Datastructuren

Data Structures

Hendrik Jan Hoogeboom
Mark van den Bergh

Informatica – LIACS
Universiteit Leiden

najaar 2024

Table of Contents I

Contents

patroonherkenning

woord (P) zoeken in tekst (T)

veel algoritmen

– aantal letters bv binair
– herhalingen in patroon
– preprocessing meerdere malen gebruiken
  KMP één voorwaartse scan
  AC meerdere woorden tegelijk
– letters in P, vs letters in T, achterwaarts, hash

tekst (T) voldoet aan patroon (P)

– reguliere expressie [opdracht 3]
– eigenschap: palindroom

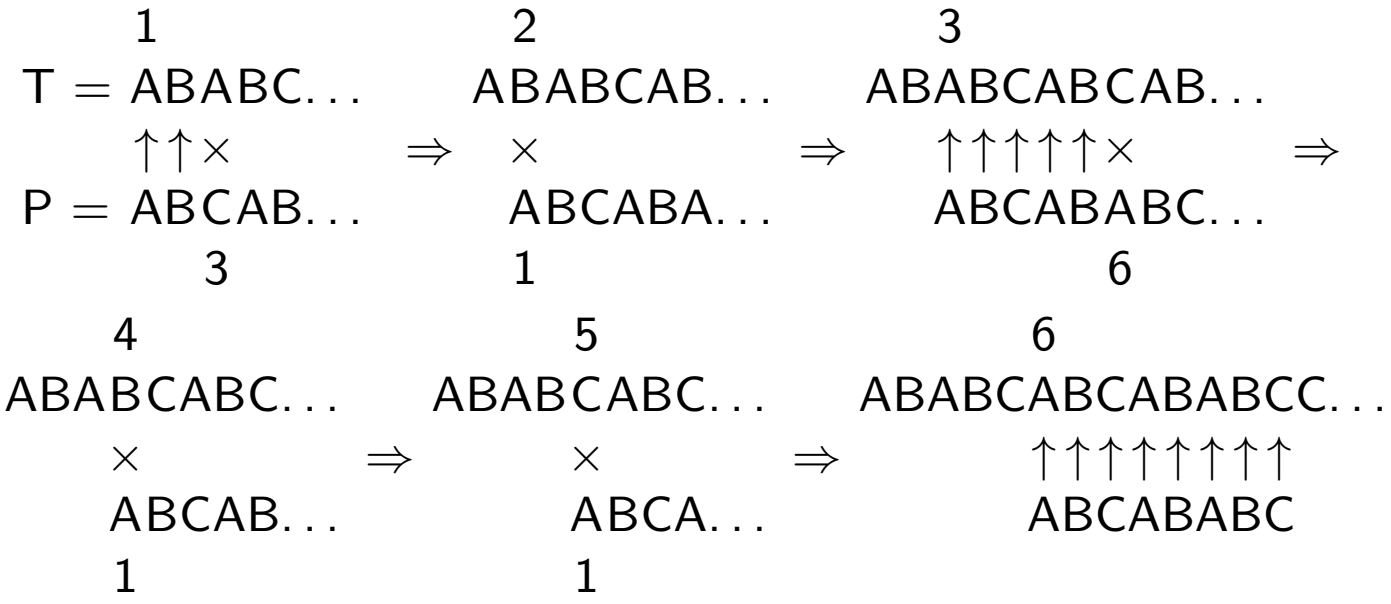
wel/geen fouten

bioinformatica

– alignment [om te lezen]
– gene prediction signalen (geen harde karakterisatie)

naive pattern matching

P = 'ABCABABC'
T = 'ABABCABCABABCC...'



Contents

KMP

Donald Knuth, James H. Morris, and Vaughan Pratt (1970,1977)

- *failure links*
- linear time preprocessing pattern
- search will never back-up in text

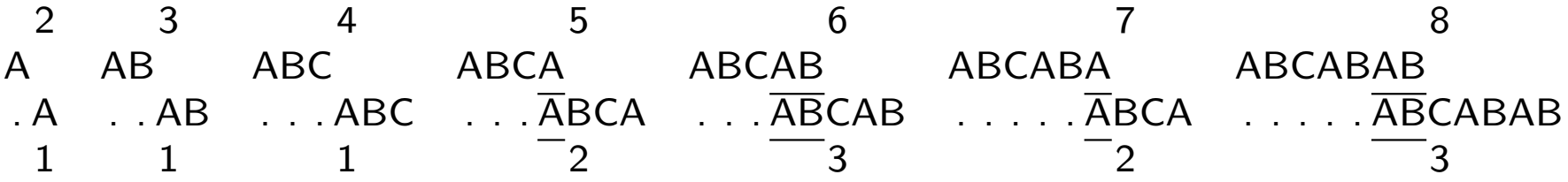
match pattern against itself *niet zo!*

$$T = \ldots ABCABAB?\ldots$$
$$P = \quad\; ABCABAB\times$$
$$\qquad\qquad\qquad\qquad\quad 8$$

$$\begin{array}{cccc}
\overset{2}{A}BCABAB\overset{8}{.} & A\overset{3}{B}CABAB\overset{8}{.} & AB\overset{4}{C}ABAB\overset{8}{.} & ABCA\overset{5}{B}AB\overset{8}{.} \\
.ABCABAB & ..ABCABAB & ...ABCABAB & ....ABCABAB \\
\times & \times & \times & \times
\end{array}$$

$$\begin{array}{c}
\qquad\qquad\quad \overset{6}{A}\overset{8}{B}CABAB. \\
.....\underline{AB}CABAB \\
\qquad\qquad\quad\; 3
\end{array}$$

failure links



k	1	2	3	4	5	6	7	8
P[k]	A	B	C	A	B	A	B	C
FLink[k]	0	1	1	1	2	3	2	3

at position k: the maximal $r < k$ such that

$$P_1 \dots P_{r-1} = P_{k-r+1} \dots P_{k-1}$$

mismatch at position k, then continue at position FLink[k]
(and *same* position in Text)

FLink[k] $= 0$: next position in Text, first position Pattern

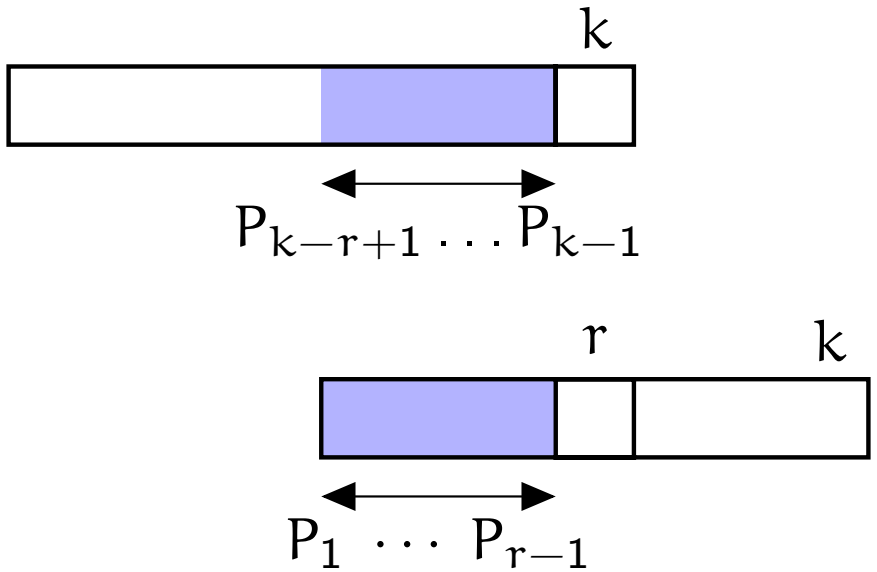
Failure link

maximal suffix = prefix

FLink[k] = r

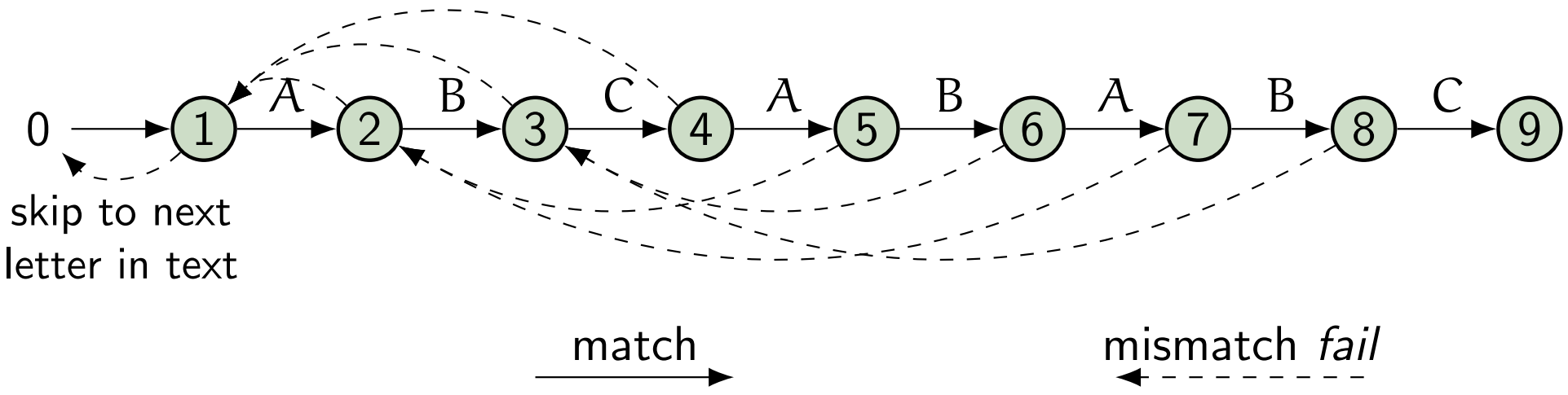
at position k: the maximal $r < k$ such that

$$P_1 \dots P_{r-1} = P_{k-r+1} \dots P_{k-1}$$



KMP failure links

k	1	2	3	4	5	6	7	8
P[k]	A	B	C	A	B	A	B	C
FLink[k]	0	1	1	1	2	3	2	3



KMP search

KMP using failure links

```
Pos = 1      // position in pattern
TPos = 1     // position in text
while ((Pos <= PatLen) and (TPos <= TextLen))
do  if (P[Pos] == Text[TPos])
    thenPos ++;
        TPos ++;
    else Pos = FLink[Pos]
        if (Pos == 0) // next position in text
        thenPos = 1
            TPos ++;
        fi
    fi
od
if (Pos > PatLen) // found?
thenPattern found at position TPos-PatLen+1 in Text
fi
```

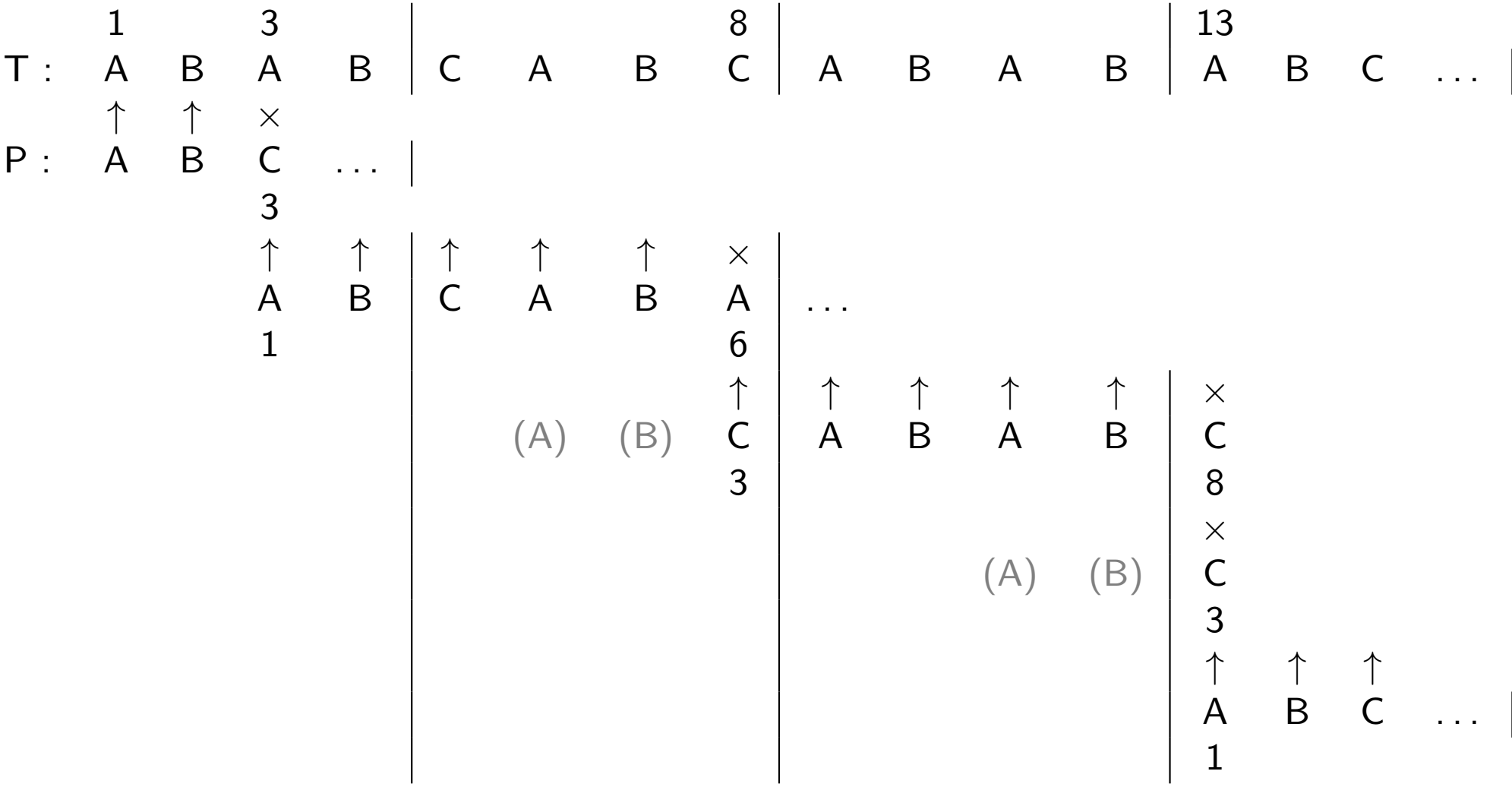
using KMP links

k	1	2	3	4	5	6	7	8
P[k]	A	B	C	A	B	A	B	C
FLink[k]	0	1	1	1	2	3	2	3

```
        1 3             3    8             6 8     13
T = ABABC...        ABABCABCAB...      BCABCABABABC...
        ↑↑×      ⇒      ↑↑↑↑↑×      ⇒      ↑↑↑↑↑×        ⇒
P = ABCAB...            ABCABABC...      ABCABABC
          3             1    6             3     8

     13              13
ABABABC..         BABABC..
     ×       ⇒     ↑ ↑↑        ⇒ ...
   ABCABABC        ABCABABC
     3             1
```

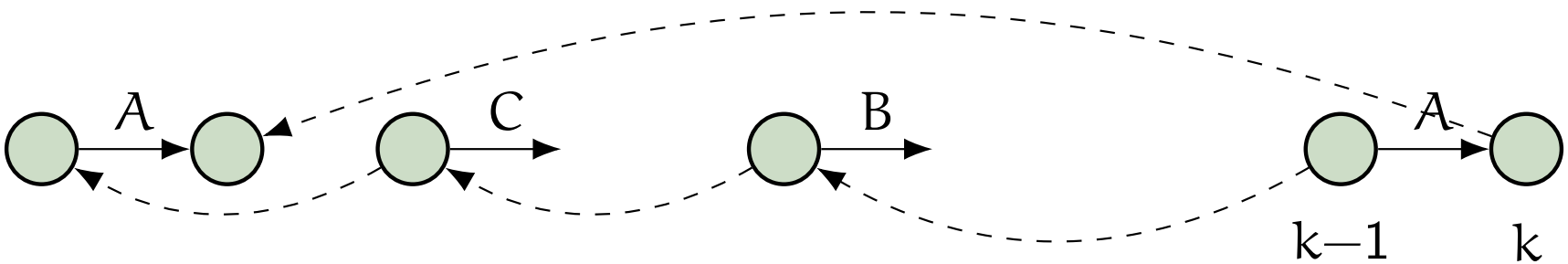
using KMP link (overview)



KMP failure links

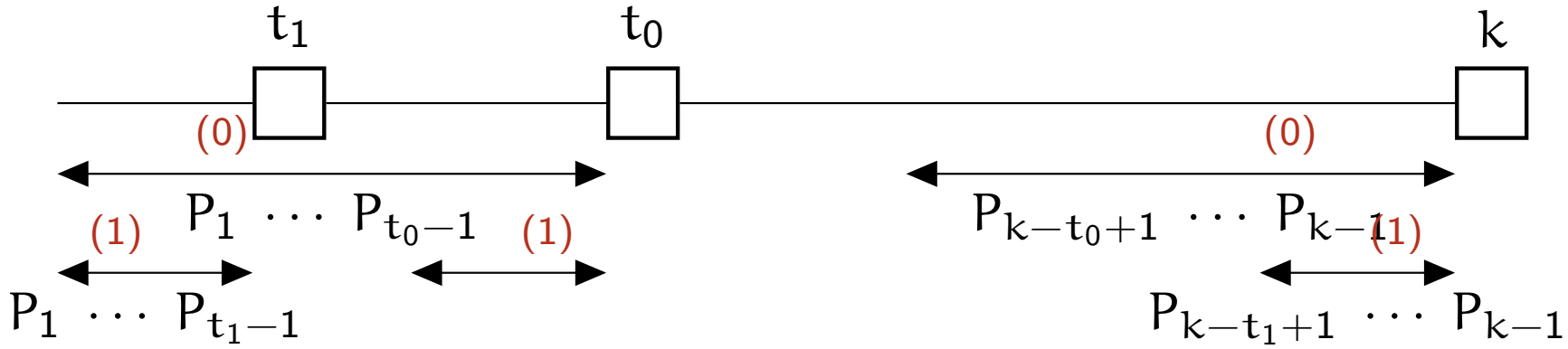
computing KMP failure links

```
Pos = 1 // position in pattern
FLink[1] = 0
for Pos = 2 to PatLen
do  Fail = FLink[Pos-1]
    while ( (Fail > 0) and (P[Fail] != P[Pos-1]) )
    do  Fail = FLink[Fail]
    od
    FLink[Pos] = Fail+1
od
```



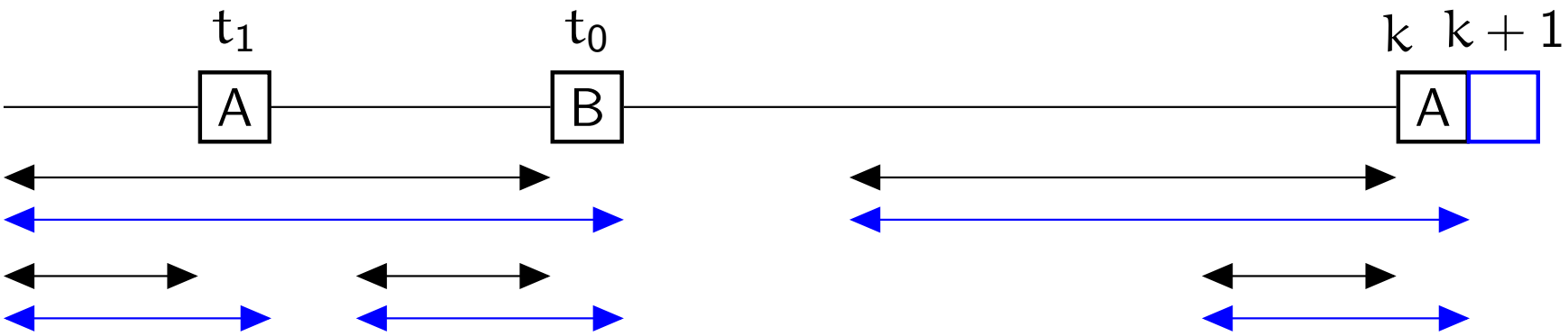
why does this work

all prefixes that are also suffix
can be found following failure links

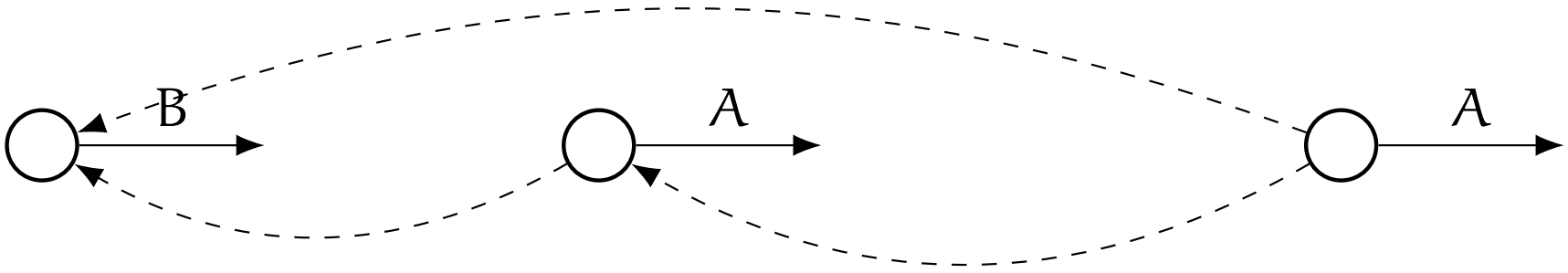


$\mathrm{FLink}[k] = t_0, \quad \mathrm{FLink}[t_0] = t_1$

why . . .



link to same character



updating KMP failure links

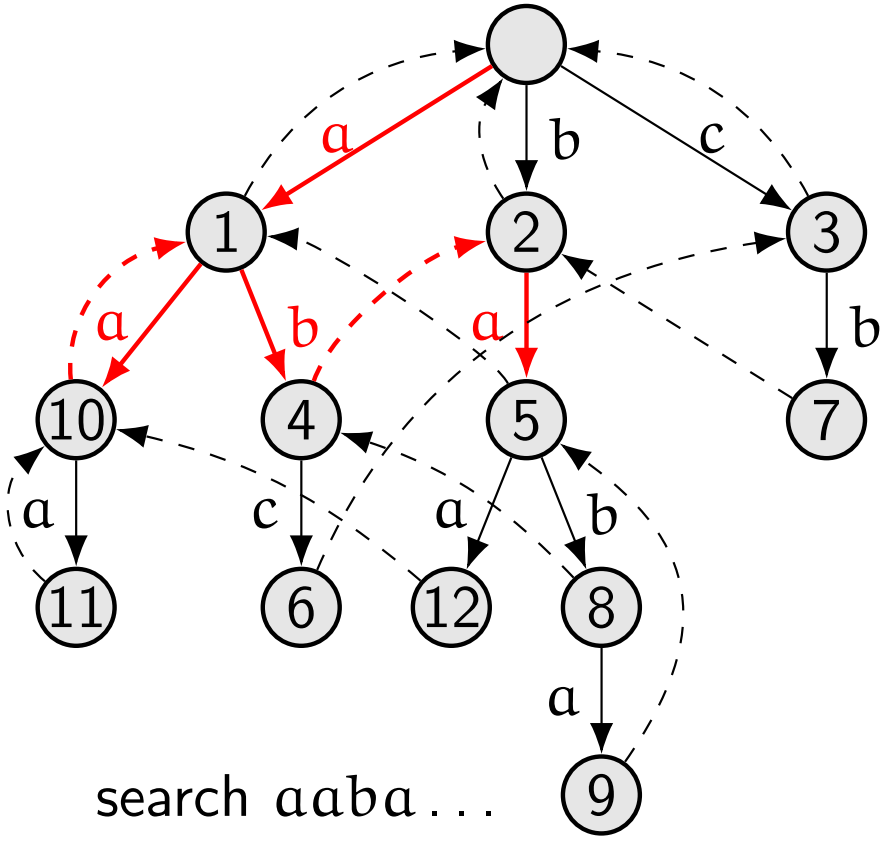
```
for Pos = 2 to PatLen
do  if ( P[Pos] == P[FLink[Pos]] )
    thenFLink[Pos] = FLink[FLink[Pos]]
    fi
od
```

k	1	2	3	4	5	6	7	8
P[k]	A	B	C	A	B	A	B	C
FLink[k]	0	1	1	1	2	3	2	3
FLink′[k]	0	1	1	0	1	3	1	1

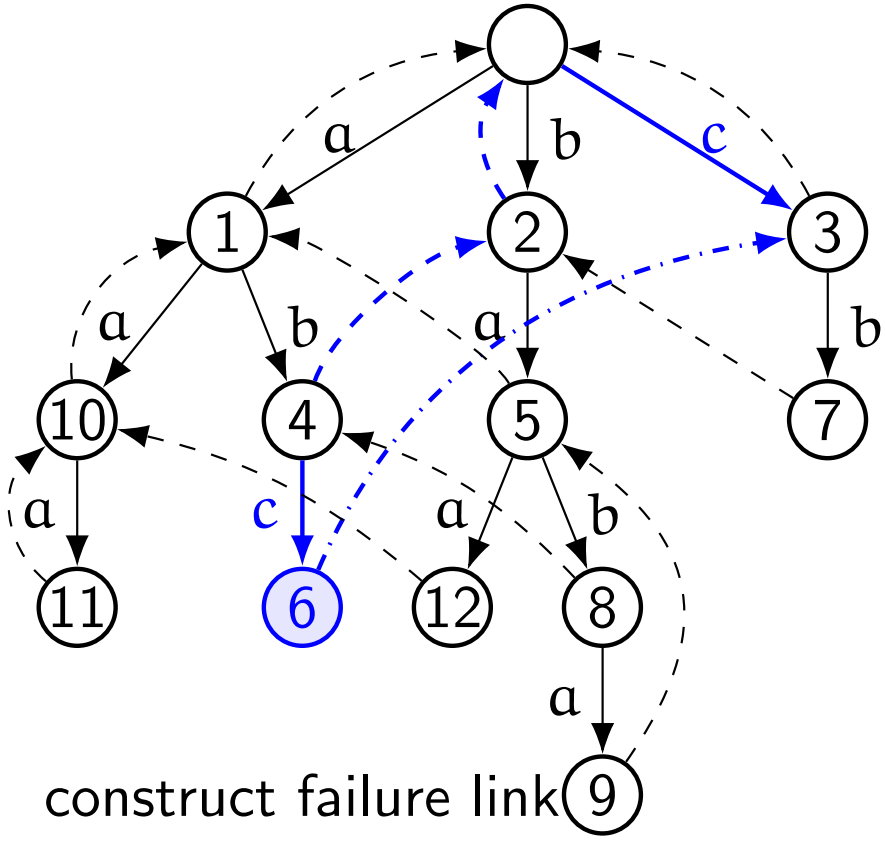
Contents

Aho-Corasick

$P = \{aaa, abc, baa, baba, cb\}$



search $aaba\ldots$



construct failure link

Contents

alignment

enzymes and their amino acids

```
82 TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
81 --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                        ****: .***:  * *:** * :****.:* *******..

82 PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
81 HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
    **** *:***********:***:**.: .***************    :  *.: :
```

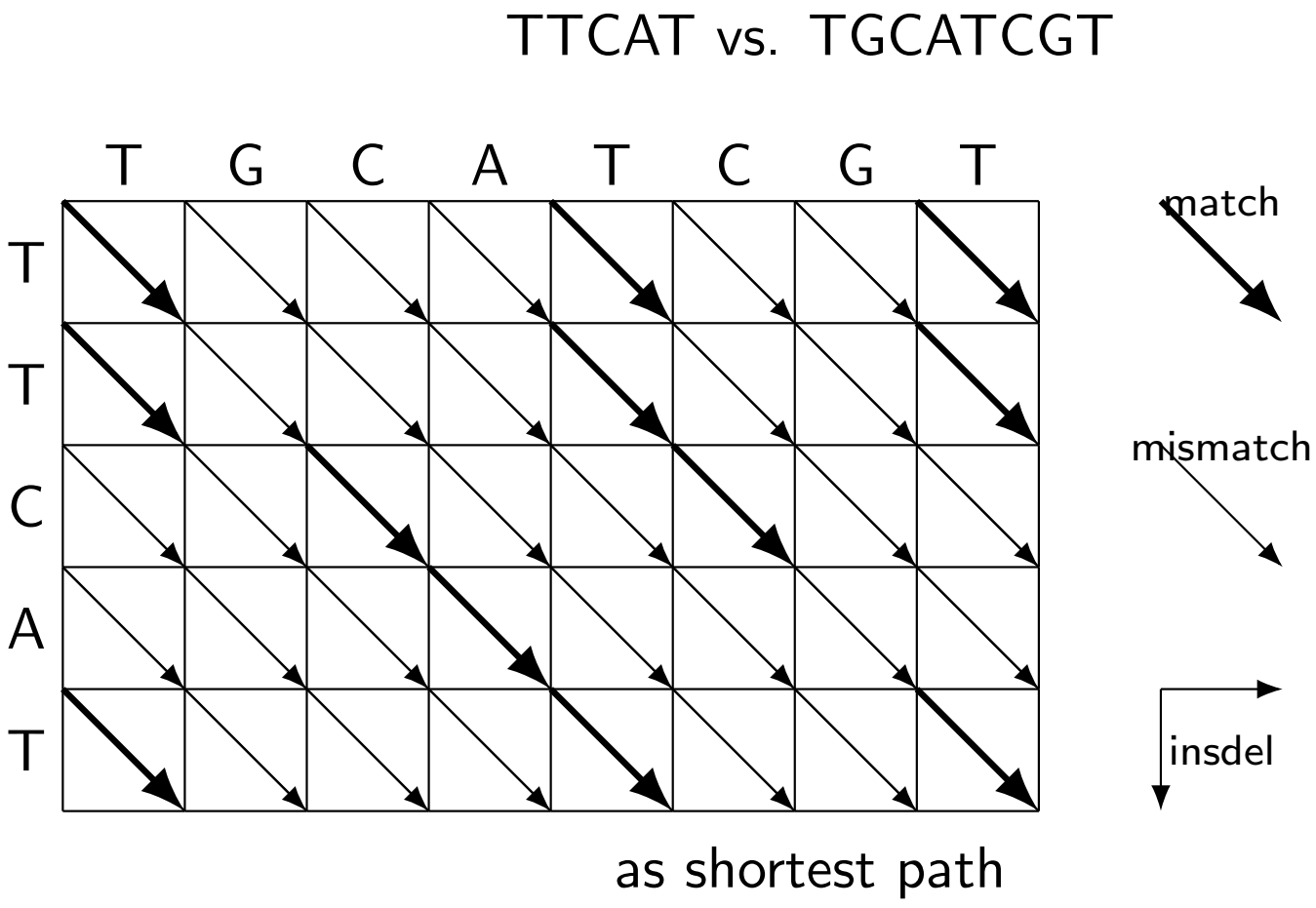
similarity TCAGACGATTG and TCGGAGCTG

TCAG-ACG-ATTG	TCAGACGATTG
TC-GGA-GC-T-G	TCGGA-GCT-G

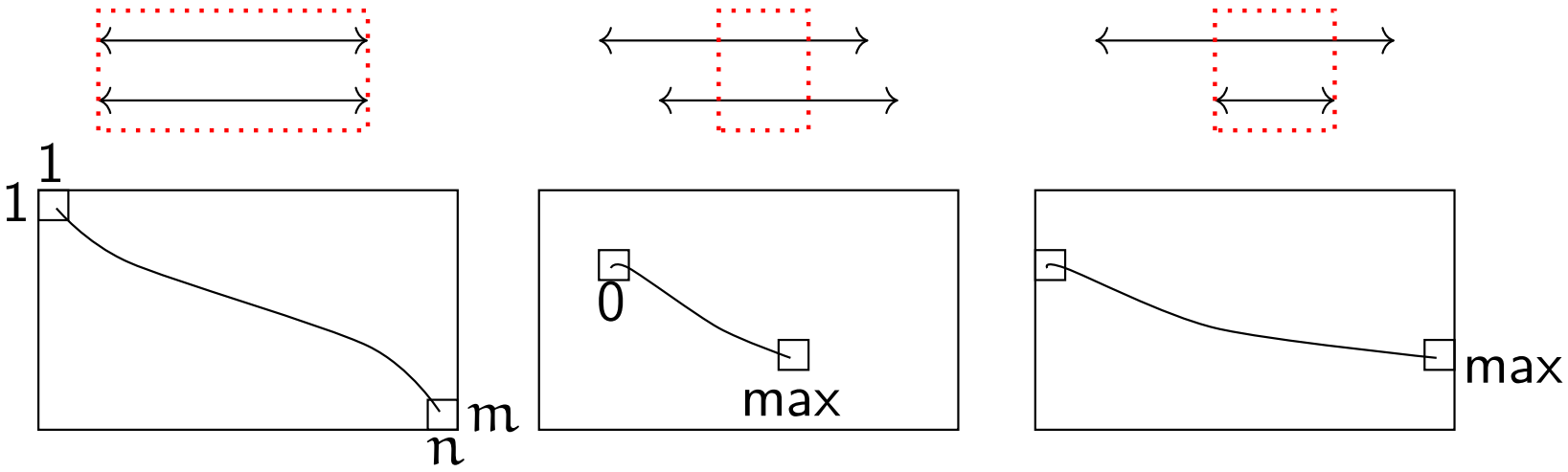
match, mismatch, insdel (gap)

G	A	-	A
G	G	G	-

global alignment



global versus local alignment



Levenshtein distance (1966)
Needleman-Wunsch (1970)
Smith-Waterman (1981)

end.