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

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

Algorismes de:

Cerca de patrons (exacta i aproximada)
(String matching i Pattern matching)

Indexació de textos:
Suffix trees, Suffix arrays

String Matching

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

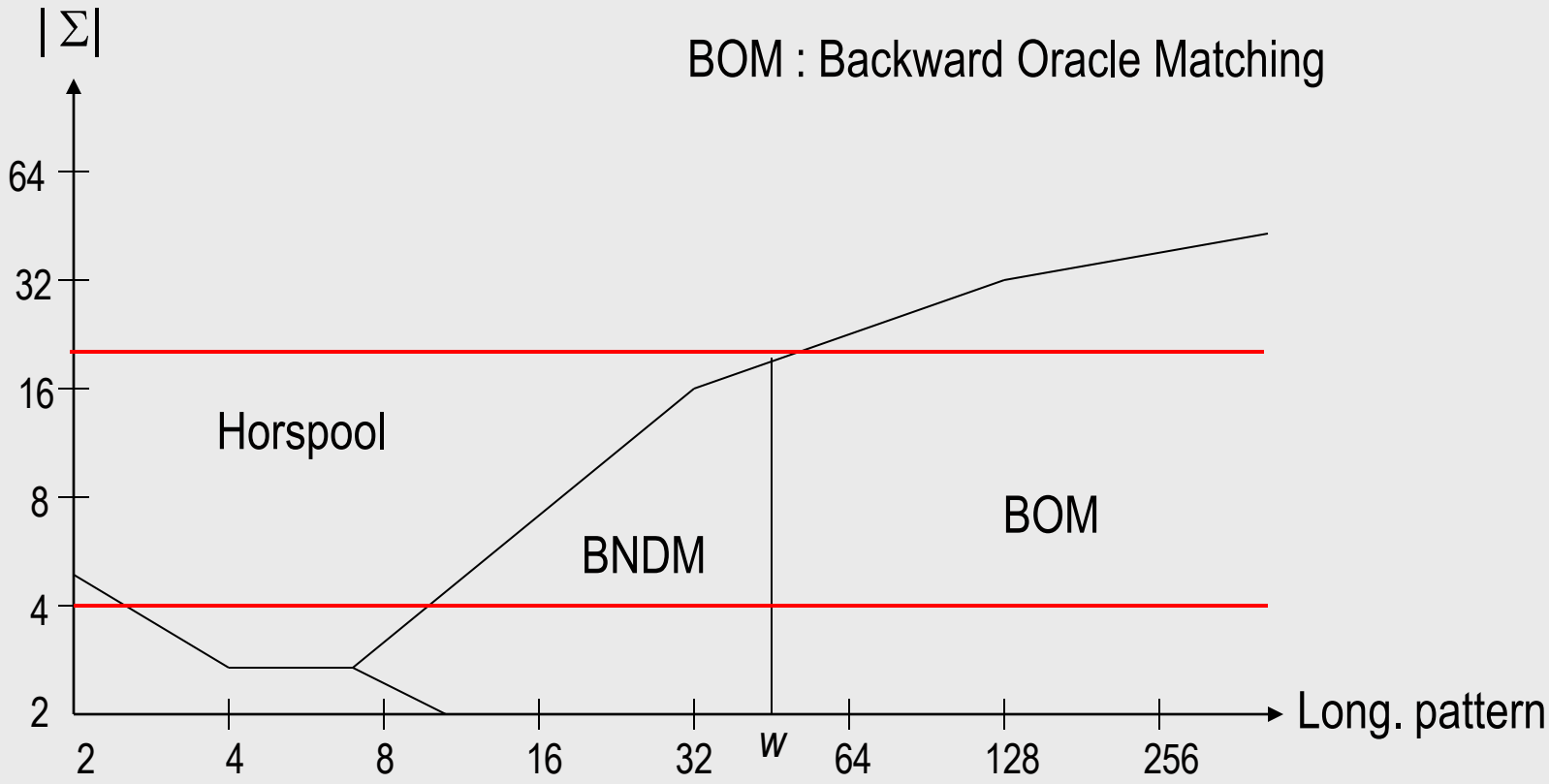
- **Exact matching:** depends on what we have: text or patterns
  - **The patterns ---> Data structures for the patterns**
    - 1 pattern ---> The algorithm depends on |p| and |Σ|
    - k patterns ---> The algorithm depends on k, |p| and |Σ|
  - **Extensions**
  - **Regular Expressions**
  - **The text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

Exact string matching: one pattern (text on-line)

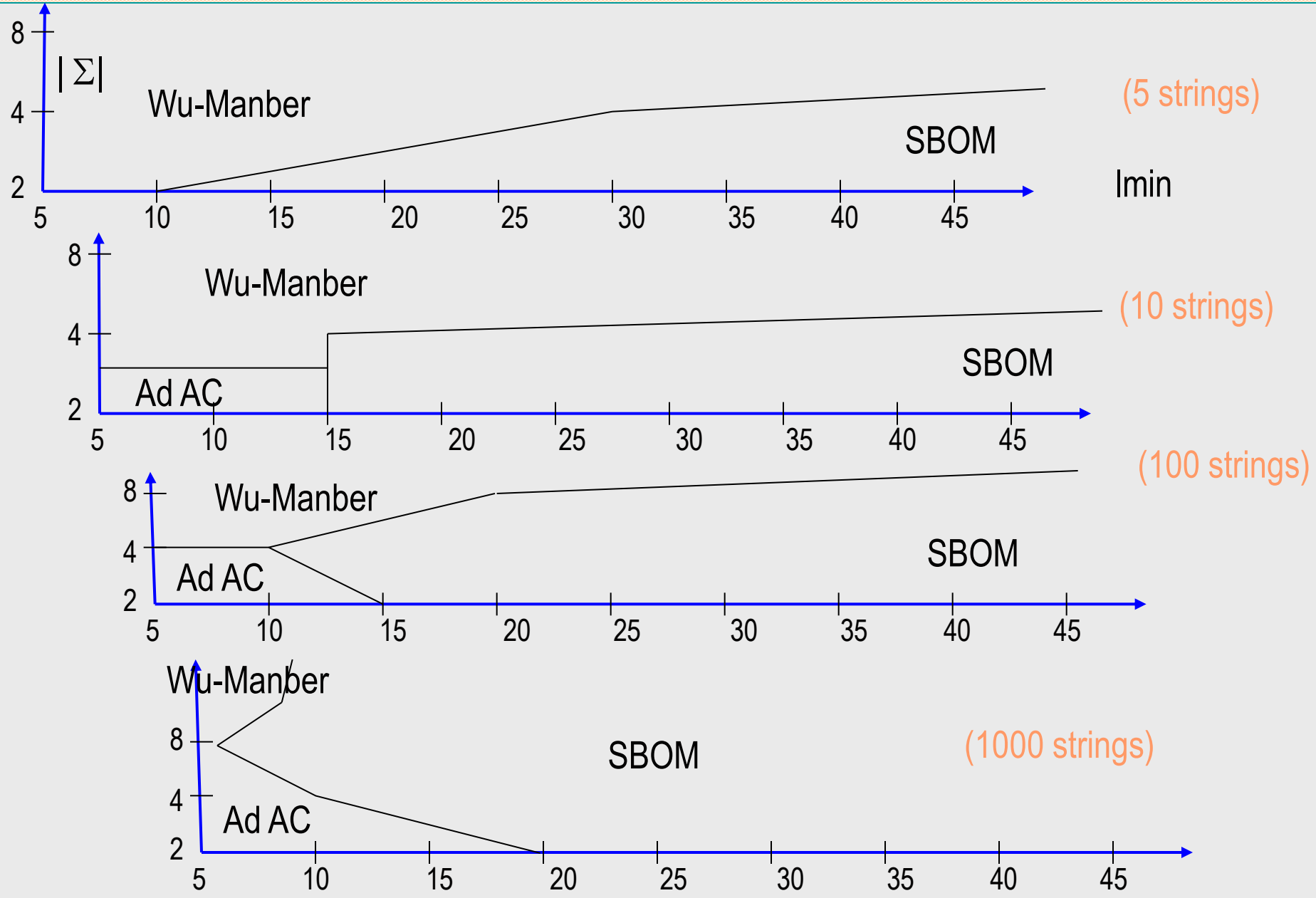
Experimental efficiency (Navarro & Raffinot)

BNDM : Backward Nondeterministic Dawg Matching

BOM : Backward Oracle Matching



Multiple string matching

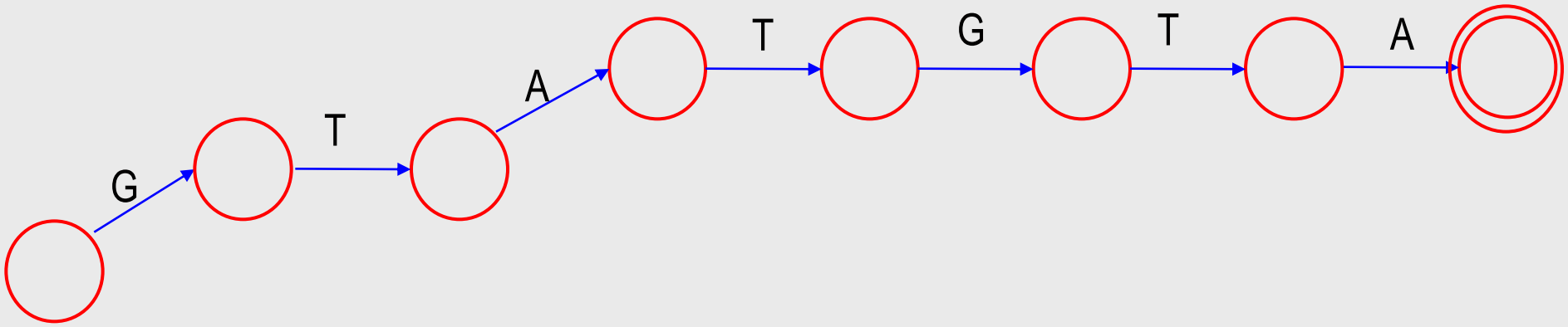


Trie

Construct the trie of GTATGTA,GTAT,TAATA,GTGTA

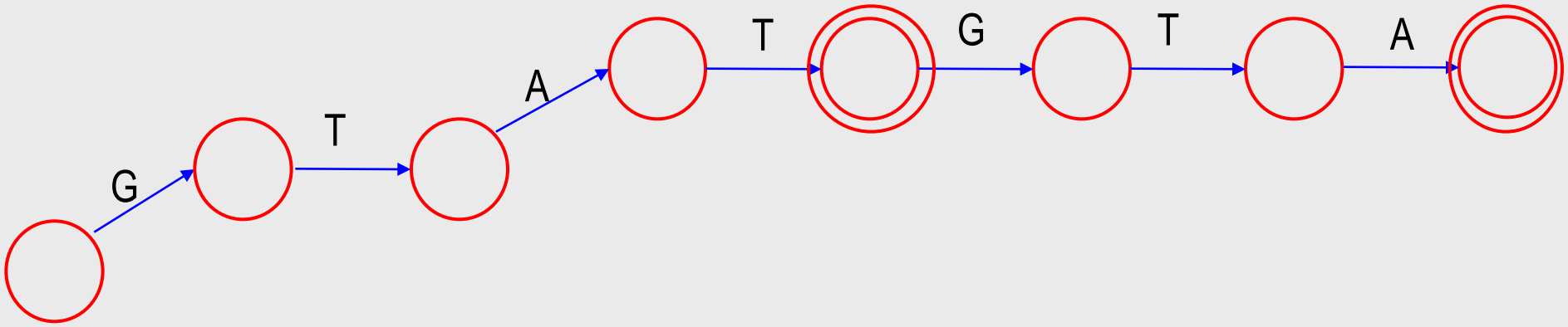
Trie

Construct the trie of
GTATGTA,GTAT,TAATA,GTGTA



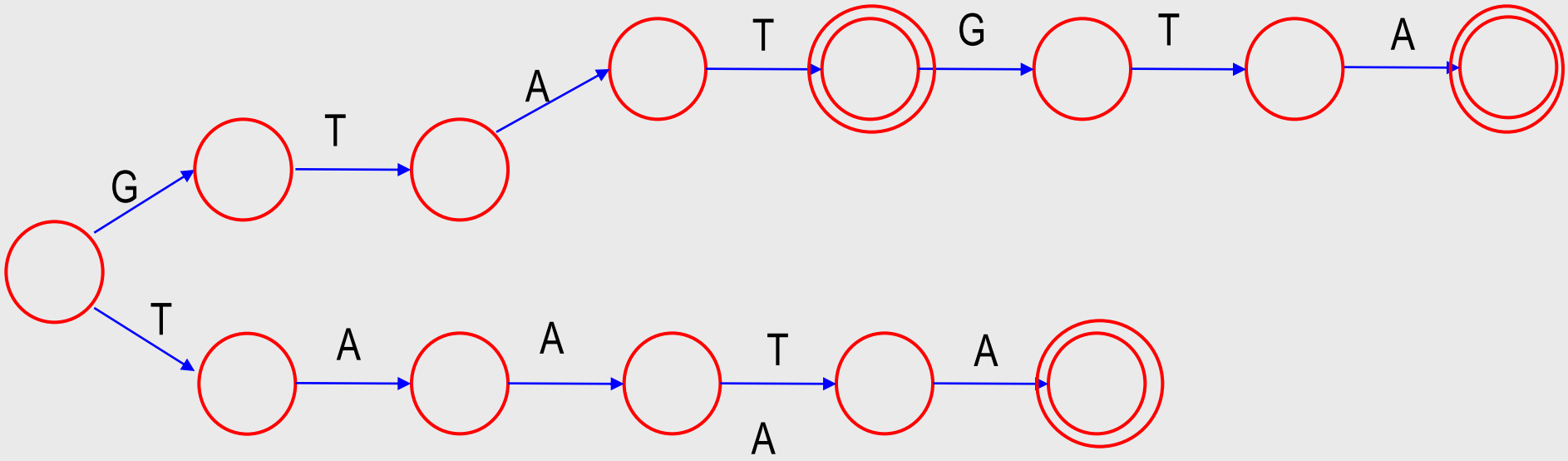
Trie

Construct the trie of
GTATGTA,GTAT,TAATA,GTGTA



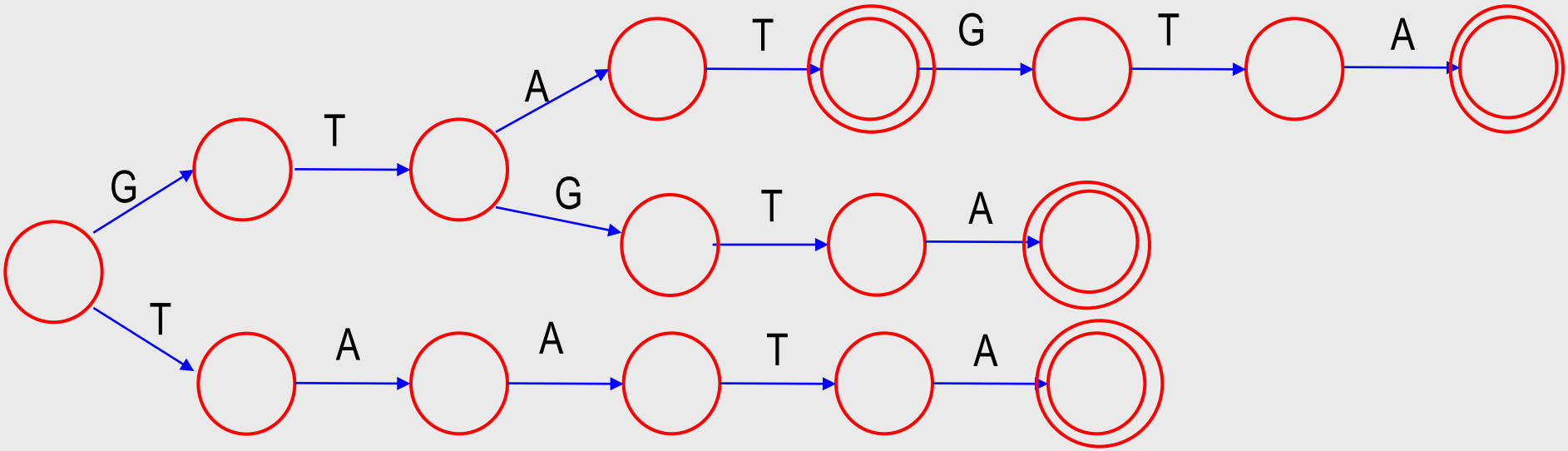
Trie

Construct the trie of
GTATGTA,GTAT,TAATA,GTGTA



Trie

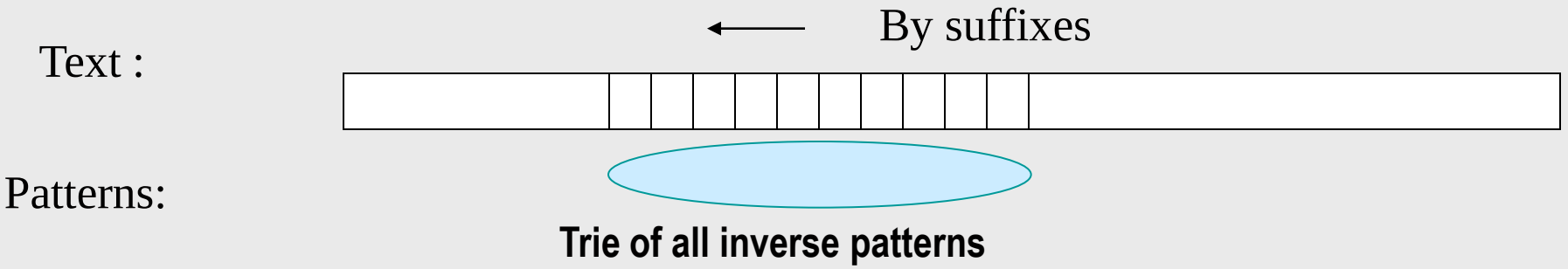
Construct the trie of
GTATGTA,GTAT,TAATA,GTGTA



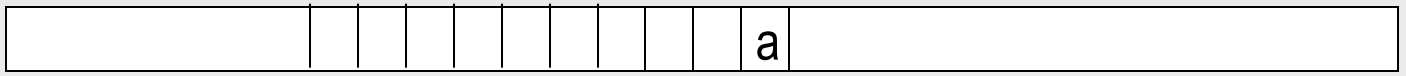
Which is the cost?

Set Horspool algorithm

- **How the comparison is made?**



- **Which is the next position of the window?**

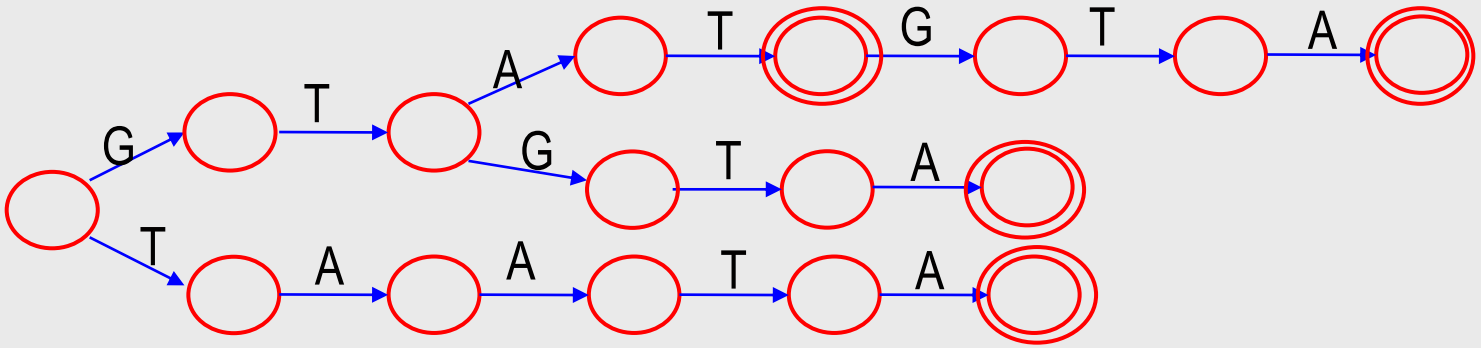


We shift until a is aligned with the first a in the trie not longer than *lmin*, or *lmin*

Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG

1. Construct the trie of GTATGTA, GTAT, TAATA i GTGTA

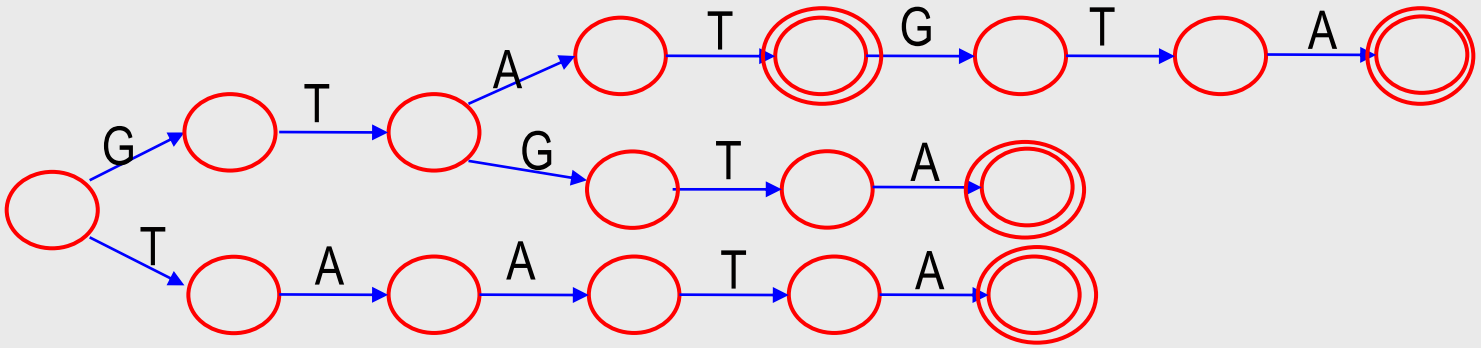


2. Determine *lmin*=

Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG

1. Construct the trie of GTATGTA, GTAT, TAATA i GTGTA



2. Determine *lmin=4*

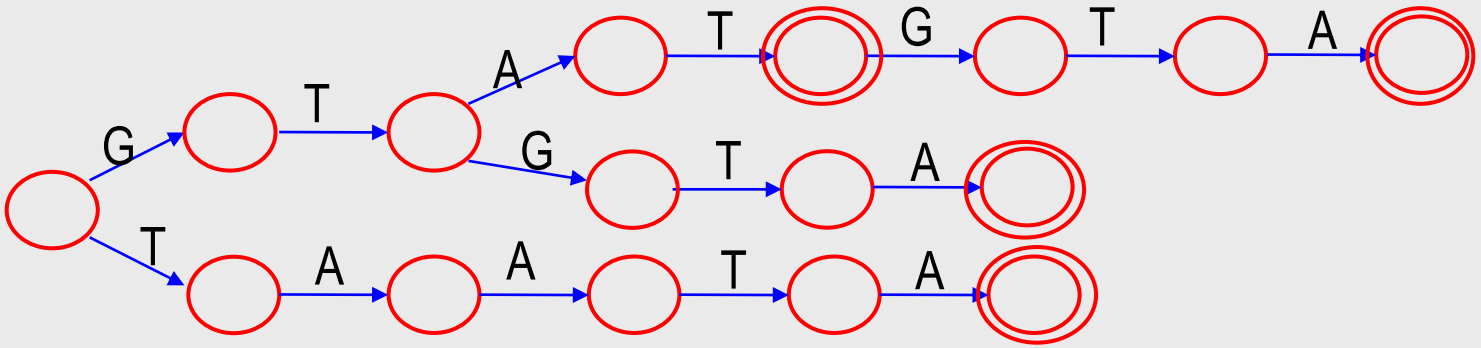
3. Determine the shift table

A	1
C	4 (lmin)
G	
T	

Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG

1. Construct the trie of GTATGTA, GTAT, TAATA i GTGTA



2. Determine *lmin=4*

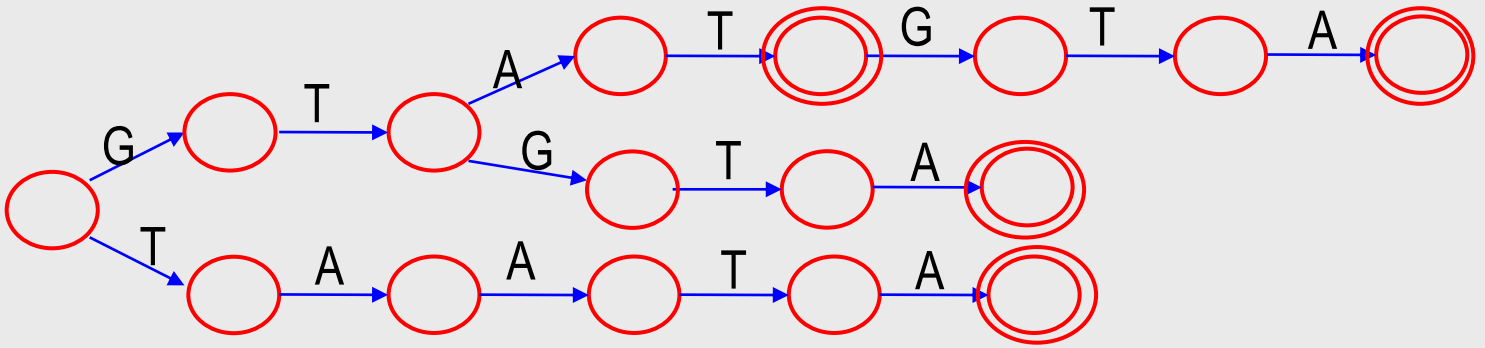
3. Determine the shift table

A	1
C	4 (lmin)
G	2
T	

Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG

1. Construct the trie of GTATGTA, GTAT, TAATA i GTGTA



2. Determine *lmin=4*

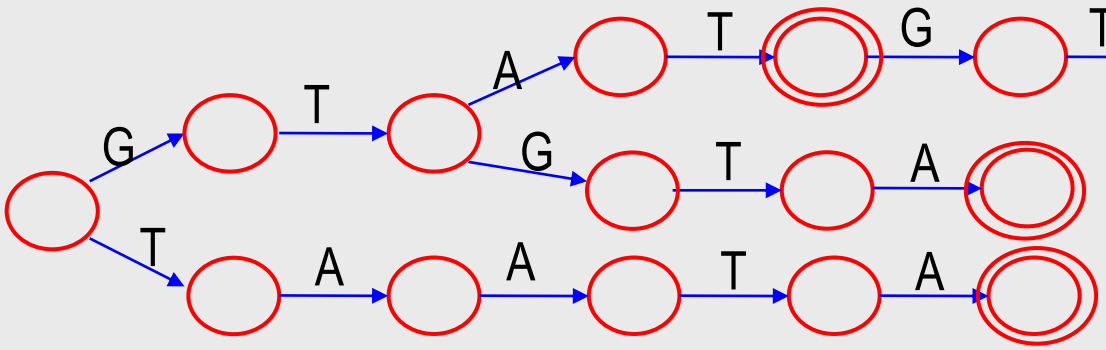
3. Determine the shift table

A	1
C	4 (lmin)
G	2
T	1

4. Find the patterns

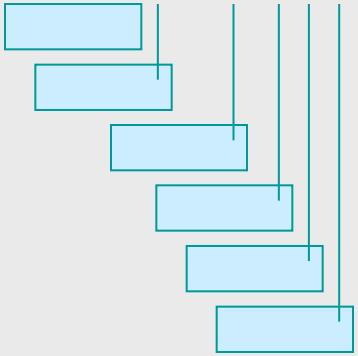
Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



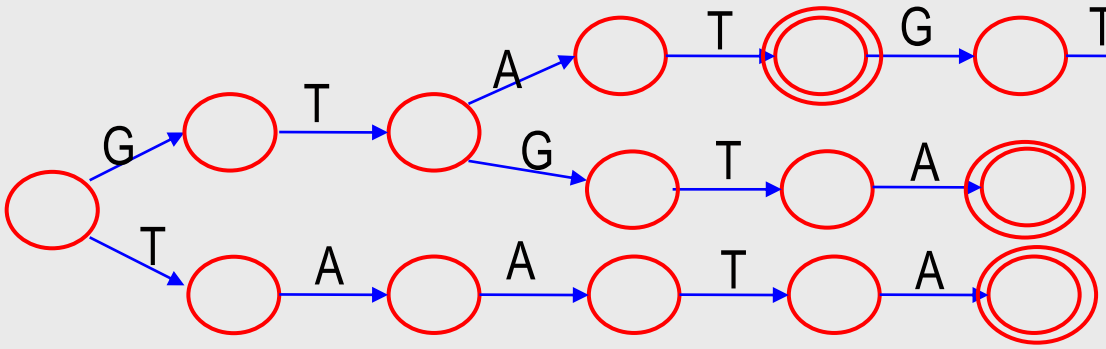
A	1
C	4 (lmin)
G	2
T	1

text: ACATGCTATGTGACA…



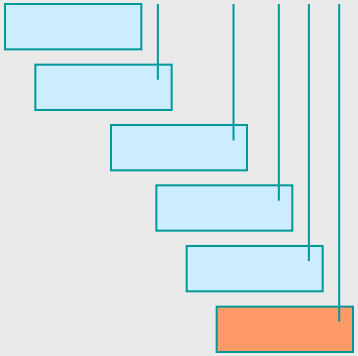
Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



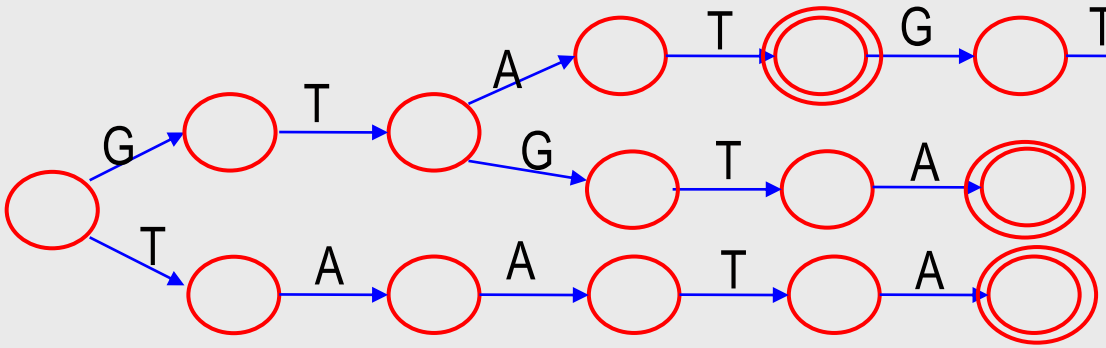
A	1
C	4 (lmin)
G	2
T	1

text: ACATGCTATGTGACA…

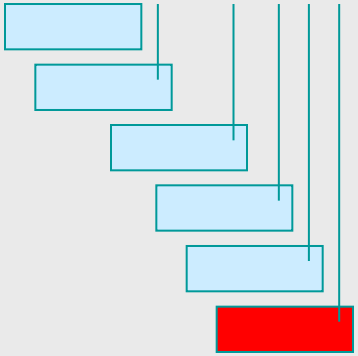


Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



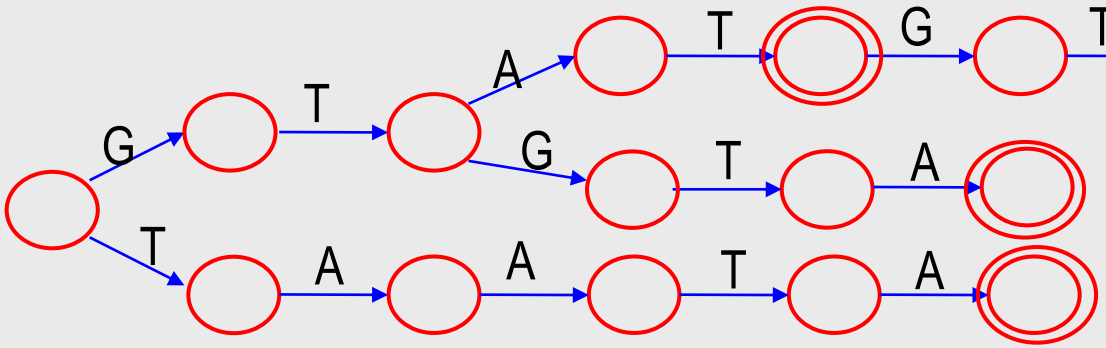
text: ACATGCTATGTGACA…



A	1
C	4 (lmin)
G	2
T	1

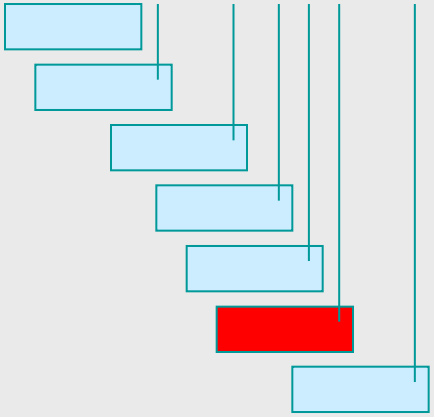
Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



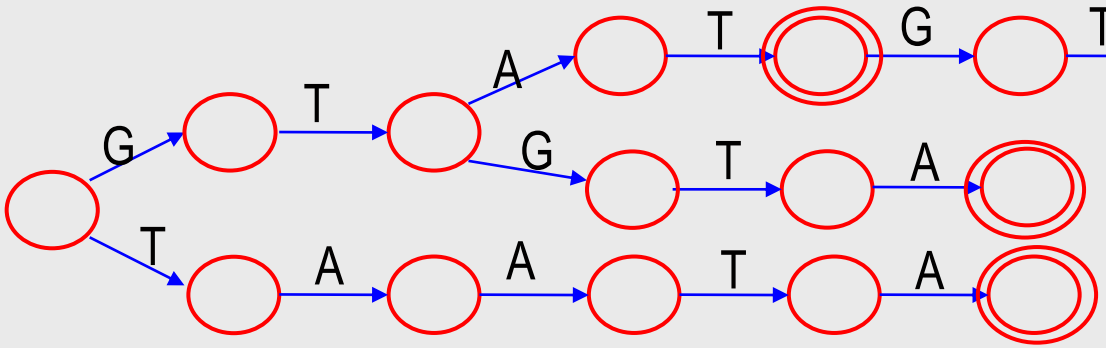
A	1
C	4 (lmin)
G	2
T	1

text: ACATGCTATGTGACA…

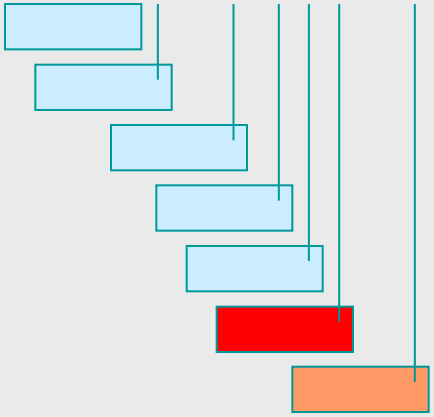


Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



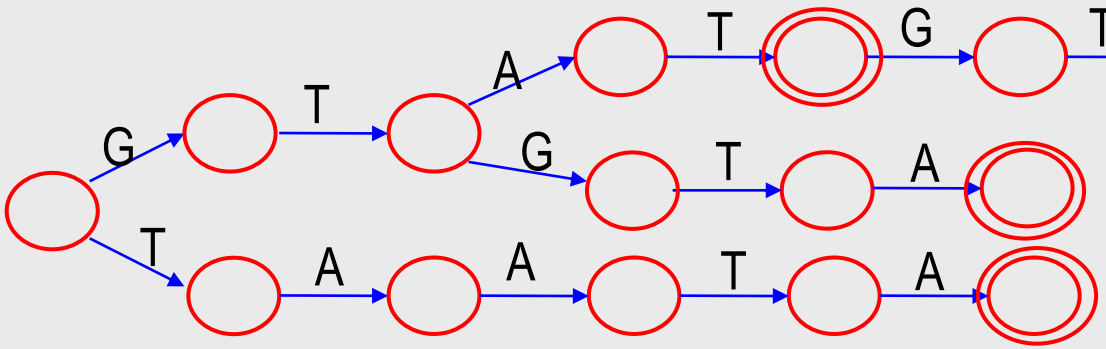
text: ACATGCTATGTGACA…



A	1
C	4 (lmin)
G	2
T	1

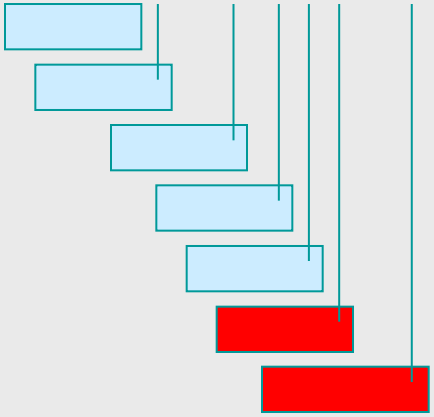
Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



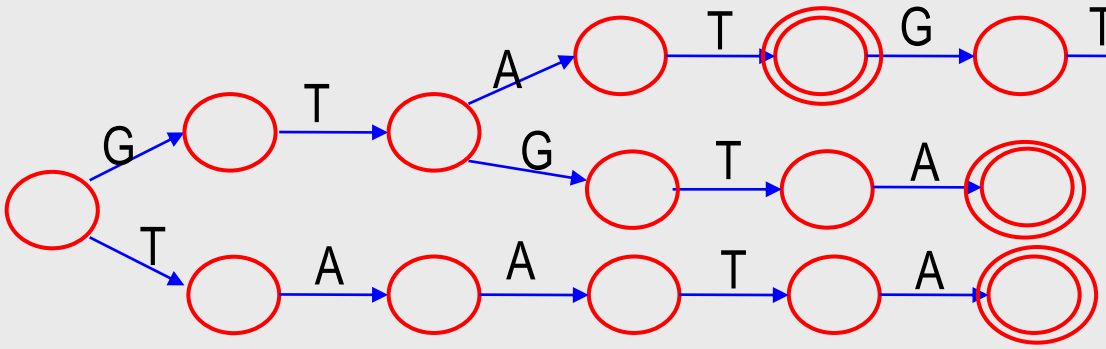
A	1
C	4 (lmin)
G	2
T	1

text: ACATGCTATGTGACA…



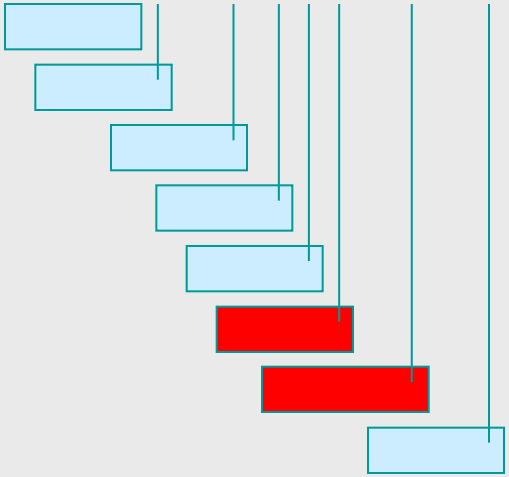
Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



A	1
C	4 (lmin)
G	2
T	1

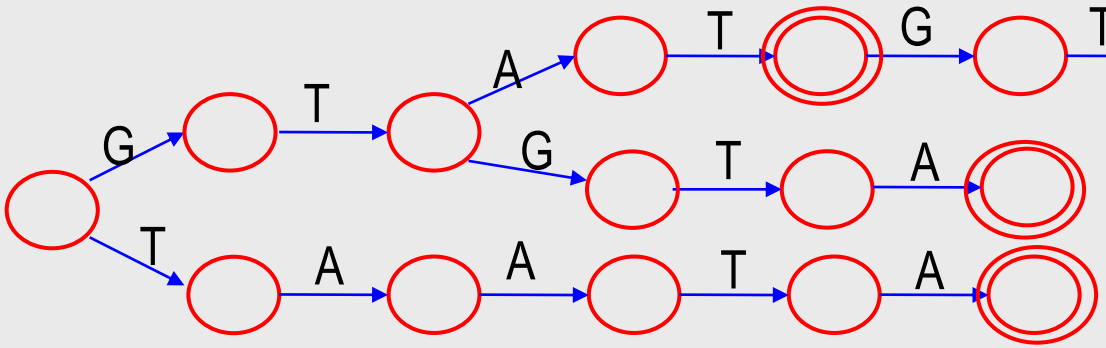
text: ACATGCTATGTGACA…



…

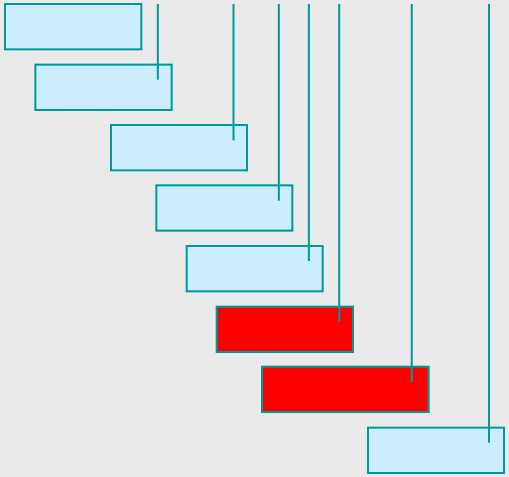
Set Horspool algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



A	1
C	4 (lmin)
G	2
T	1

text: ACATGCTATGTGACA…



…

As more patterns we search for, shorter shifts we do!

Is the expected length of the shifts related with the number of patterns?

Set Horspool algorithm →Wu-Manber algorithm

How the length of shifts can be increased?

By reading blocks of symbols instead of only one!
Given ATGTATG,TATG,ATAAT,ATGTG

1 símbol

A	1
C	4 (lmin)
G	2
T	1

2 símbols

AA	1
AC	3 (*LMIN-L+1*)
AG	
AT	
CA	
CC	
CG	
…	

Set Horspool algorithm →Wu-Manber algorithm

How the length of shifts can be increased?

By reading blocks of symbols instead of only one!
Given ATGTATG,TATG,ATAAT,ATGTG

1 símbol

A	1
C	4 (lmin)
G	2
T	1

2 símbols

AA	1
AC	3 (*LMIN-L+1*)
AG	3
AT	
CA	
CC	
CG	
…	

Set Horspool algorithm →Wu-Manber algorithm

How the length of shifts can be increased?

By reading blocks of symbols instead of only one!
Given ATGTATG,TATG,ATAAT,ATGTG

1 símbol

A	1
C	4 (lmin)
G	2
T	1

2 símbols

AA	1
AC	3 (*LMIN-L+1*)
AG	3
AT	1
CA	
CC	
CG	
…	

Set Horspool algorithm →Wu-Manber algorithm

How the length of shifts can be increased?

By reading blocks of symbols instead of only one!
Given ATGTATG,TATG,ATAAT,ATGTG

1 símbol

A	1
C	4 (lmin)
G	2
T	1

2 símbols

AA	1
AC	3 (*LMIN-L+1*)
AG	3
AT	1
CA	3
CC	3
CG	3
…	

Set Horspool algorithm →Wu-Manber algorithm

How the length of shifts can be increased?

By reading blocks of symbols instead of only one!
Given ATGTATG,TATG,ATAAT,ATGTG

1 símbol

A	1
C	4 (lmin)
G	2
T	1

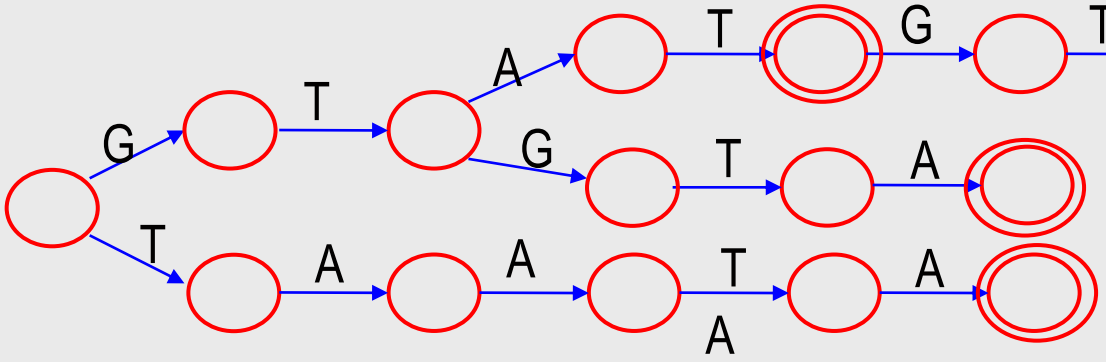
2 símbols

AA	1
AC	3 (*LMIN-L+1*)
AG	3
AT	1
CA	3
CC	3
CG	3
…	

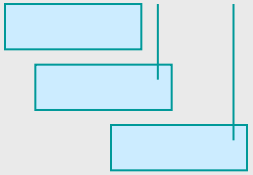
AA	1
AT	1
GT	1
TA	2
TG	2

Wu-Manber algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



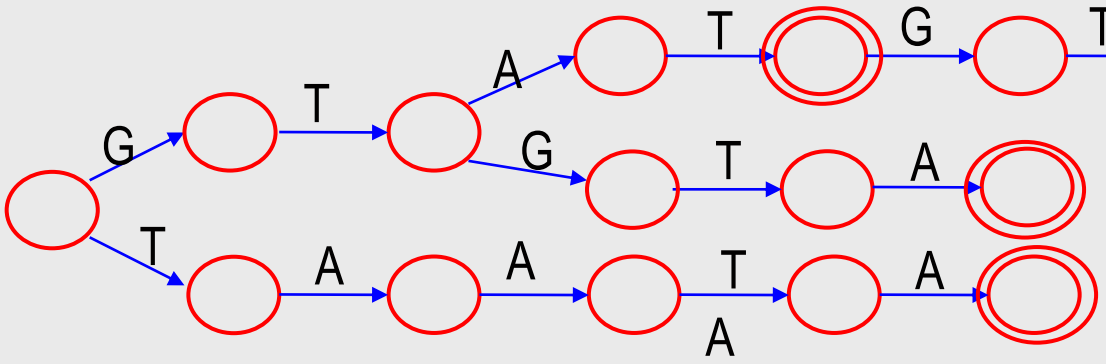
text: ACATGCTATGTGACATAATA



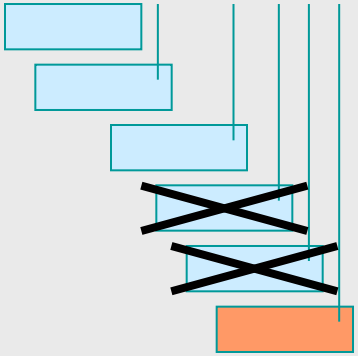
AA	1
AT	1
GT	1
TA	2
TG	2

Wu-Manber algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



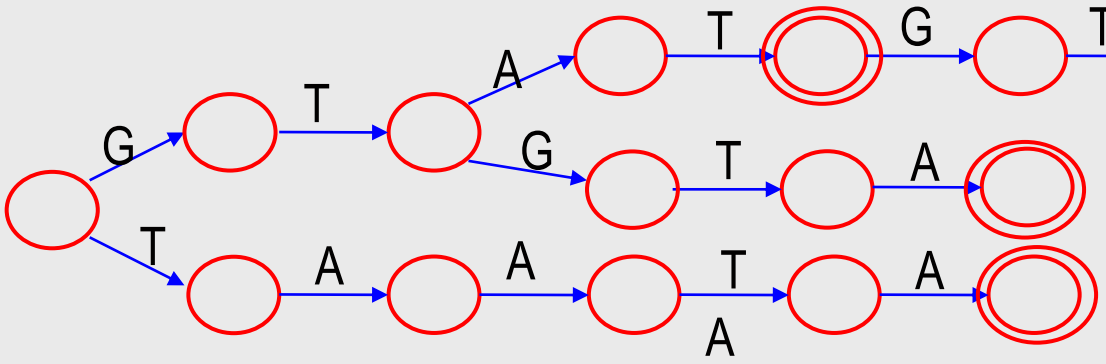
text: ACATGCTATGTGACATAATA



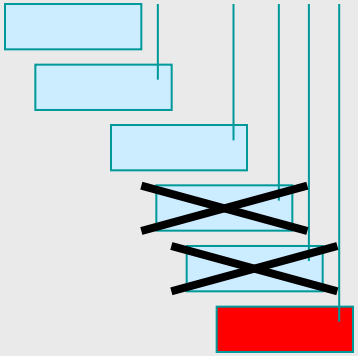
AA	1
AT	1
GT	1
TA	2
TG	2

Wu-Manber algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



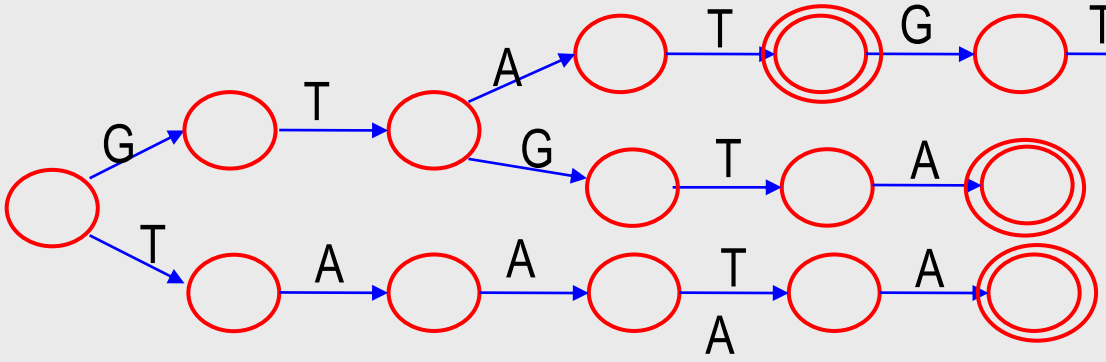
text: ACATGCTATGTGACATAATA



AA	1
AT	1
GT	1
TA	2
TG	2

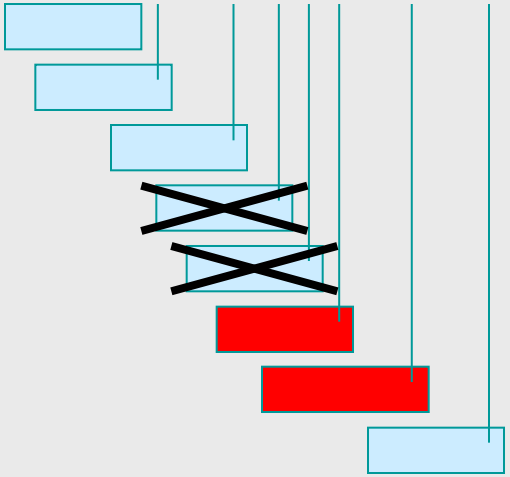
Wu-Manber algorithm

Search for ATGTATG,TATG,ATAAT,ATGTG



AA	1
AT	1
GT	1
TA	2
TG	2

text: ACATGCTATGTGACATAATA

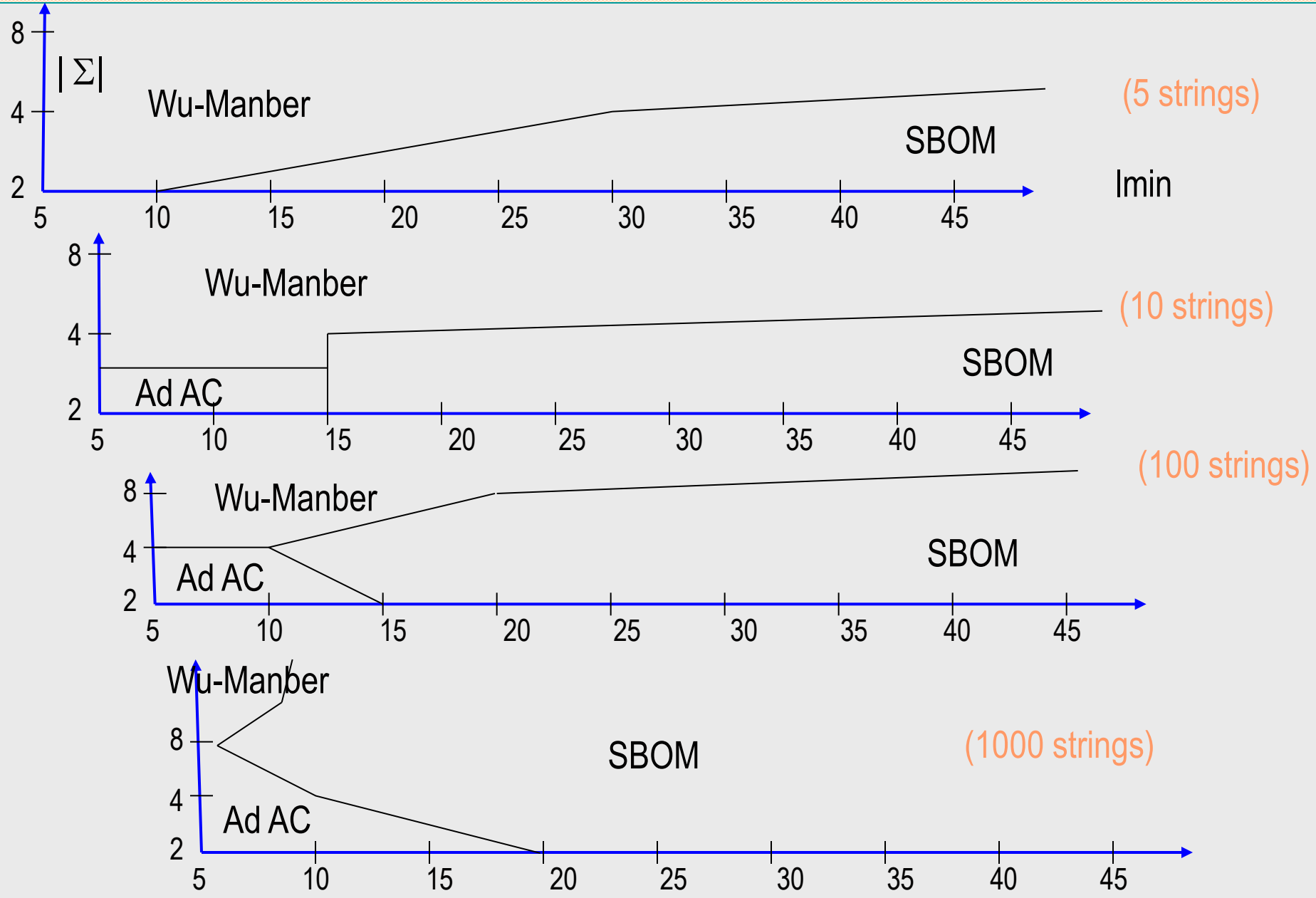


...

But given k patterns, how many symbols we should take ?

$\log_{|\Sigma|} 2*lmin*k$

Multiple string matching



BOM algorithm (Backward Oracle Matching)

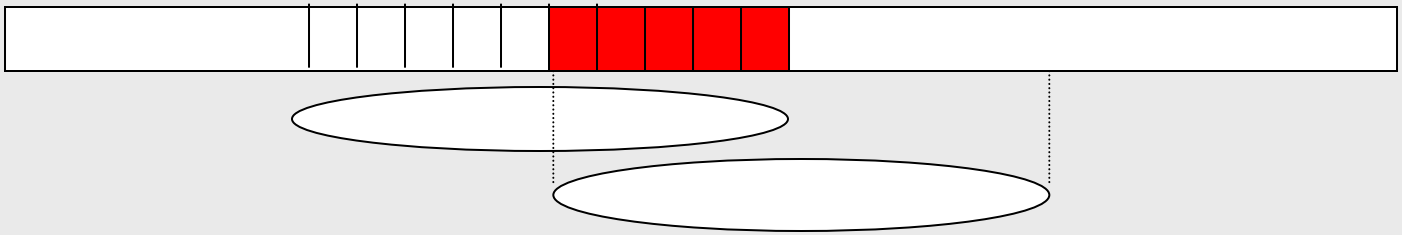
- **How the comparison is made?**

Text :

Pattern : Automata: Factor Oracle

Check if the suffix is a factor of any pattern

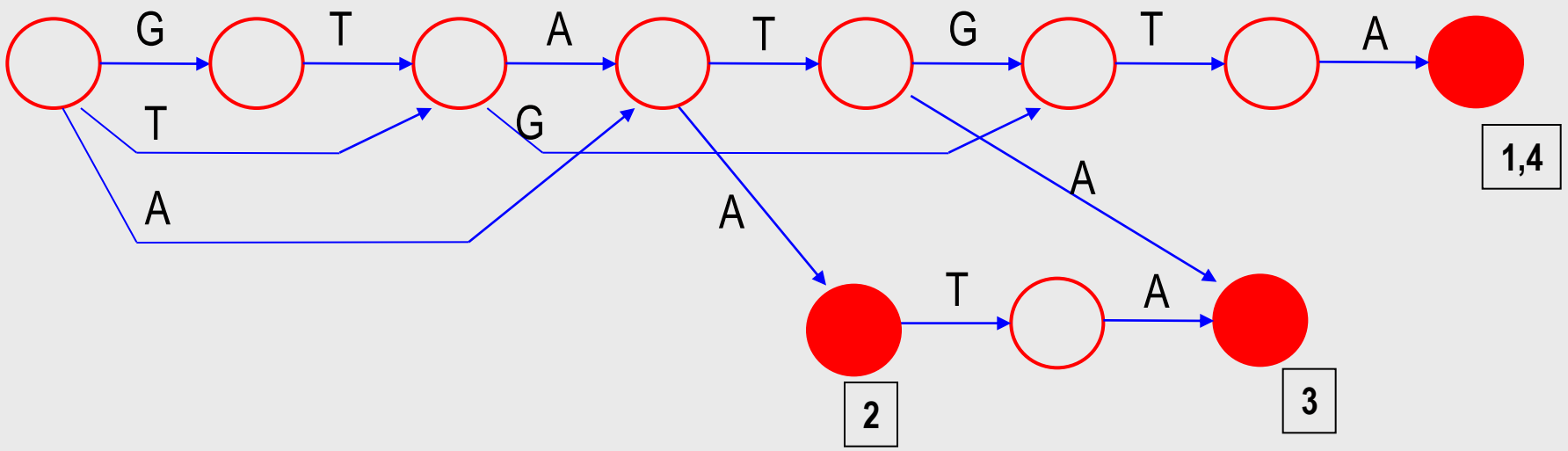
- **Which is the next position of the window?**



The position determined by the last character of the text
with a transition in the automata

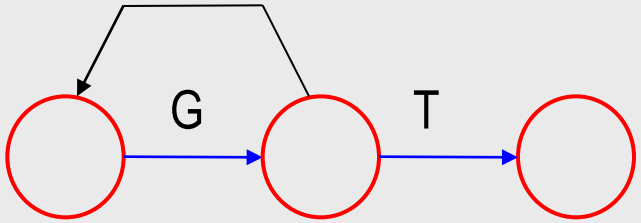
Factor Oracle of k strings

How can we build the Factor Oracle of GTATGTA, GTAA, TAATA i GTGTA ?



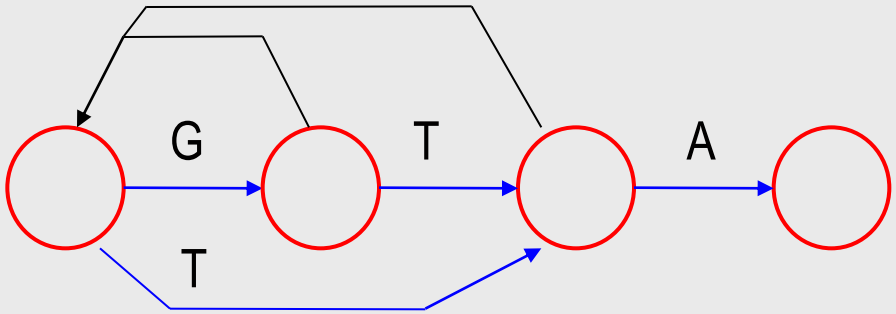
Factor Oracle of k strings

Given the Factor Oracle of GTATGTA



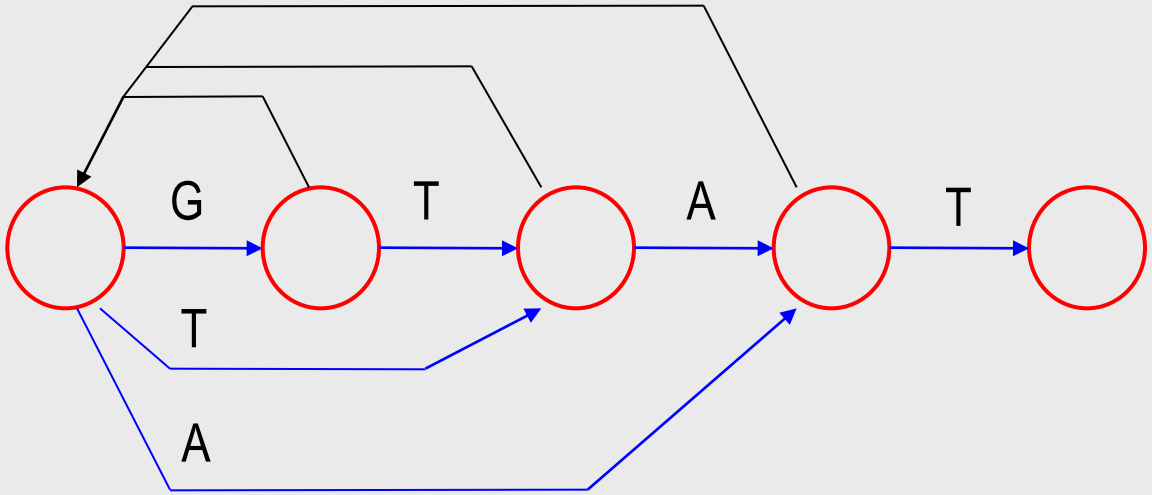
Factor Oracle of k strings

Given the Factor Oracle of GTATGTA



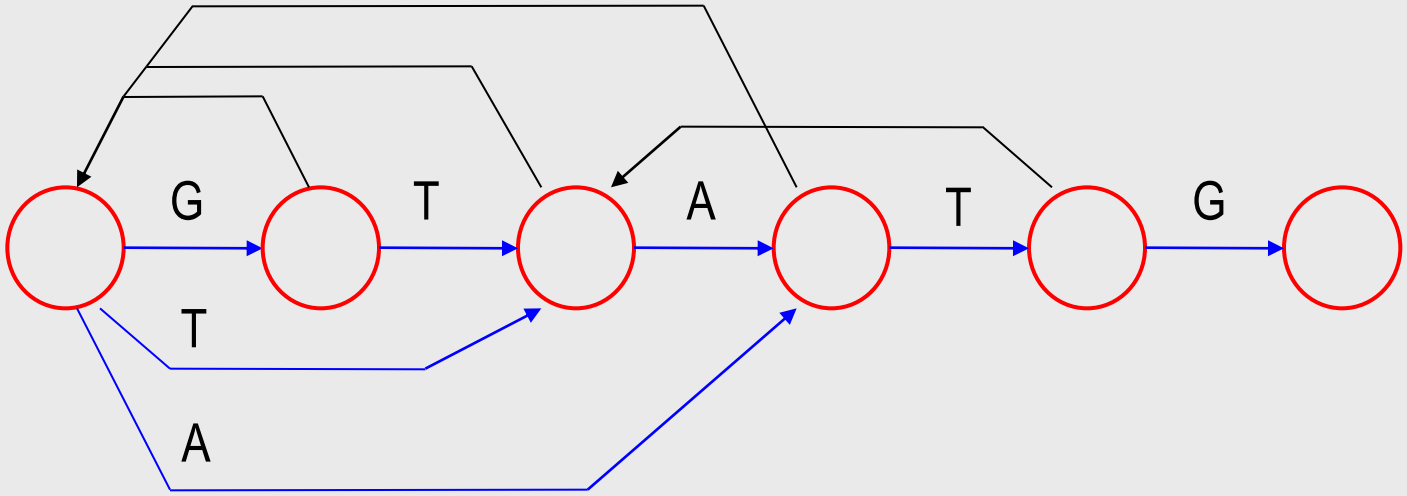
Factor Oracle of k strings

Given the Factor Oracle of GTATGTA



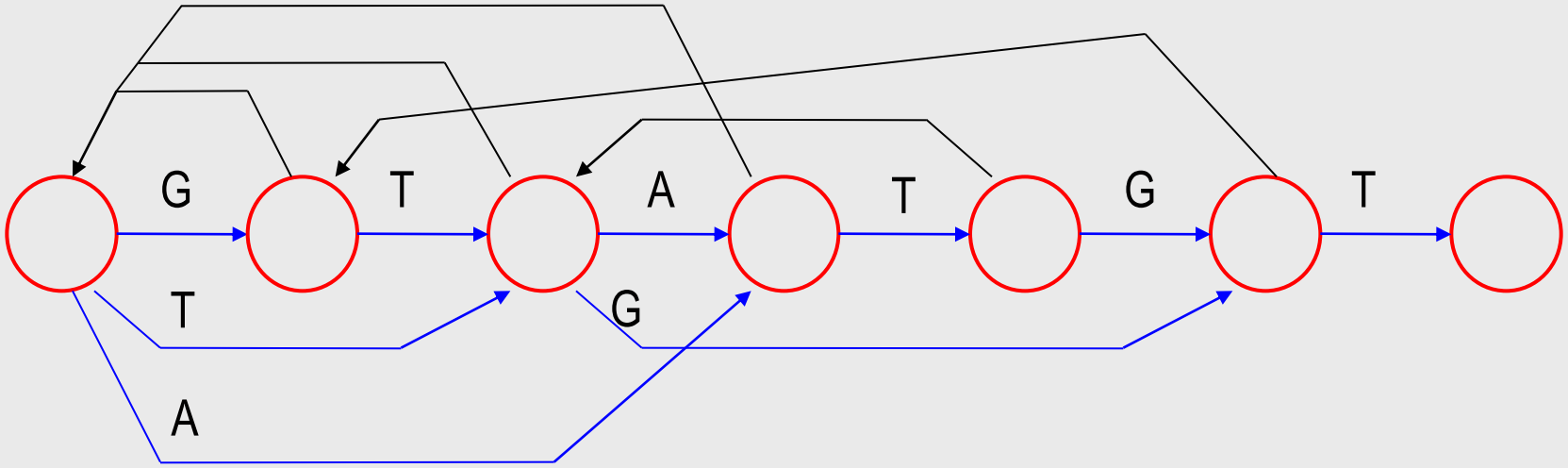
Factor Oracle of k strings

Given the Factor Oracle of GTATGTA



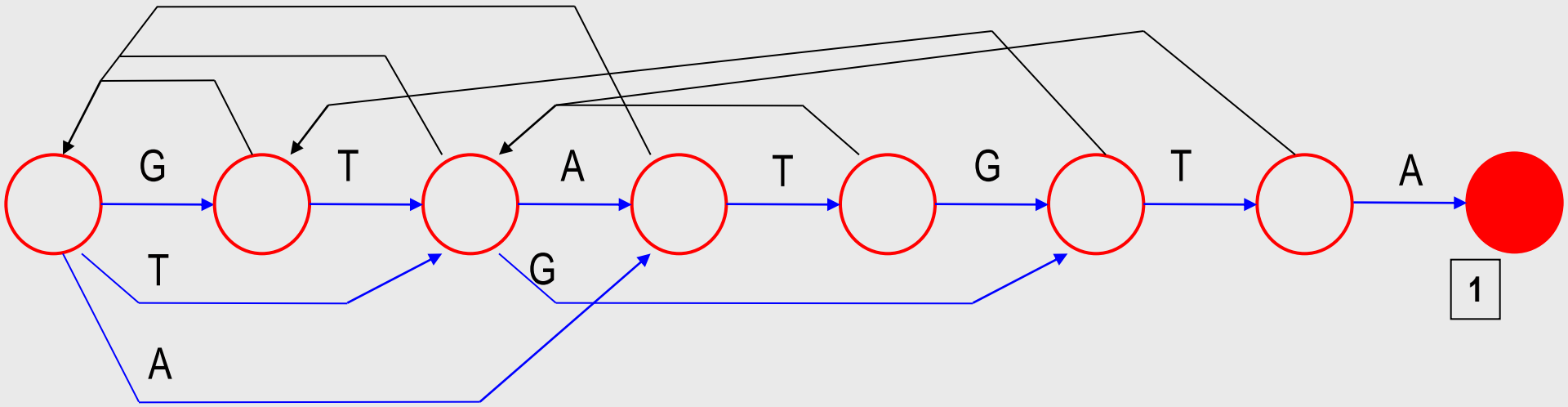
Factor Oracle of k strings

Given the Factor Oracle of GTATGTA



Factor Oracle of k strings

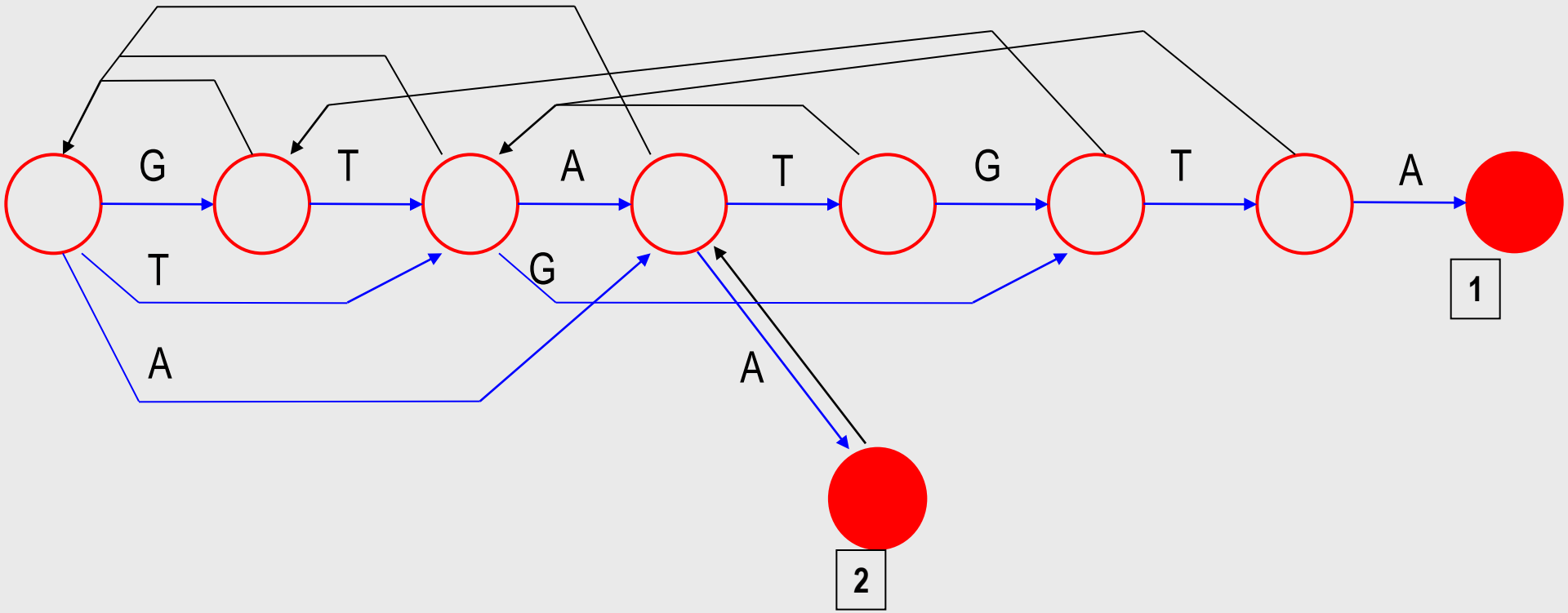
Given the Factor Oracle of GTATGTA



… we insert GTAA

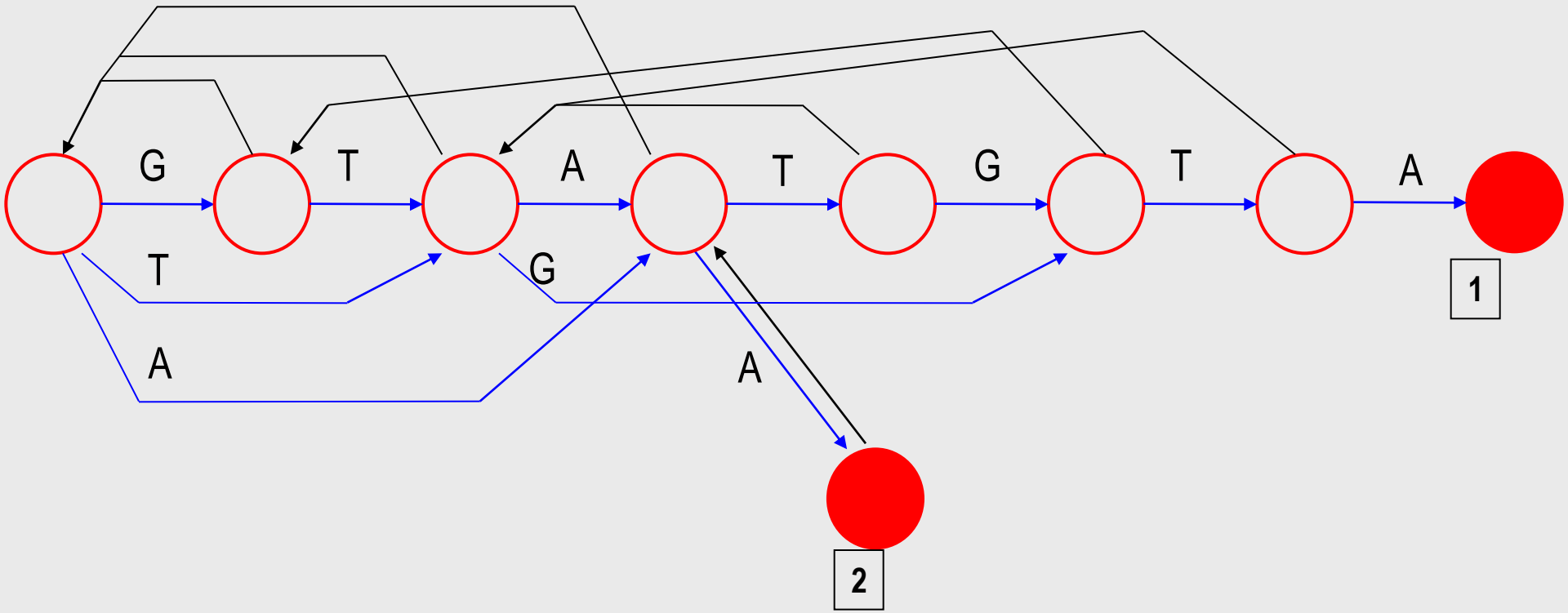
Factor Oracle of k strings

…inserting GTAA



Factor Oracle of k strings

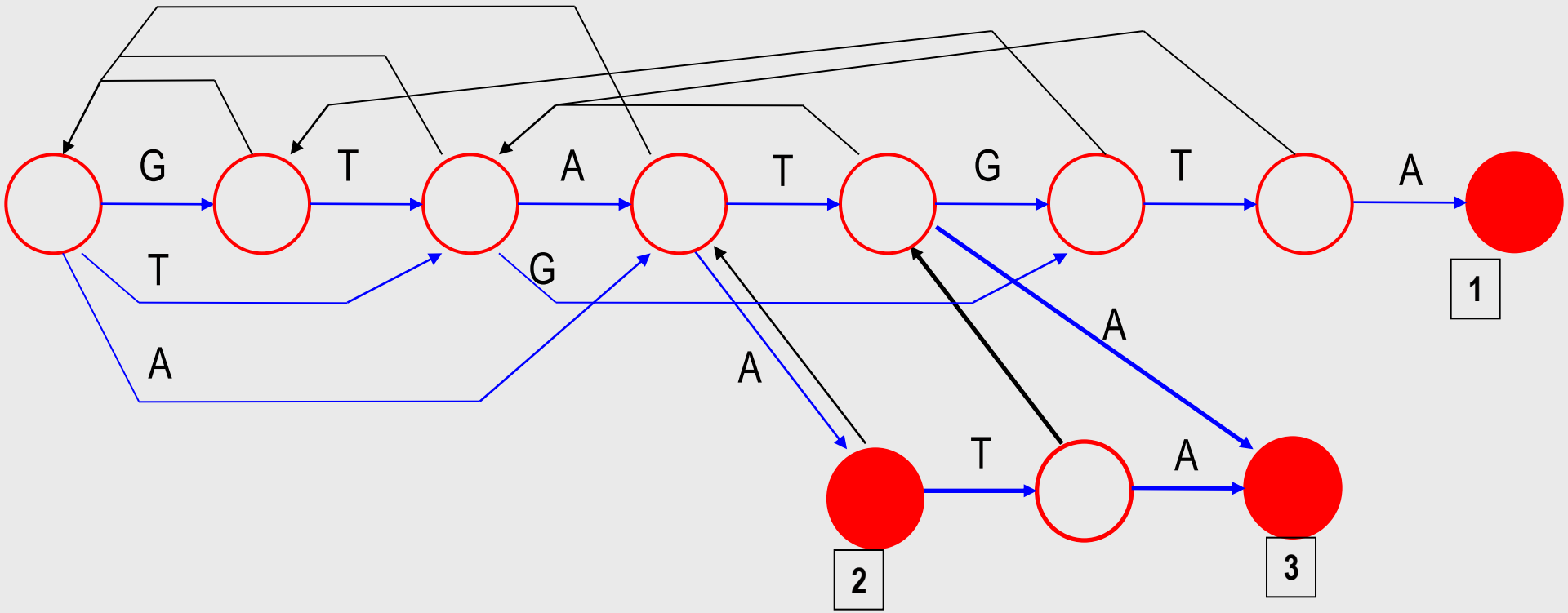
Given the AFO of GTATGTA and GTAA



… we insert TAATA

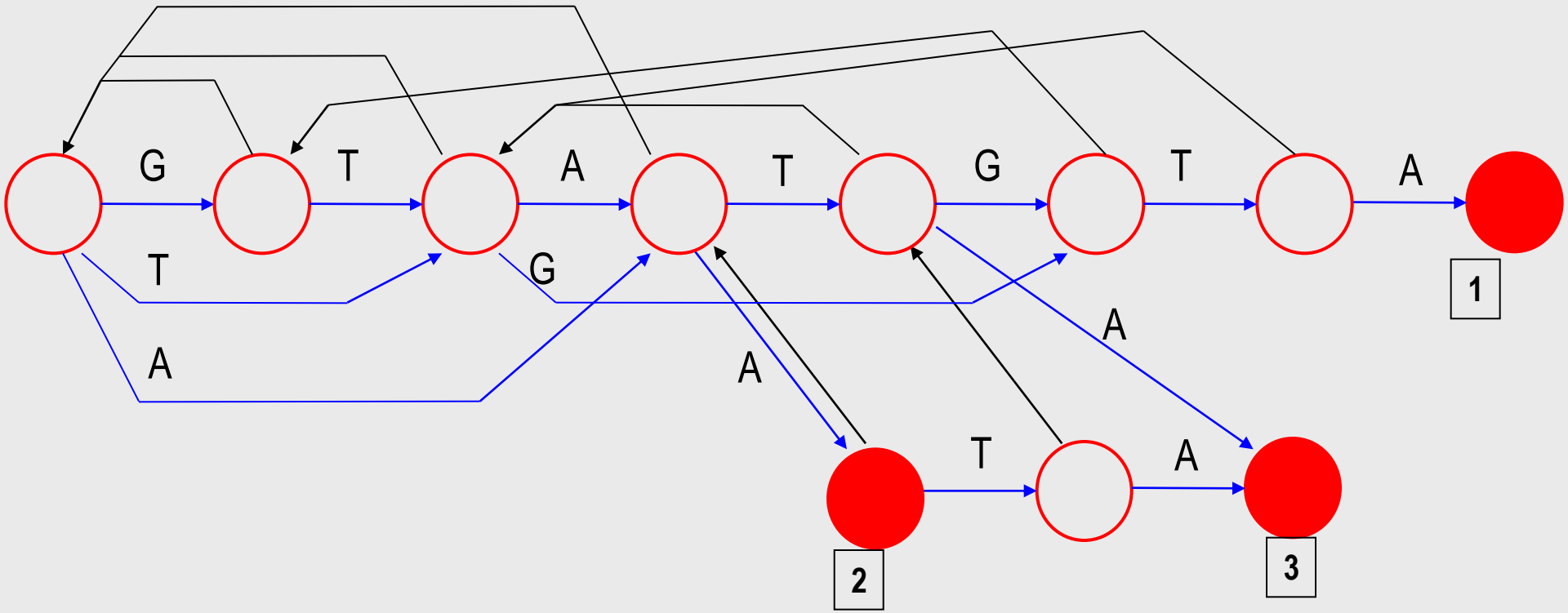
Factor Oracle of k strings

… inserting TAATA



Factor Oracle of k strings

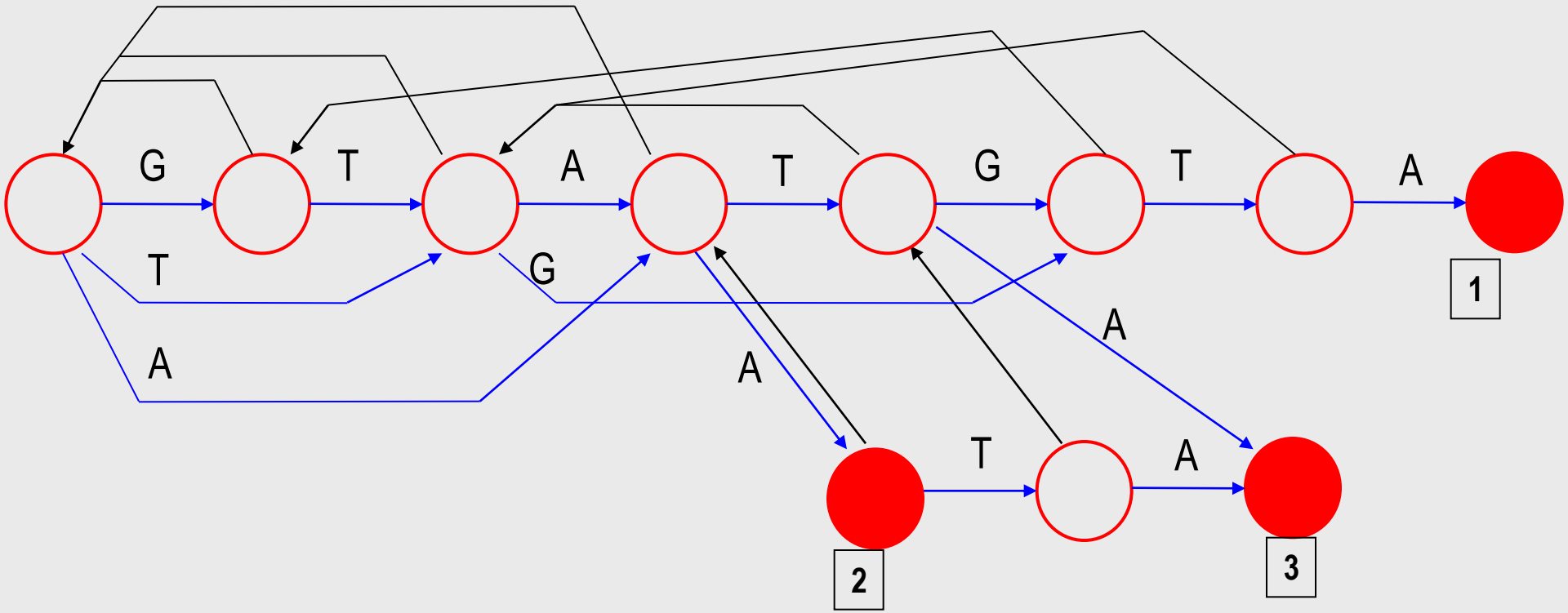
Given the AFO of GTATGTA, GTAA and TAATA



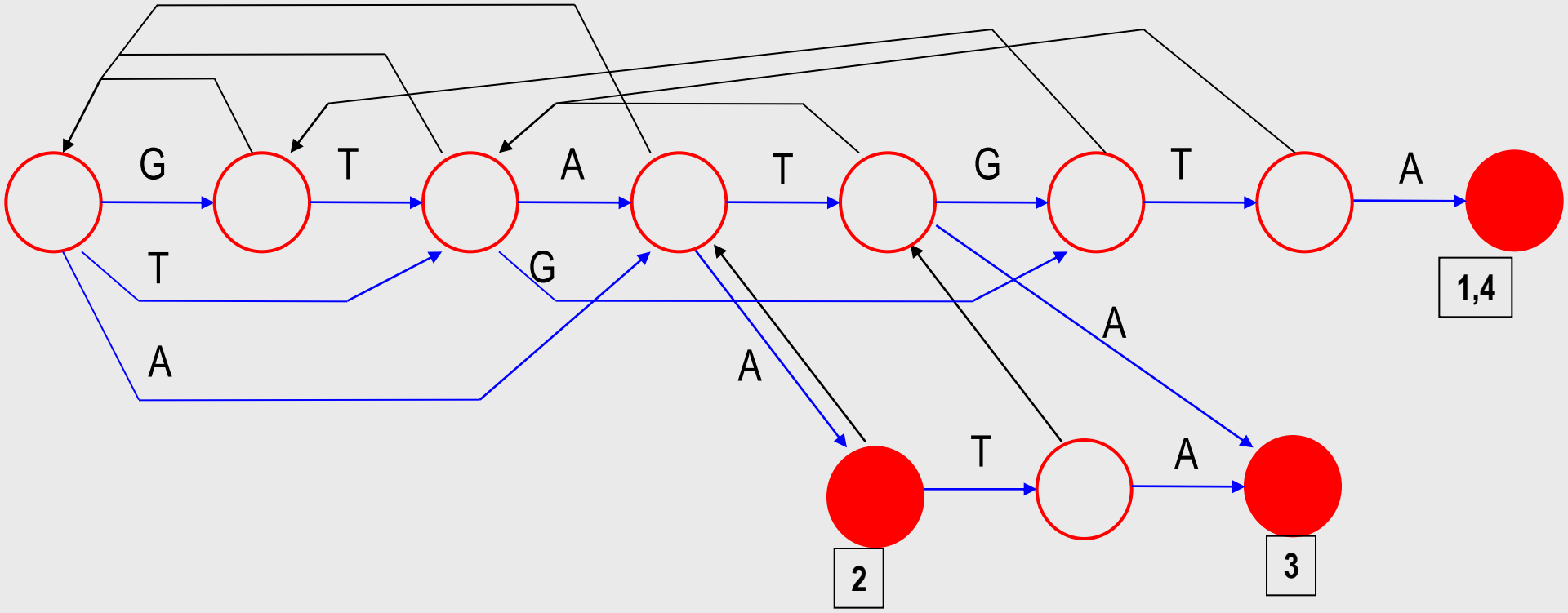
…we insert GTGTA

Factor Oracle of k strings

…inserting GTGTA



Factor Oracle of k strings



This is the Automata Factor Oracle of GTATGTA, GTAA, TAATA and GTGTA

SBOM algorithm

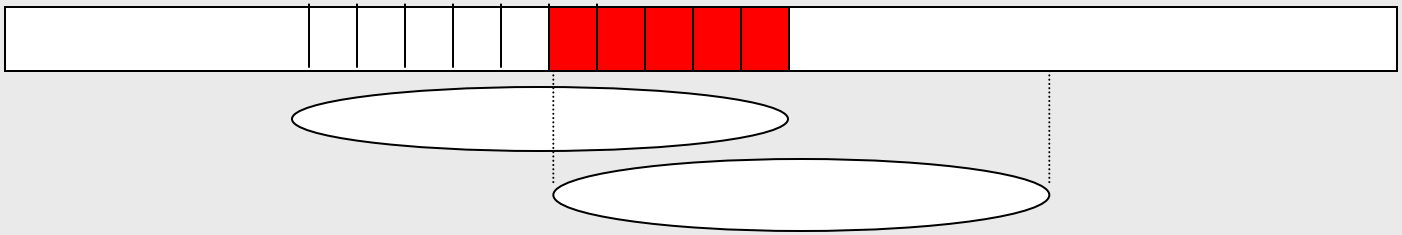
- **How the comparison is made?**

Text :

Pattern : Automata: Factor Oracle (Inverse patterns of length *lmin)*

Check if the suffix is a factor of any pattern

- **Which is the next position of the window?**

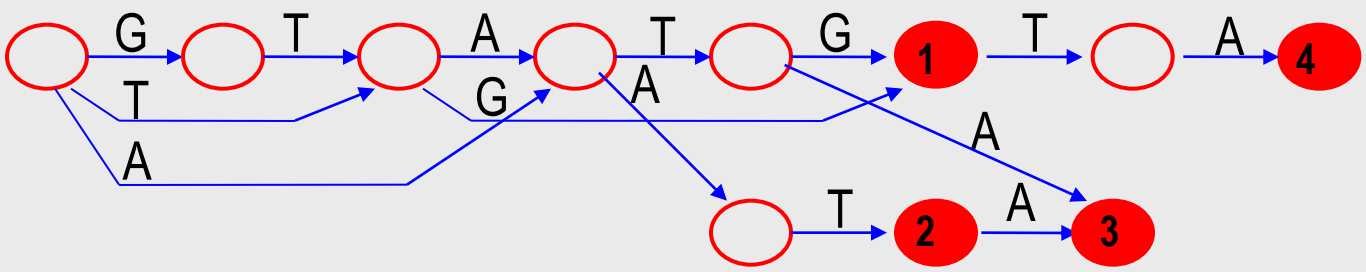


The position determined by the last character of the text
with a transition in the automata

SBOM algorithm: example

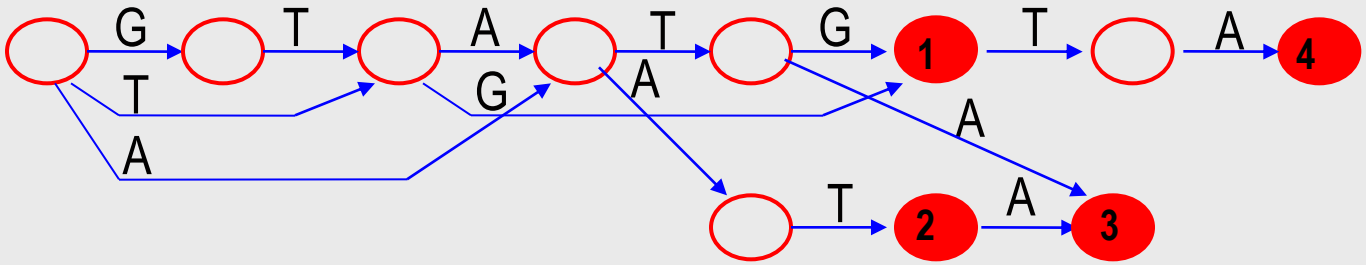
We search for the patterns
ATGTATG, TAATG,TAATAAT i AATGTG

… the we build the Automata Factor Oracle of
GTATG, GTAAT, TAATA and GTGTA
of length *lmin=5*

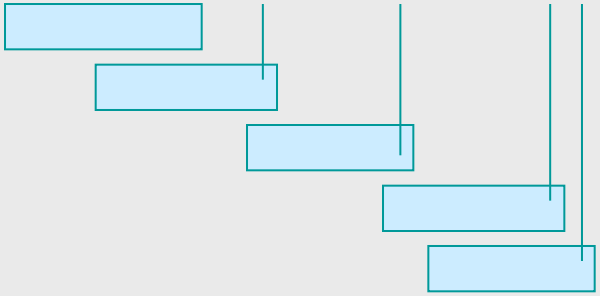


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

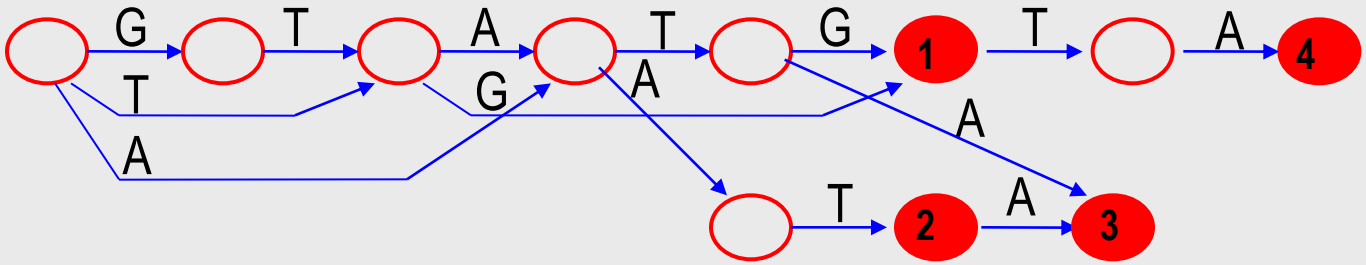


text: ACATGCTAGCTATAATAATGTATG

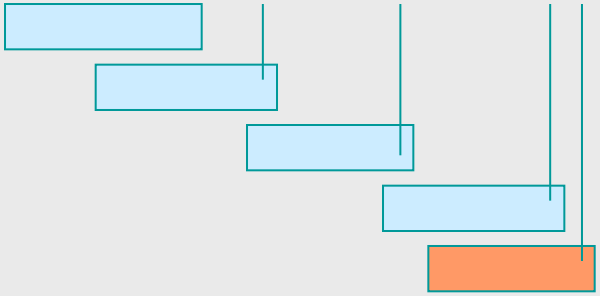


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

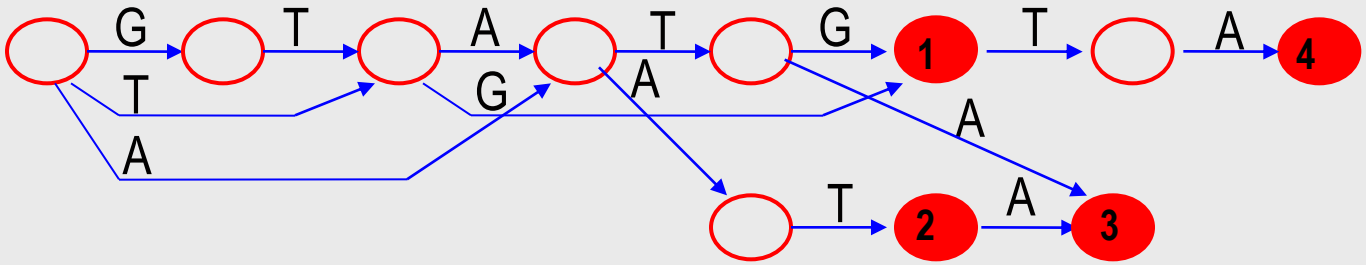


text: ACATGCTAGCTATAATAATGTATG

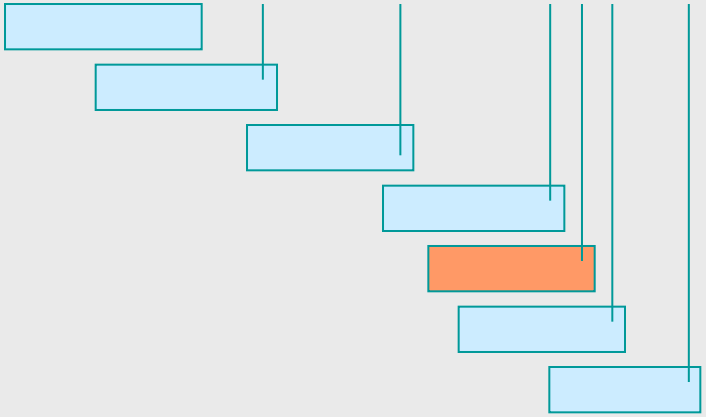


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

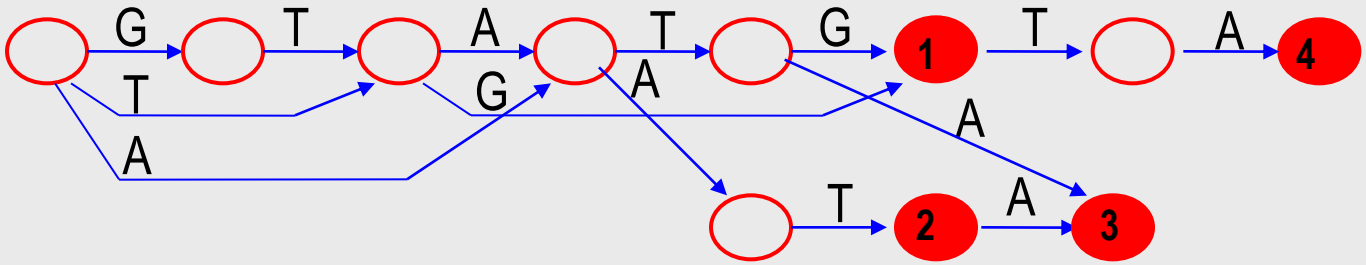


text: ACATGCTAGCTATAATAATGTATG

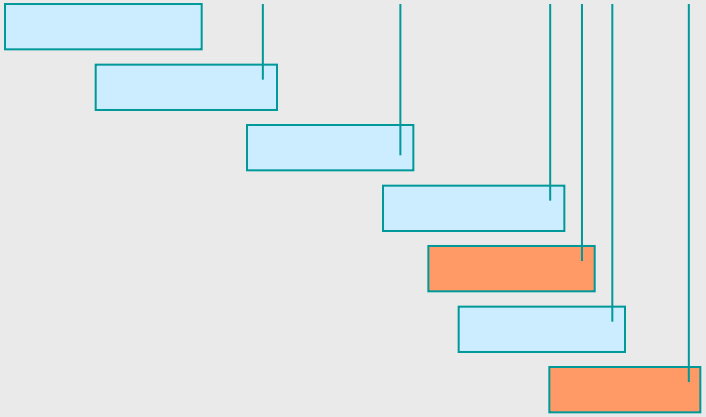


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

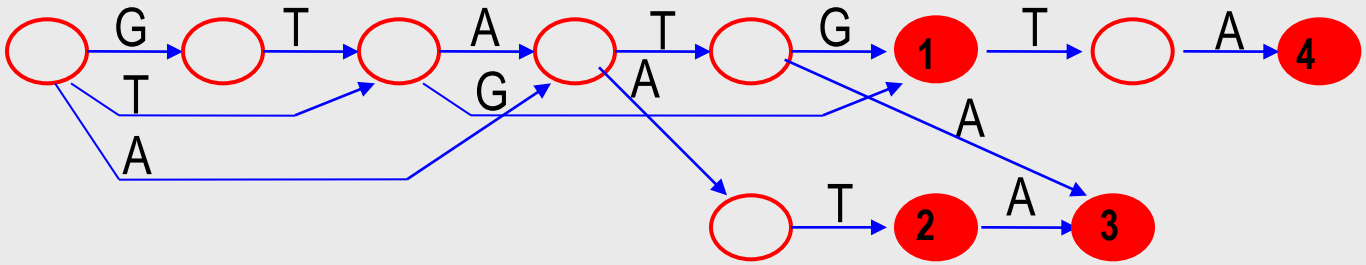


text: ACATGCTAGCTATAATAATGTATG

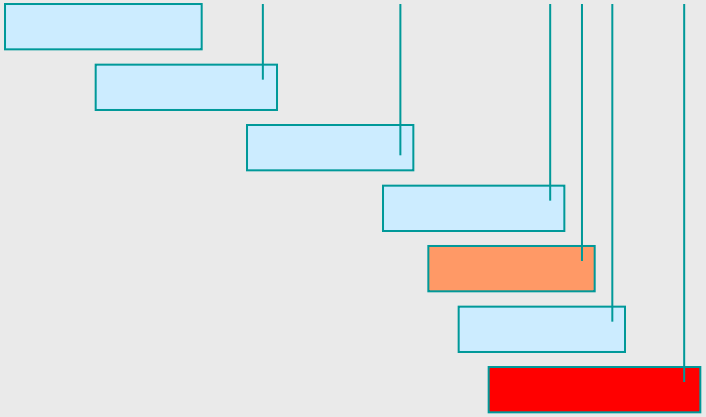


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

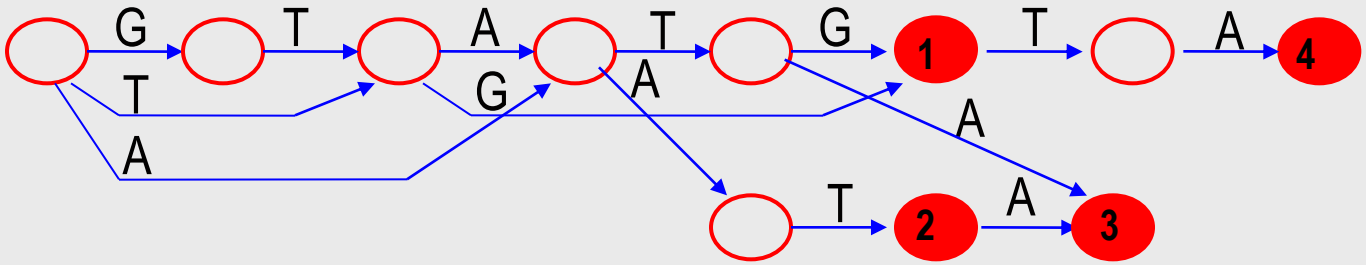


text: ACATGCTAGCTATAATAATGTATG

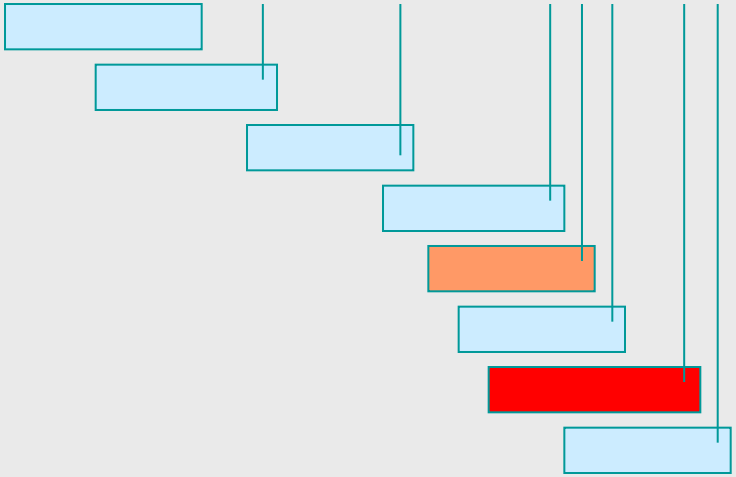


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

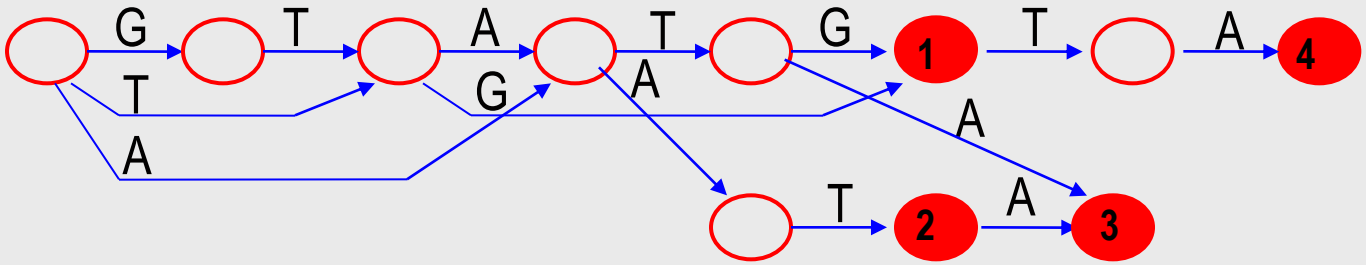


text: ACATGCTAGCTATAATAATGTATG

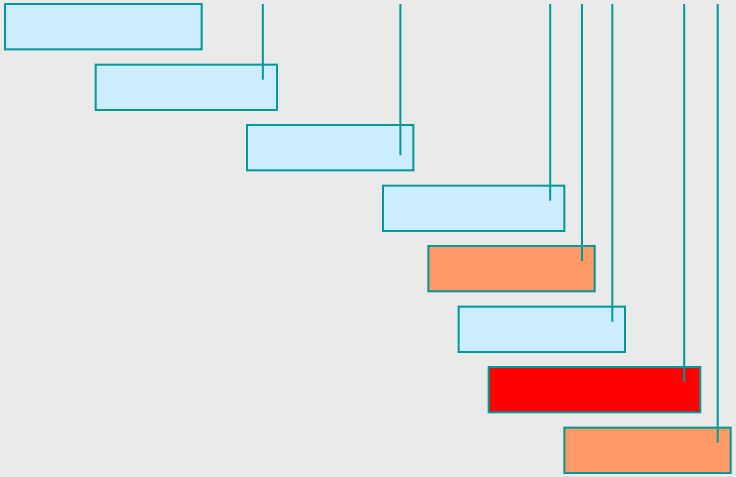


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG

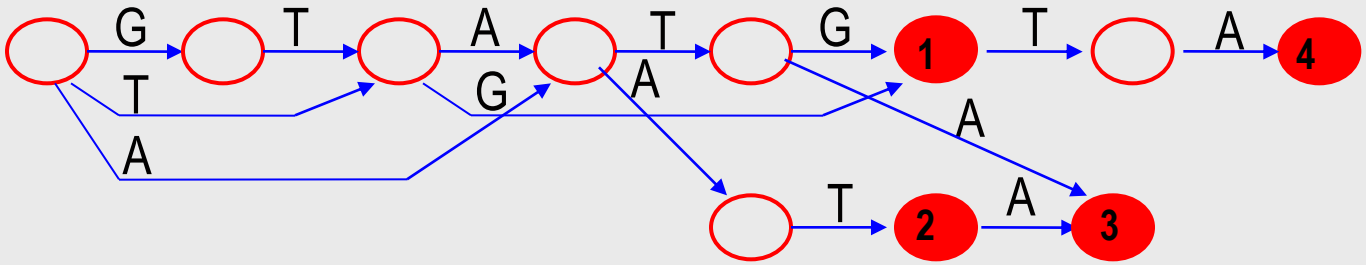


text: ACATGCTAGCTATAATAATGTATG

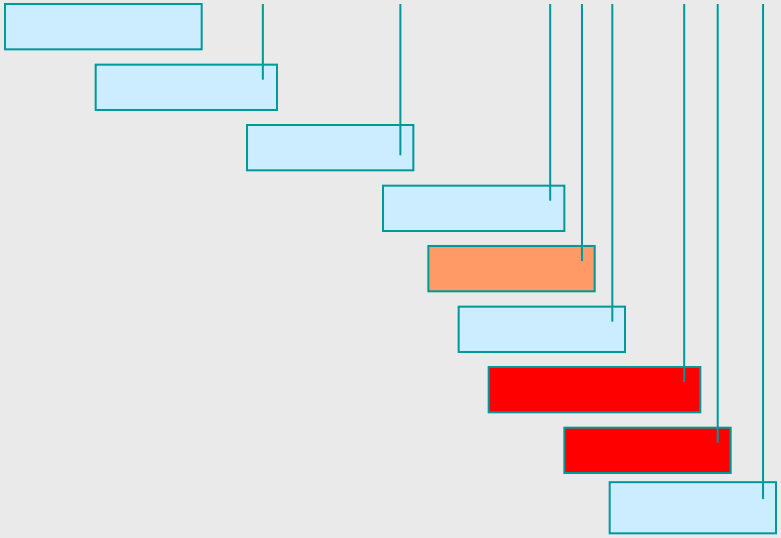


SBOM algorithm: example

Search for ATGTATG, TAATG,TAATAAT i AATGTG



text: ACATGCTAGCTATAATAATGT…



Multiple string matching