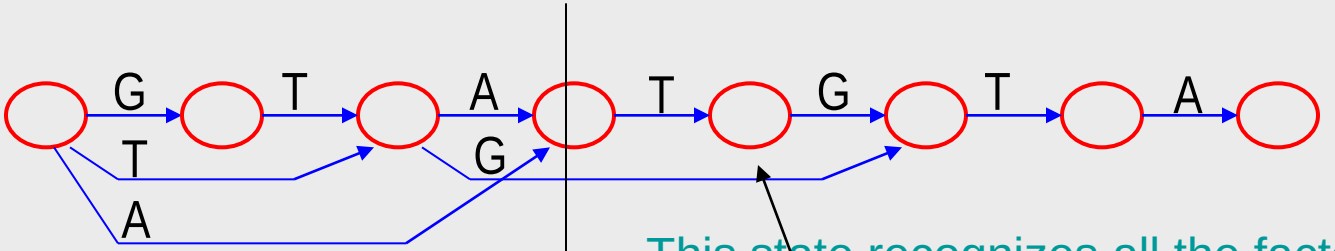
… and at the end of the last century …

Automaton Factor Oracle: properties

Given the word G T A T G T A



All states are accepting states ==> Recognize all the factors …. and more



Hip: recognize all factors of GTA

This state recognizes all the factors that ends in the fourth letter that have not been accepted before: GTAT, TAT, AT (note that T had been recognized before).

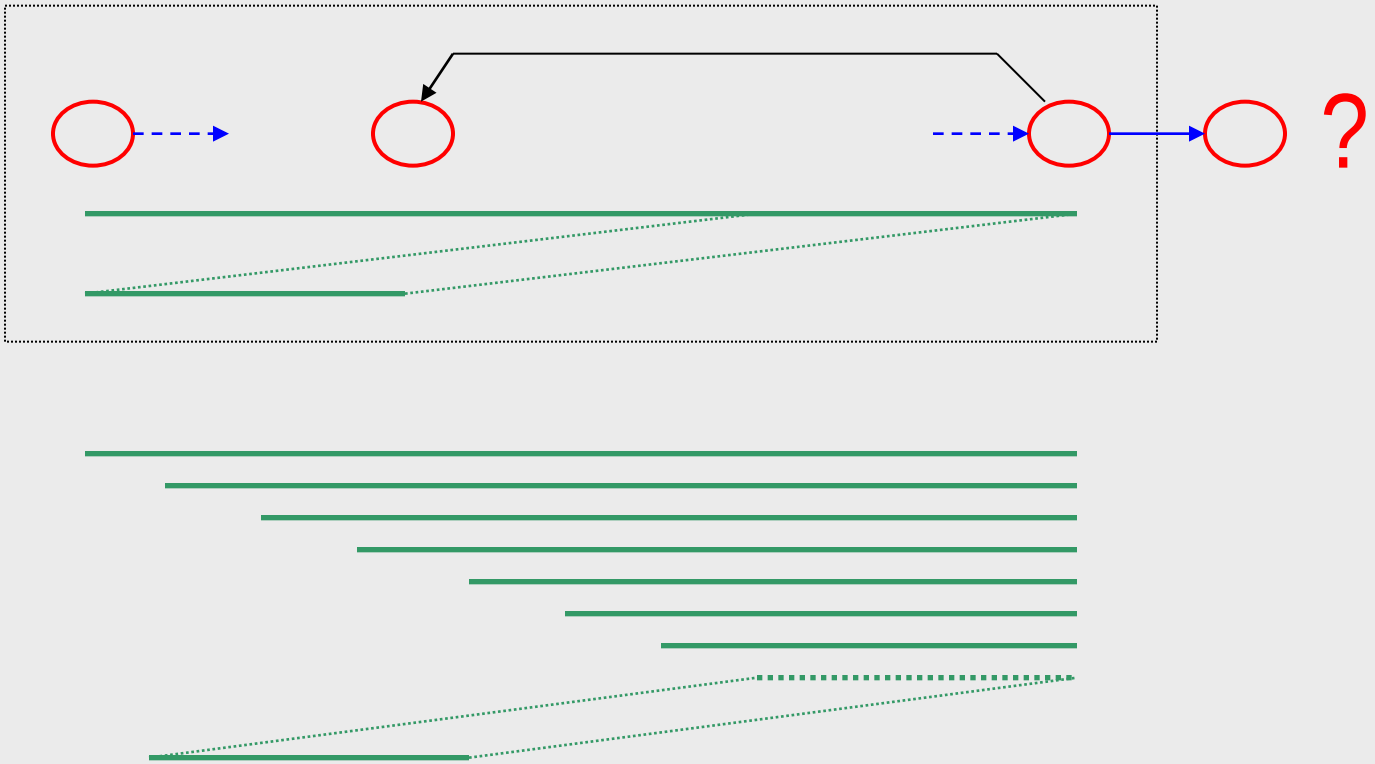
All the factors of the first four letters have been recognized.

Automaton Factor Oracle: algorithm

Algorithm:

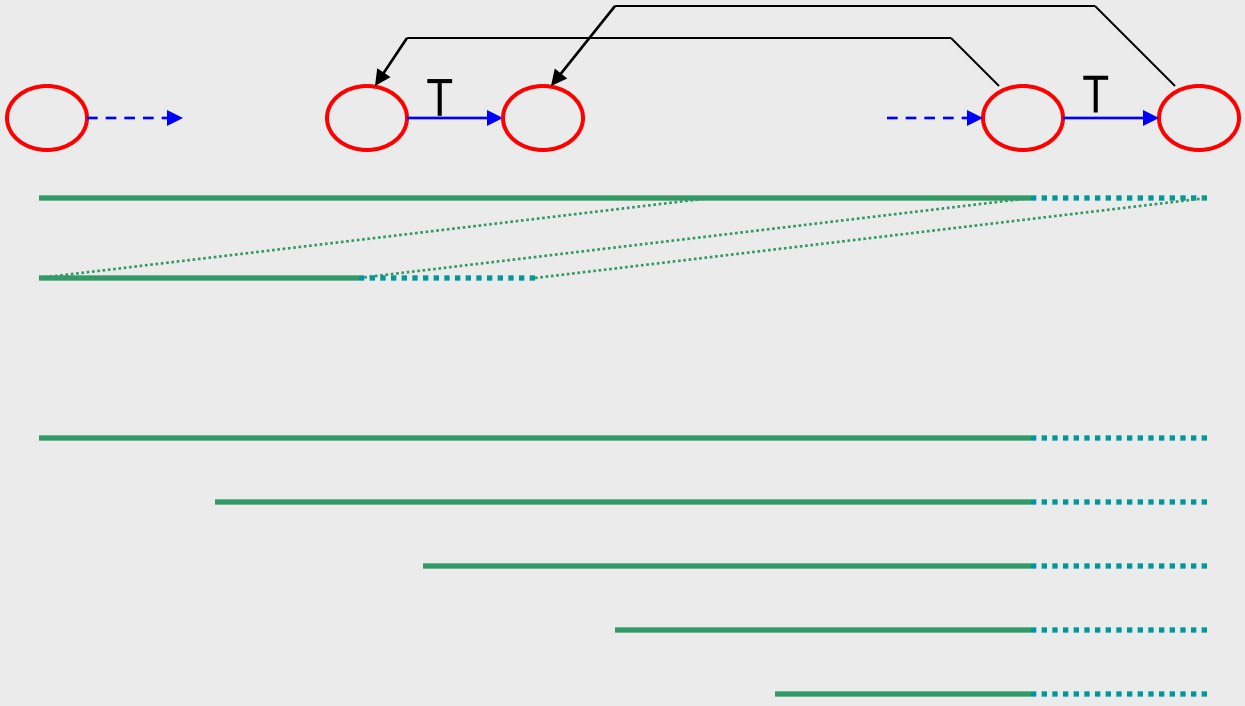
for *i=1* **to** *p* **do**

Add those transitions that recognize the new factors that end in letter i;



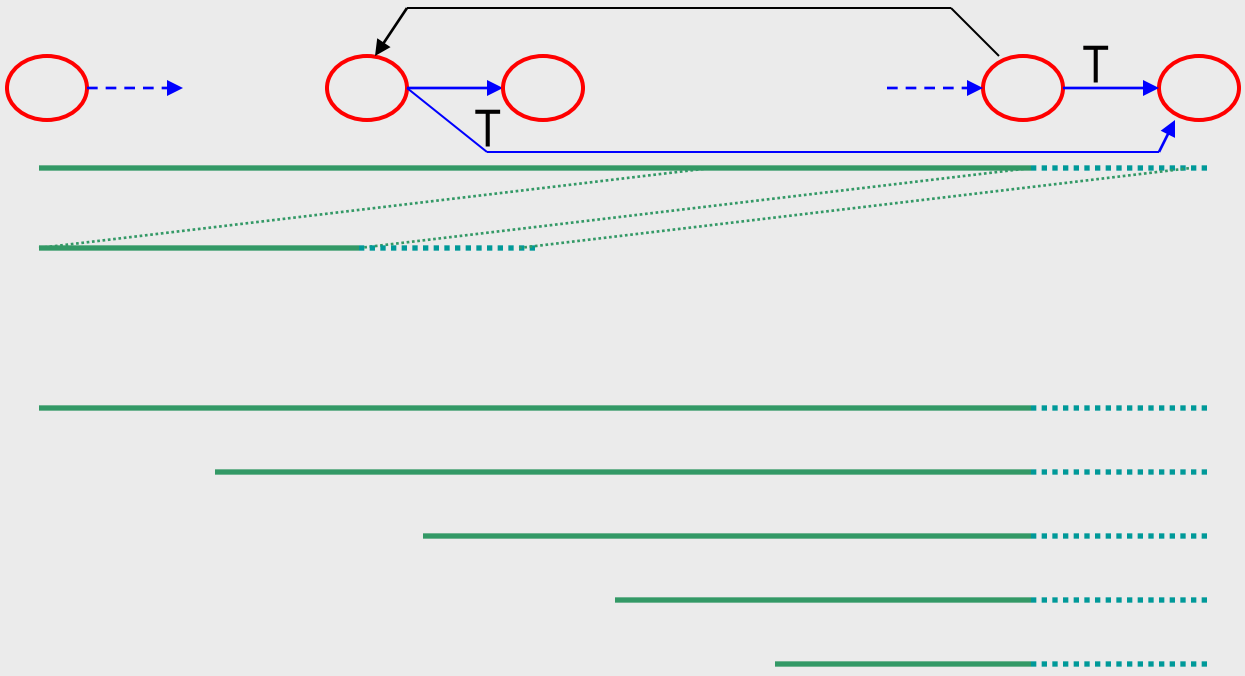
Automaton Factor Oracle: algorithm

What happens if the transition is in the automaton?

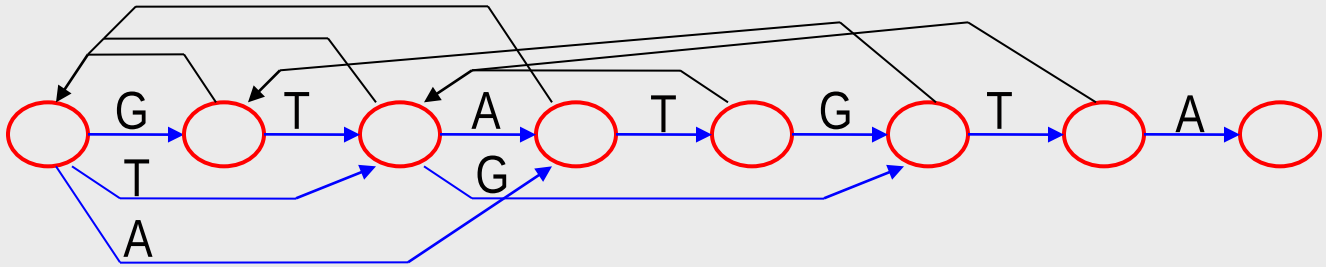


Automaton Factor Oracle: algorithm

What happens if transition isn't in the automaton?



Automaton Factor Oracle and BOM.



This automaton recognizes words that are not factors of GTATGTA like GTGTA => the affirmative answer is not informative, …. but

The negative answer ==> the word isn't a factor!

Is the strategy of the BOM algorithm.

Algorithm BOM (Backward Oracle Matching)

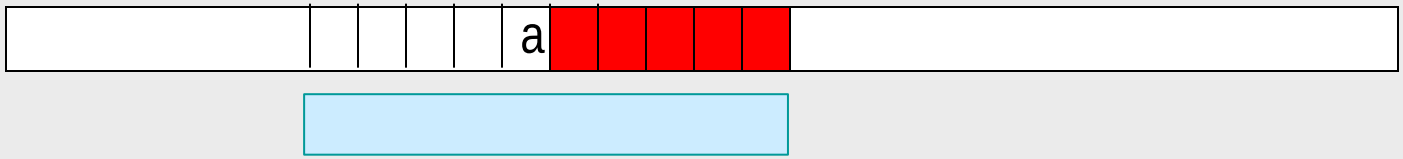
- How the comparison is made?

Text :

Pattern : Autòmaton Factor Oracle of the reverse pattern

Check if the suffix is a factor of the pattern.

- Which is the next position of the window?

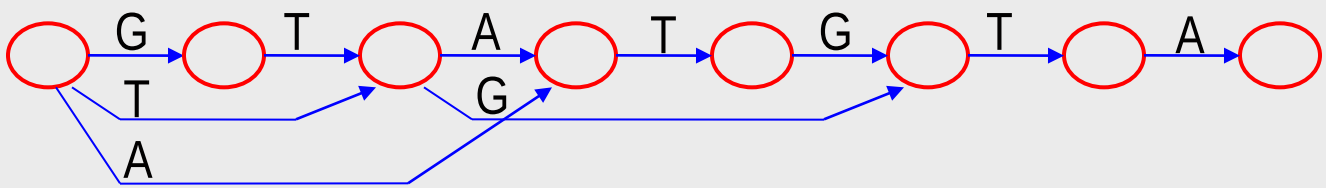


• If some letter is not found

• If the pattern is found:

BOM algorithm

- How the comparison is made?
  - Given the pattern ATGTATG we construct the automaton of the reverse pattern:

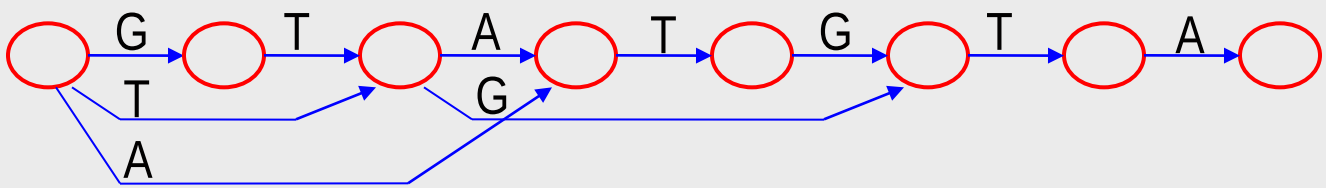


  - And the search phase : G T A C T A G A A T G T G T A G A C A T G T A T G G T G A...

A T G T A T G

BOM algorithm

- How the comparison is made?
  - Given the pattern ATGTATG we construct the automaton of the reverse pattern



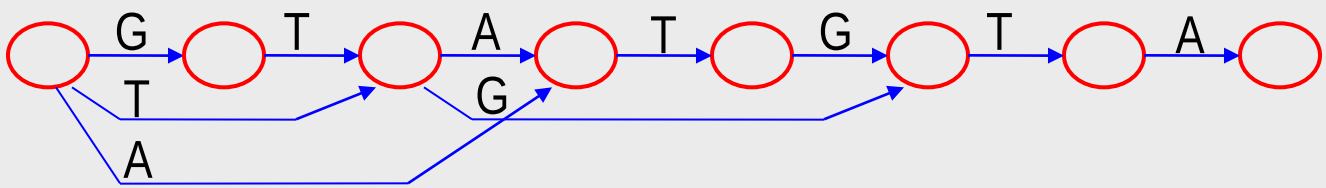
- And the search phase : G T A C T A G A A T G T G T A G A C A T G T A T G G T G

A T G T A T G

A T G T A T G

BOM algorithm

- How the comparison is made?
  - Given the pattern ATGTATG we construct the automaton of the reverse pattern



- 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

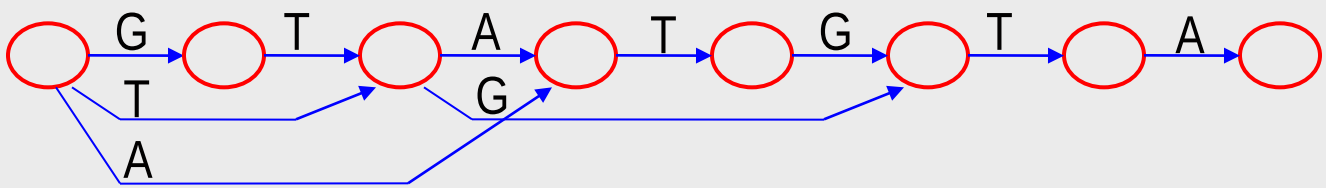
A T G T A T G

A T G T A T G

A T G T A T G

BOM algorithm

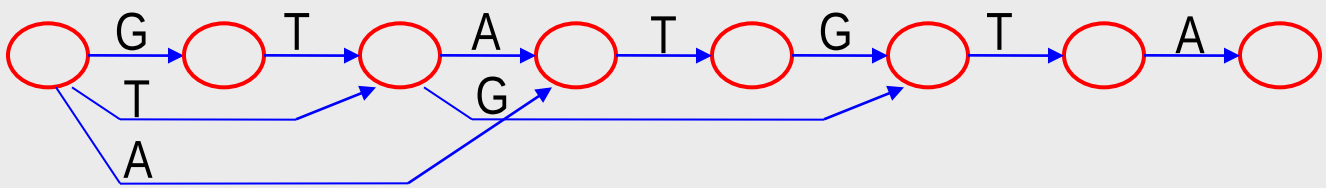
- How the comparison is made?
  - Given the pattern ATGTATG we construct the automaton of the reverse pattern



- 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

BOM algorithm

- How the comparison is made?
  - Given the pattern ATGTATG we construct the automaton of the reverse pattern



•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 T G T A T G

A T G T A T G

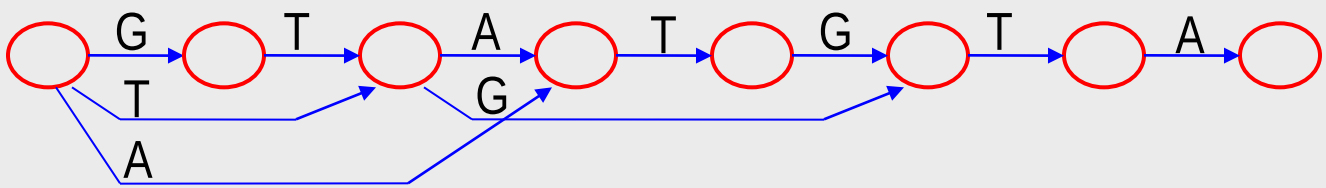
A T G T A T G

A T G T A T G

A T G T A T G

BOM algorithm

- How the comparison is made?
  - Given the pattern ATGTATG we construct the automaton of the reverse pattern



- 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 ...