Pattern matching optimal

Gilles Didier

Institut de Mathématiques de Marseille
Aix-Marseille Université

Journées du GDR IM

Montpellier - 16 mars 2017

Asymptotic speed

Asymptotic speed of an algorithm **A** w.r.t a pattern w and a text model $\mathcal{M}$:

$$\mathrm{AS}_{w,\mathcal{M}}(\mathbf{A}) = \lim_{n\to\infty} \sum_{t\in\mathcal{A}^n} \frac{|t|}{a_{\mathbf{A},w}(t)} p_{\mathcal{M}}(t)$$

$\mathcal{A}$: alphabet

$p_{\mathcal{M}}(t)$: probability of t under $\mathcal{M}$

$a_{\mathbf{A},w}(t)$: number of text accesses performed by **A** to search w in t

- Compute the asymptotic speed of usual algorithms w.r.t. patterns w and iid models
- Given a pattern w and an iid model, return a *de novo* optimal algorithm (i.e. with the greatest asymptotic speed)
- Evaluate the approach from theoretical and practical points of view

Matching machines

Matching machines

Matching machines

Matching machines

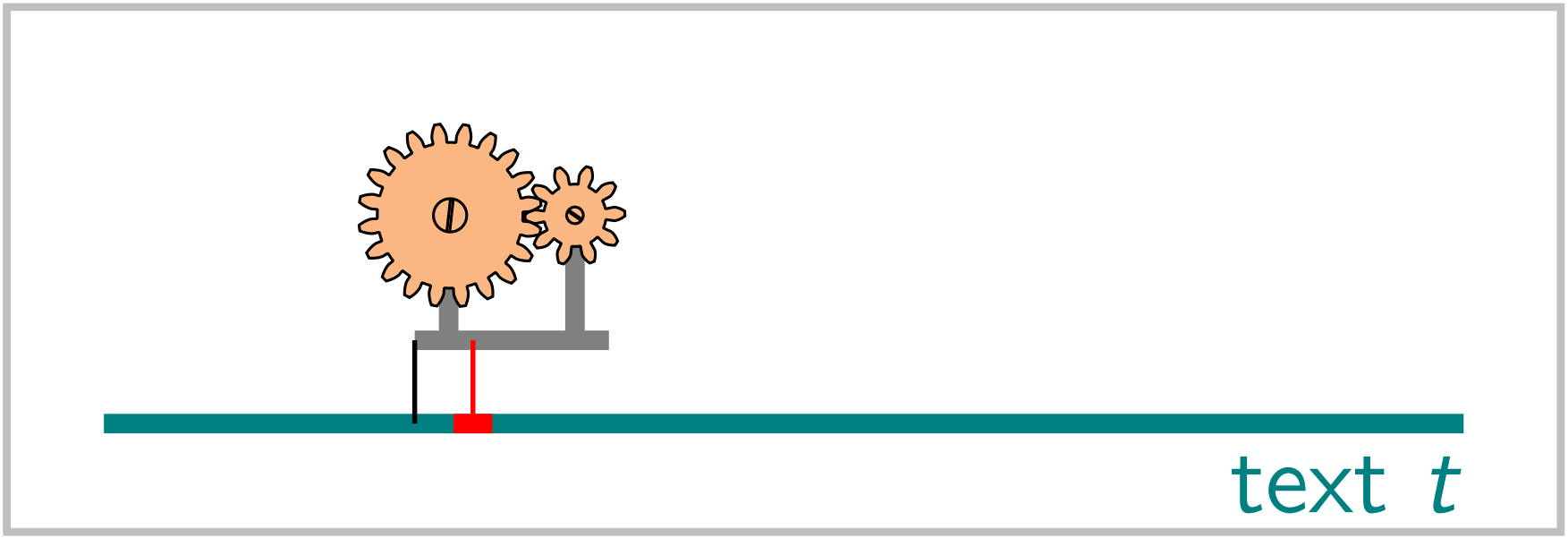
Matching machines

Matching machines

Matching machines

Matching machines

Matching machines

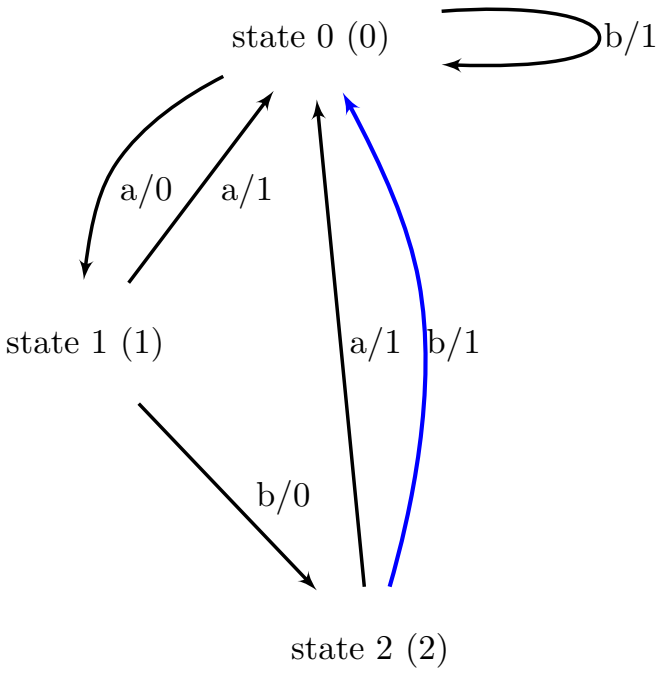


Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



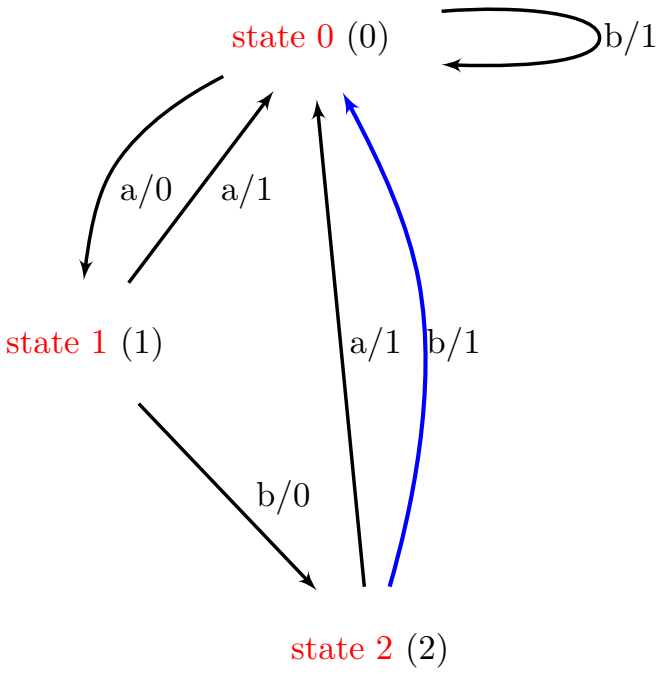
Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



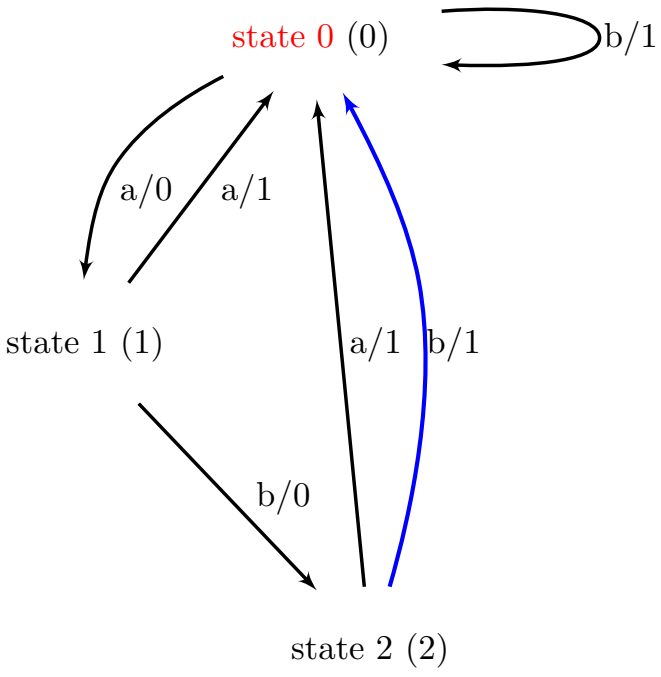
Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



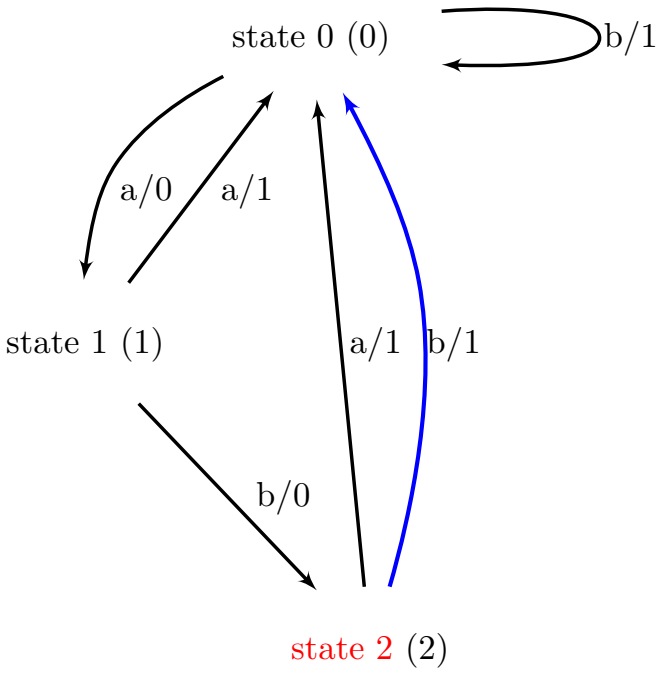
Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



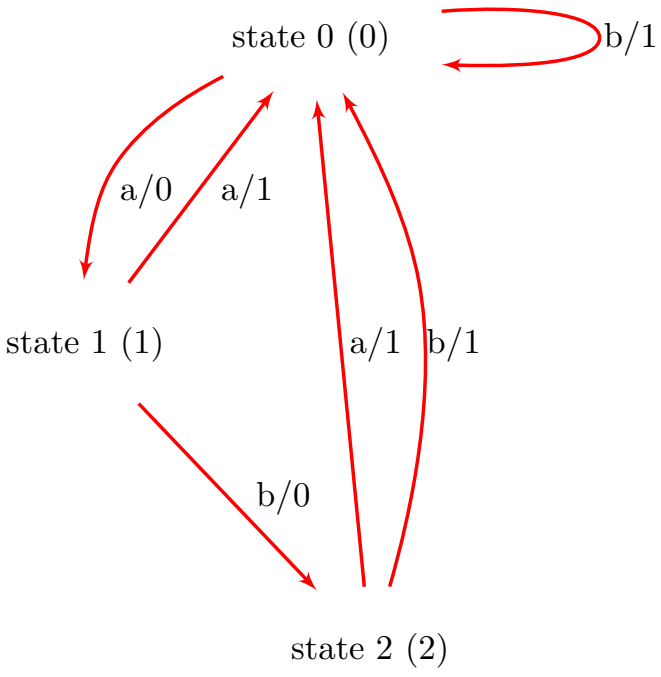
Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



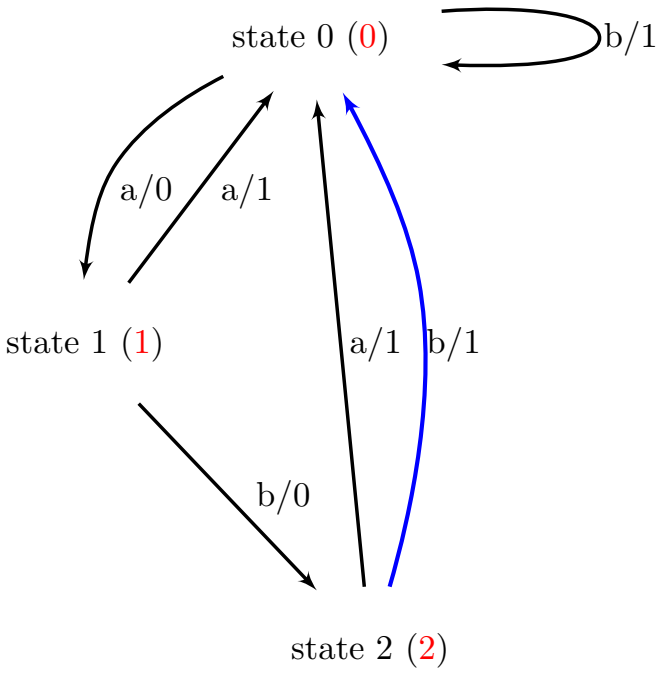
Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



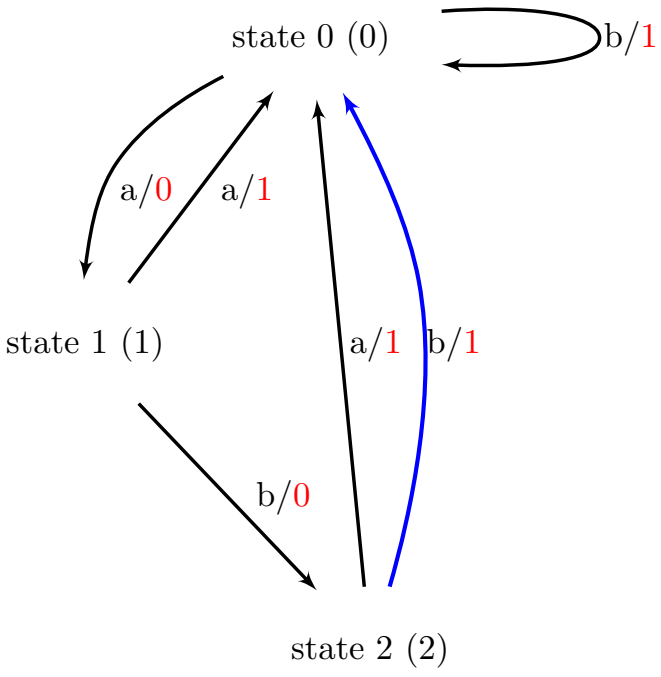
Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Matching machines

w-matching machine Γ (w : pattern on an alphabet $\mathcal{A}$)

6-uple $(Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where

- Q is a finite number of states,
- $o \in Q$ is the initial state,
- $F \subset Q$ is the subset of pre-match states,
- $\boldsymbol{\delta} : Q \times \mathcal{A} \to Q$ is the transition state function,
- $\boldsymbol{\alpha} : Q \to \mathbf{N}$ is the next-position-to-check function,
- $\boldsymbol{\gamma} : Q \times \mathcal{A} \to \mathbf{N}$ is the shift function.



Order of Γ : $O_\Gamma = \max_{q \in Q}\{\boldsymbol{\alpha}(q)\}$

Generic algorithm

input : a w-matching machine $\Gamma = (Q, o, F, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ and a text t
output: all the occurrence positions of w in t (hopefully)

$(q, p) \leftarrow (o, 0)$;
while $p \leq |t| - |w|$ **do**
  if $q \in F$ **and** $t_{p+\alpha(q)} = w_{\alpha(q)}$ **then**
    print " occurrence at position p ";
  $(q, p) \leftarrow (\delta(q, t_{p+\alpha(q)}), p + \gamma(q, t_{p+\alpha(q)}))$;

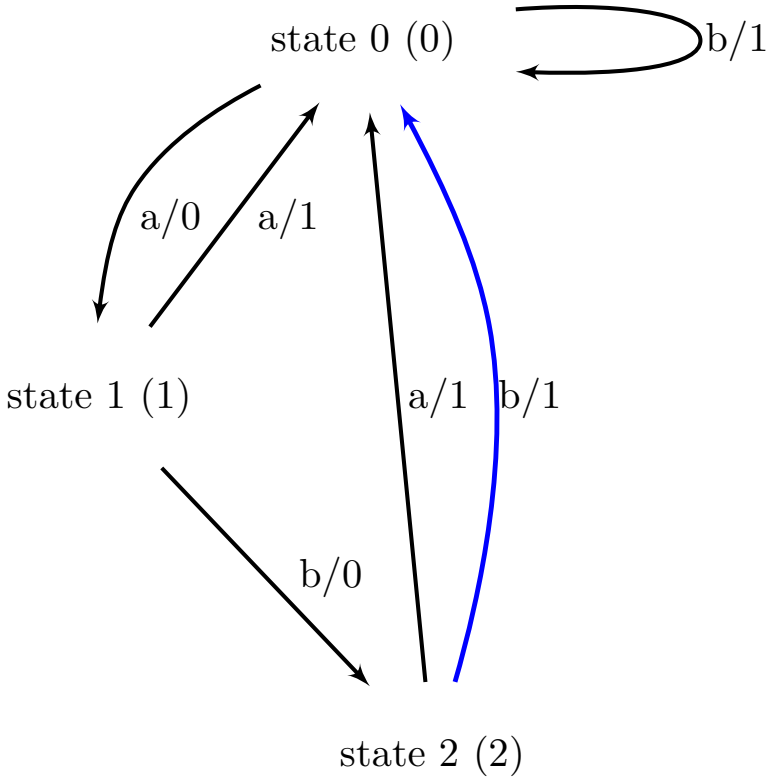
Γ is *valid* if, for all texts t, the execution of the generic algorithm on the input (Γ, t) outputs all, and only the occurrence positions of w in t.

Usual algorithms

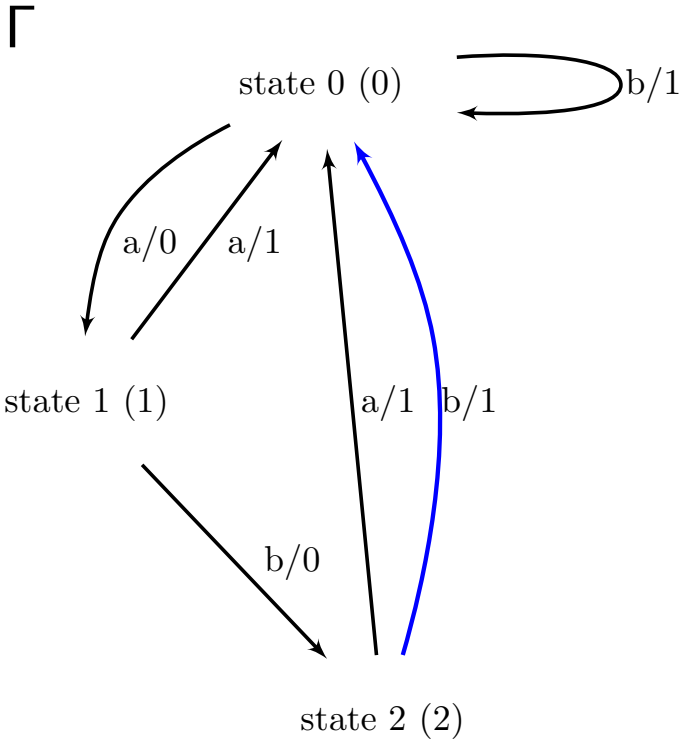
Claim

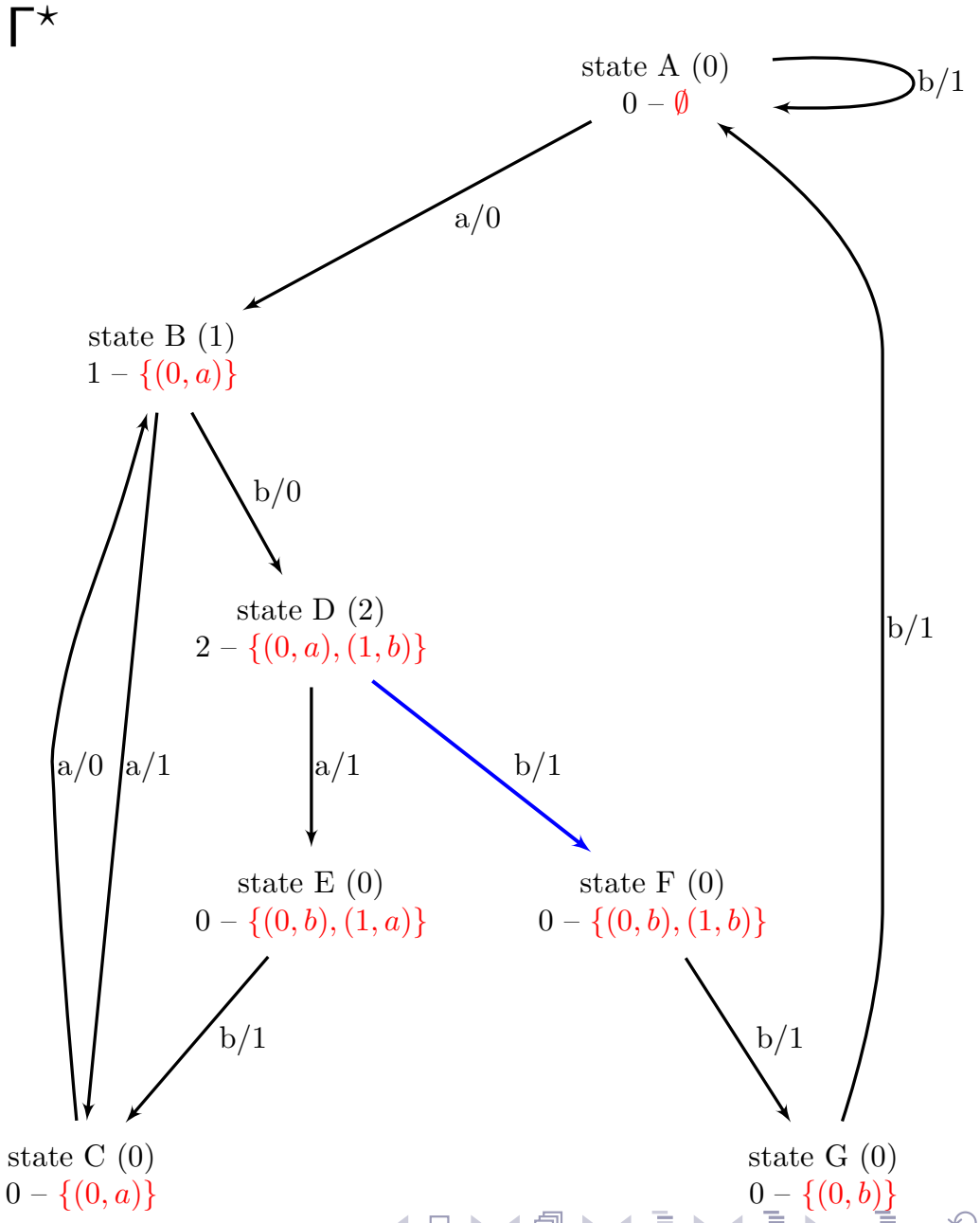
For all usual pattern matching algorithms and all patterns w, there exists a w-matching machine Γ which is such that the number of text accesses of the generic algorithm on the input (Γ, t) is equal to that of the pattern matching algorithm on the input (w, t).

Naive algorithm with *abb*



Full-memory expansion





Standard matching machine

Remark

For all texts t, the generic algorithm performs the exact same text accesses on the inputs (Γ, t) and $(\Gamma^\star, t)$.

$$\Gamma \text{ is } \textit{standard} \text{ if } \Gamma = \Gamma^\star$$

Theorem

If t is iid and Γ is standard then the sequences of states parsed by the generic algorithm on the input (Γ, t) is Markov.

Asymptotic speed

Asymptotic speed of a w-matching machine Γ under an iid model π

$$\mathrm{AS}_\pi(\Gamma) = \lim_{n\to\infty} \sum_{t\in\mathcal{A}^n} \frac{|t|}{a_\Gamma(t)} p_\pi(t)$$

Number of text accesses = Number of iterations = $a_\Gamma(t)$

$$|t| \simeq \sum_{i=0}^{a_\Gamma(t)} \gamma^\star(q_i, t_{p_i})$$

q_i the current state of Γ at the ith iteration
p_i the position read at the ith iteration

The asymptotic speed of Γ is the expectation of the shift under Markov model $\mathcal{M}_{\Gamma^\star}$ from $\Gamma^\star$ and π

$$\mathrm{AS}_\pi(\Gamma) = \mathrm{AS}_\pi(\Gamma^\star) = \sum_{q\in Q^\star} \mu_q S(q)$$

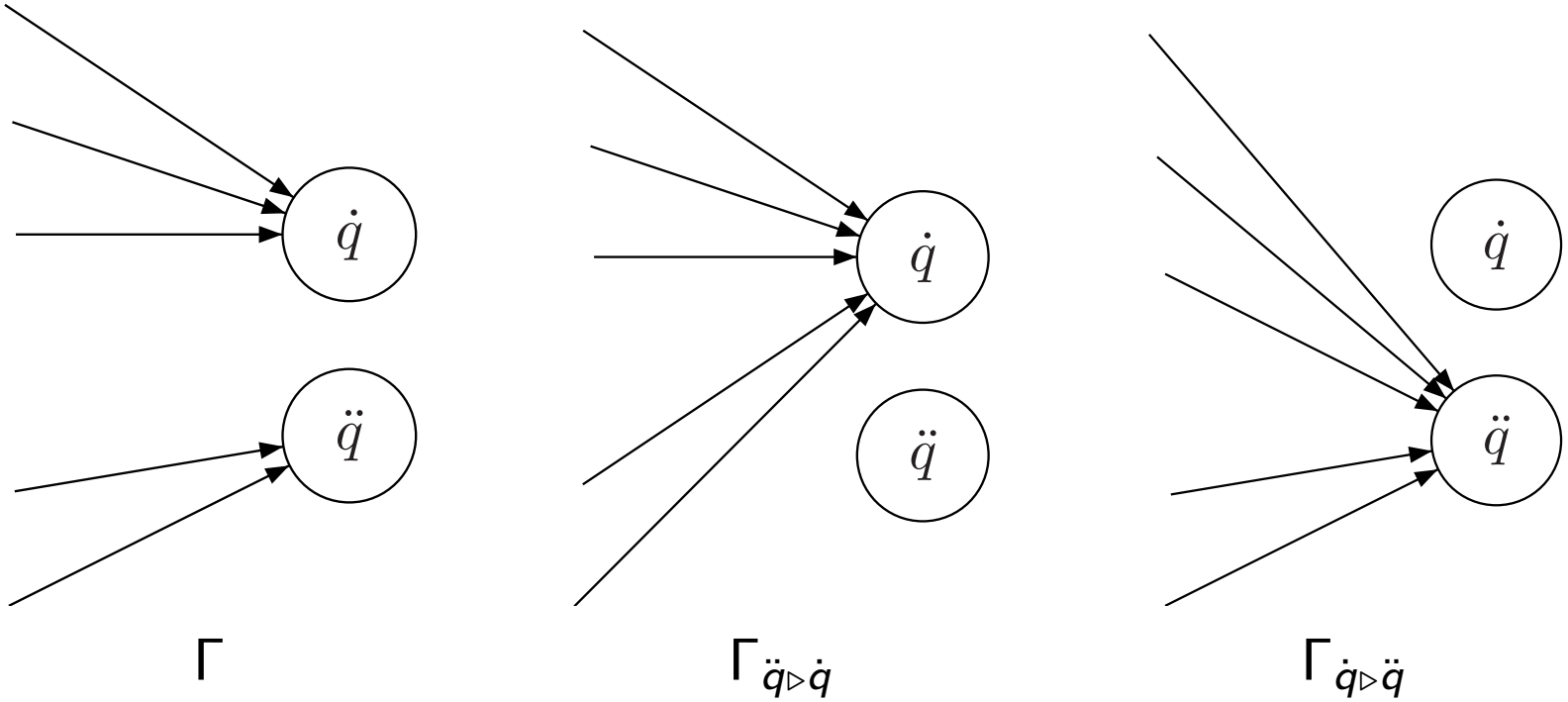
(μ_q) : limit frequencies of the states of $\Gamma^\star$ under $\mathcal{M}_{\Gamma^\star}$
$S(q)$: expected shift from q

Optimal w-matching machines

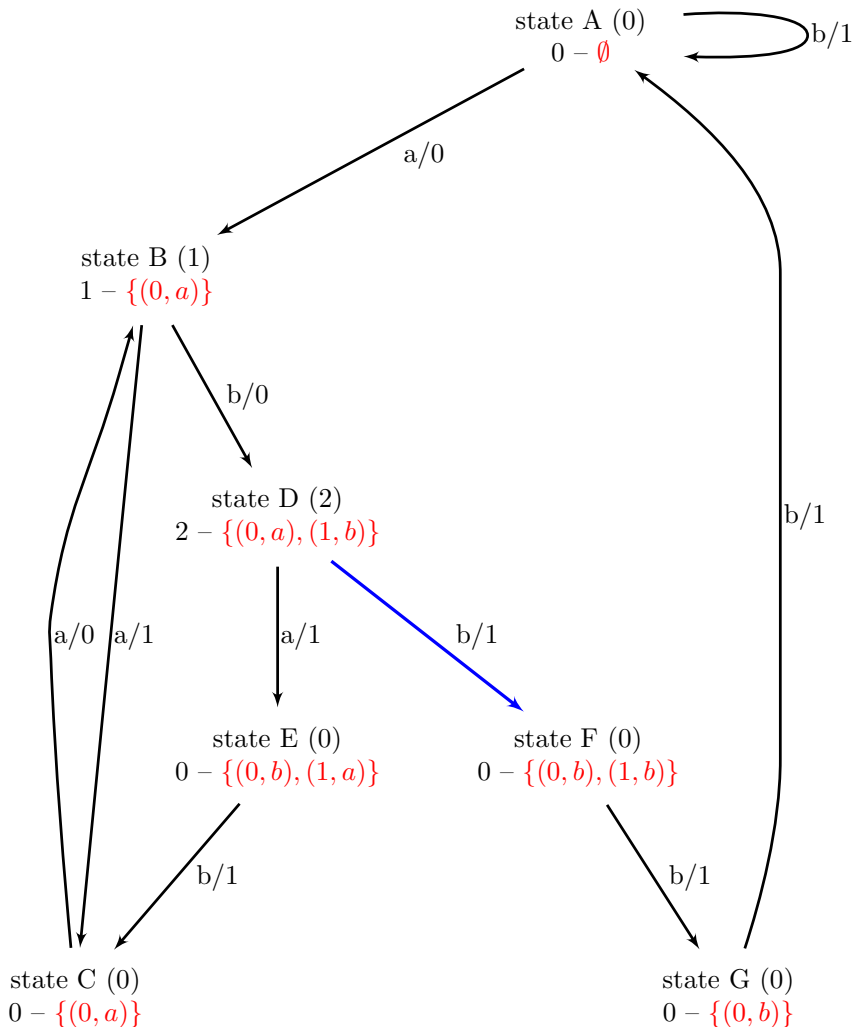
Question : being given a pattern w, an iid model π and an order n, find a matching machine Γ such that $\mathrm{AS}_\pi(\Gamma)$ is the greatest among the w-matching machines of order n.

Strategy : prove that for a given pattern w, an iid model π and an order n, there exists an optimal w-matching machine of order n which is in a finite (and enumerable) subset of w-matching machines.

Redirecting transitions



Removing states



Γ standard (i.e. $\Gamma = \Gamma^\star$)

$\mathrm{mem}_\Gamma(A) = \emptyset,$
$\mathrm{mem}_\Gamma(B) = \{(0, a)\},$
$\mathrm{mem}_\Gamma(C) = \{(0, a)\},$
$\mathrm{mem}_\Gamma(D) = \{(0, a), (1, b)\},$
$\ldots$

Theorem

If Γ is standard and valid and $\mathrm{mem}_\Gamma(\dot{q}) = \mathrm{mem}_\Gamma(\ddot{q})$ then both $\Gamma_{\ddot{q} \triangleright \dot{q}}$ and $\Gamma_{\dot{q} \triangleright \ddot{q}}$ are standard and valid.

Removing states and optimizing

Theorem

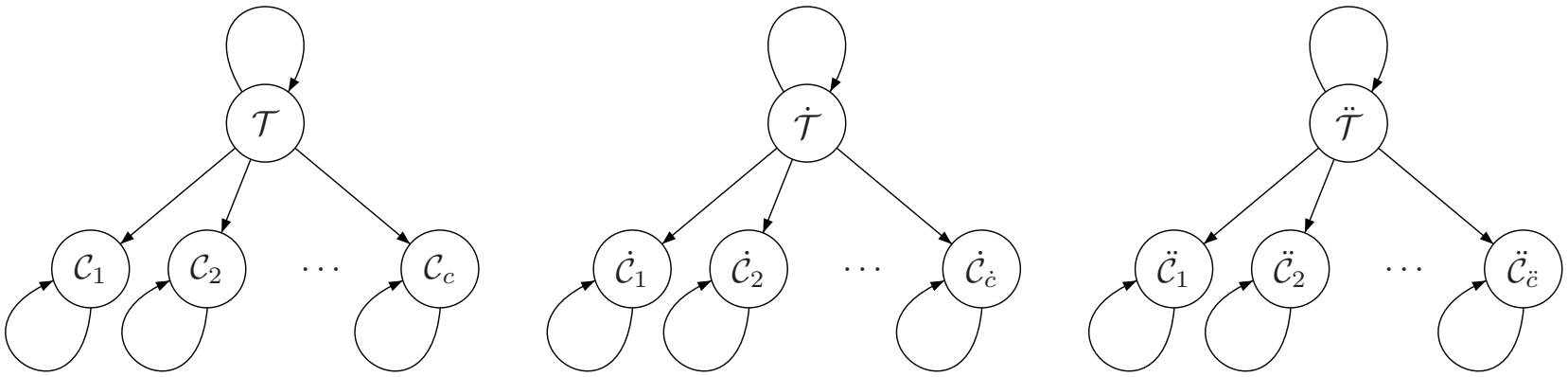
If Γ *is standard and contains two states* $\dot{q}$ *and* $\ddot{q}$ *such that* $\mathrm{mem}_\Gamma(\dot{q}) = \mathrm{mem}_\Gamma(\ddot{q})$ *then*

$$\mathrm{AS}_\pi(\Gamma) \leq \max\{\mathrm{AS}_\pi(\Gamma_{\ddot{q}\triangleright\dot{q}}), \mathrm{AS}_\pi(\Gamma_{\dot{q}\triangleright\ddot{q}})\}$$

Corollary

Being given a pattern w*, an iid model* π *and an order* n*, there exists an optimal* w*-matching machine* Γ *of order* n *such that* Q *is in bijection with a subset of partial functions from* $\{0, 1, \ldots, n\}$ *to* $\mathcal{A}$*.*

Sketch of the proof



$\Gamma + \pi \to M$ $\qquad$ $\Gamma_{\ddot{q} \triangleright \dot{q}} + \pi \to \dot{M}$ $\qquad$ $\Gamma_{\dot{q} \triangleright \ddot{q}} + \pi \to \ddot{M}$

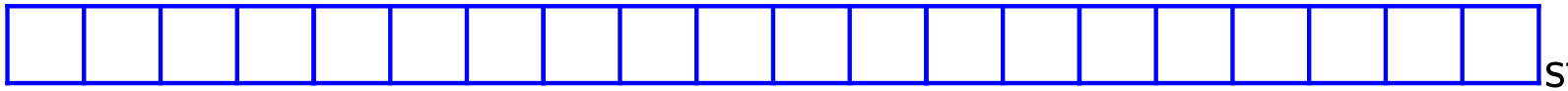
$\mathcal{T}$ transient class, $\mathcal{C}_1$, $\mathcal{C}_2$, $\ldots$, $\mathcal{C}_c$ recurrent classes

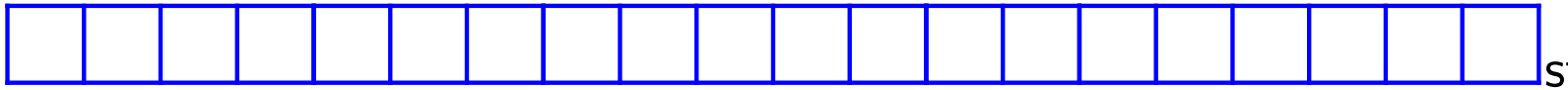
$$\mathrm{AS}_\pi(\Gamma) = \sum_{m=1}^{c} f(o, \mathcal{C}_m) \mathrm{AS}_\pi^{(m)}(\Gamma)$$

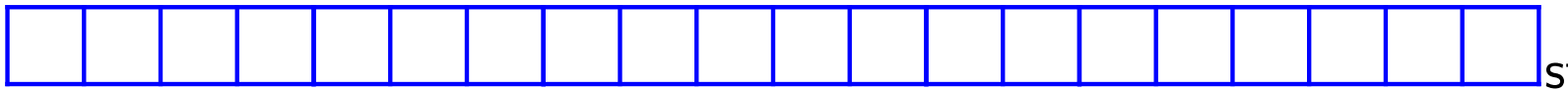
$\mathrm{AS}_\pi^{(m)}(\Gamma)$: asymptotic speed of the recurrent class $\mathcal{C}_m$
$f(o, \mathcal{C}_m)$: probability to end up in the class $\mathcal{C}_m$ by starting at o

Several cases depending on to which classes $\dot{q}$ and $\ddot{q}$ belong...

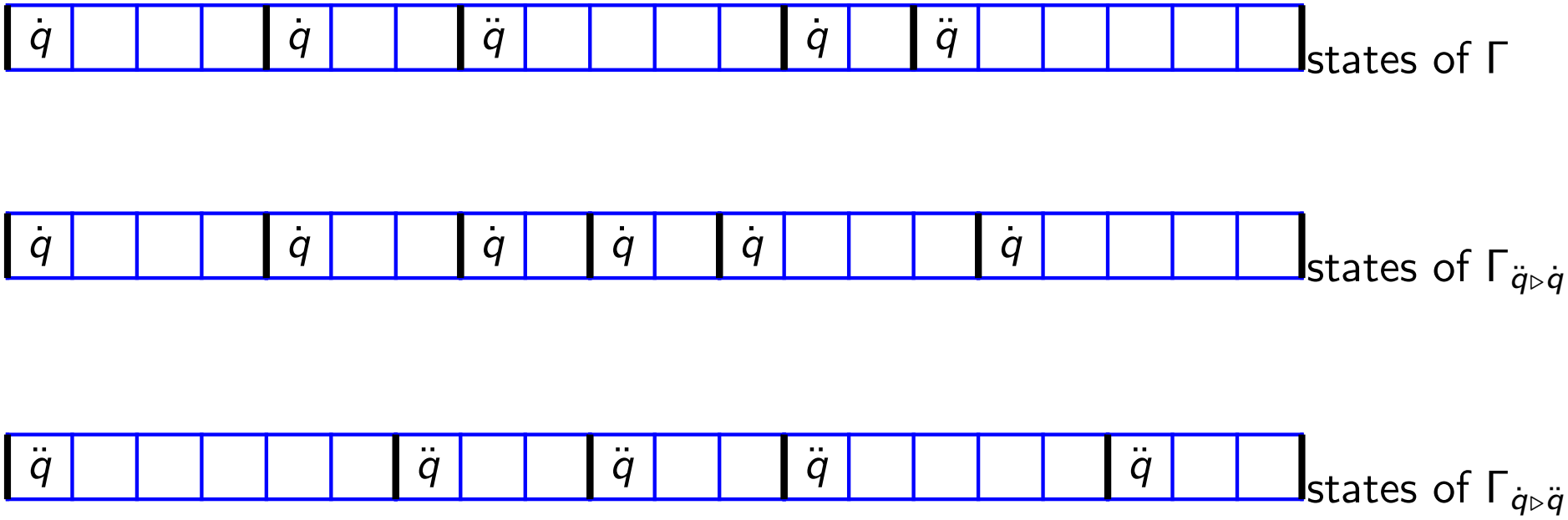
$\dot{q}$ and $\ddot{q}$ belong to a same recurrent class $\mathcal{C}_k$


states of Γ

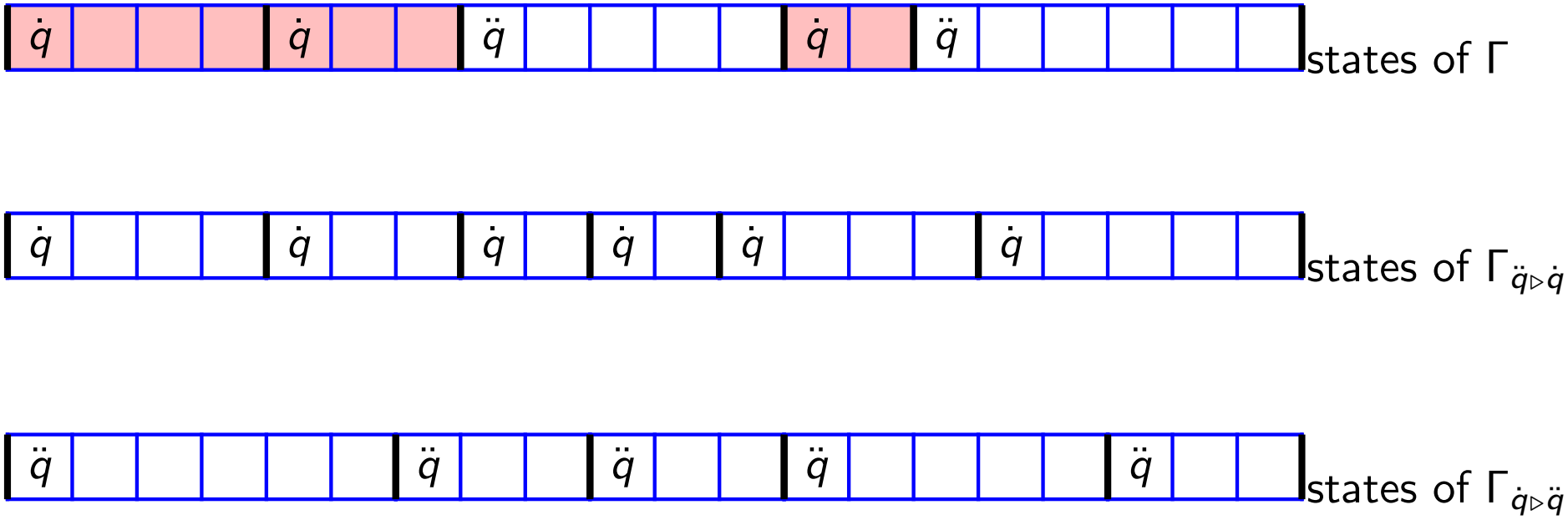

states of $\Gamma_{\ddot{q} \triangleright \dot{q}}$


states of $\Gamma_{\dot{q} \triangleright \ddot{q}}$

$\dot{q}$ and $\ddot{q}$ belong to a same recurrent class $\mathcal{C}_k$

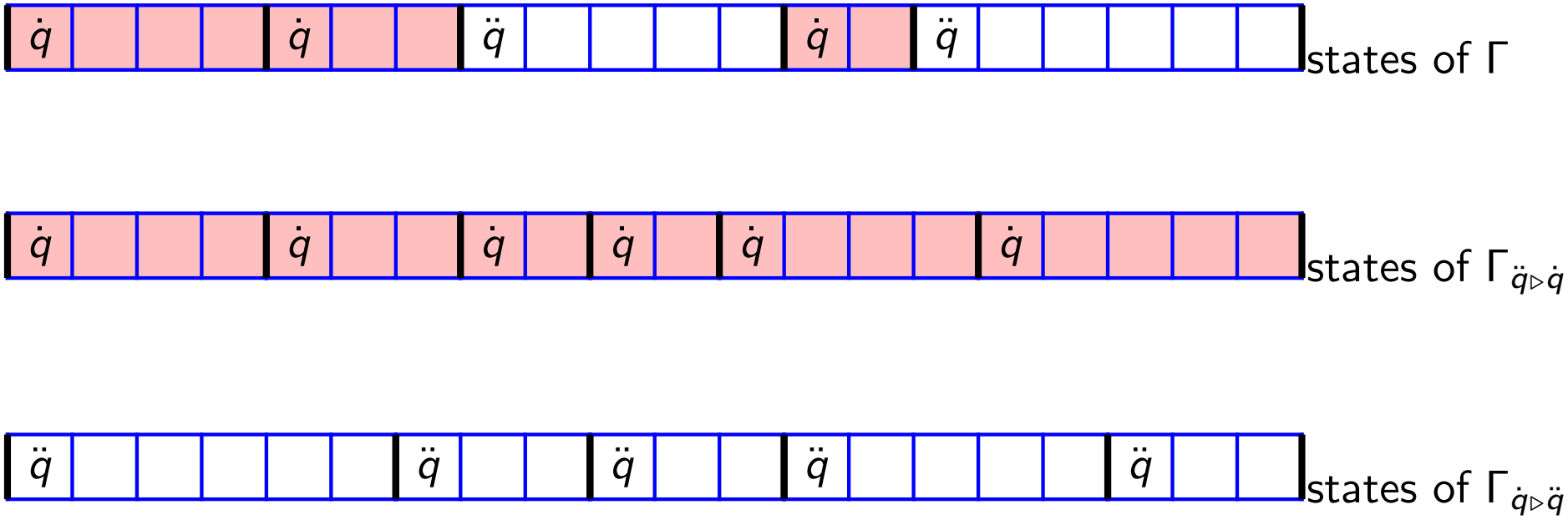


$\dot{q}$ and $\ddot{q}$ belong to a same recurrent class $\mathcal{C}_k$



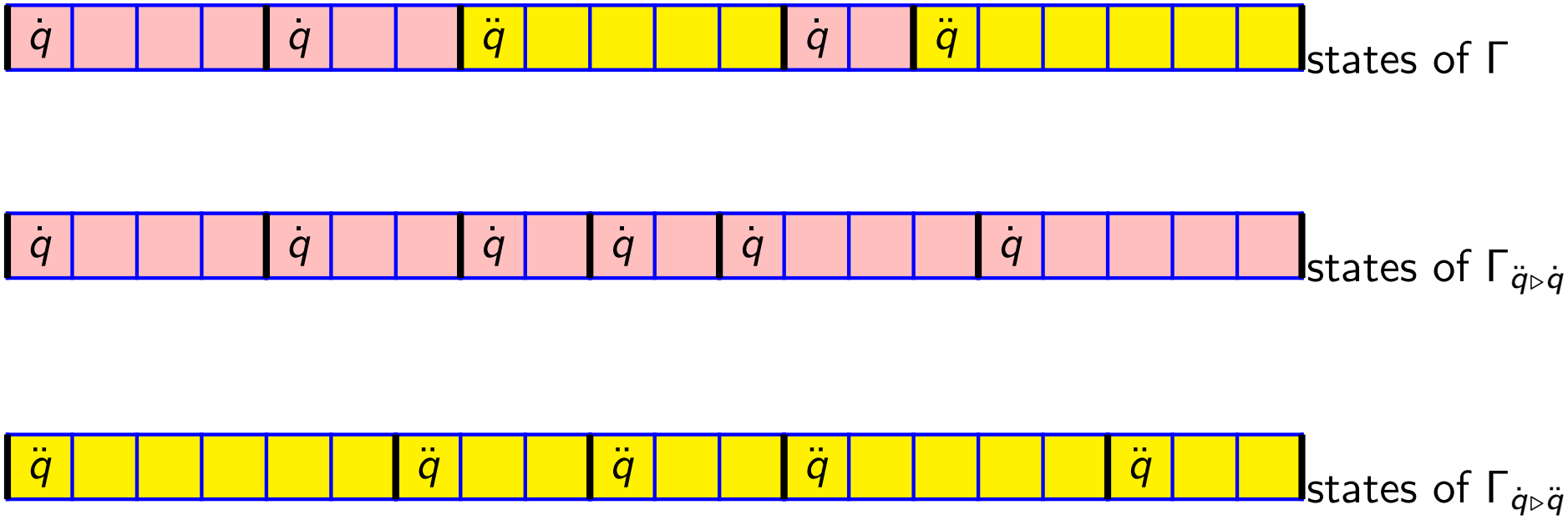
The lengths and the sums of shifts of the pink parts are independent and identically distributed

$\dot{q}$ and $\ddot{q}$ belong to a same recurrent class $\mathcal{C}_k$



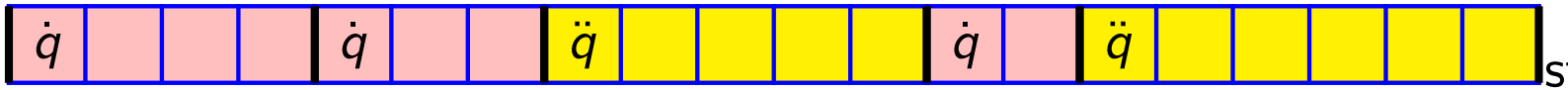
The lengths and the sums of shifts of the pink parts are independent and identically distributed

$\dot{q}$ and $\ddot{q}$ belong to a same recurrent class $\mathcal{C}_k$

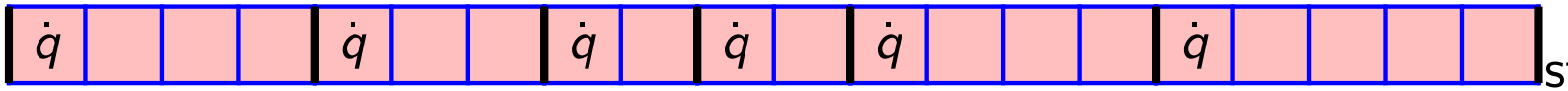


The lengths and the sums of shifts of the yellow parts are independent and identically distributed

$\dot{q}$ and $\ddot{q}$ belong to a same recurrent class $\mathcal{C}_k$



states of Γ



states of $\Gamma_{\ddot{q}\triangleright\dot{q}}$

states of $\Gamma_{\dot{q}\triangleright\ddot{q}}$

$$\mathrm{AS}_{\pi}^{(k)}(\Gamma) = \frac{\mu_{\dot{q}}}{\mu_{\dot{q}} + \mu_{\ddot{q}}} \mathrm{AS}_{\pi}^{(k)}(\Gamma_{\ddot{q}\triangleright\dot{q}}) + \frac{\mu_{\ddot{q}}}{\mu_{\dot{q}} + \mu_{\ddot{q}}} \mathrm{AS}_{\pi}^{(k)}(\Gamma_{\dot{q}\triangleright\ddot{q}})$$

$$\Downarrow$$

$$\mathrm{AS}_{\pi}^{(k)}(\Gamma) \leq \max\{\mathrm{AS}_{\pi}^{(k)}(\Gamma_{\ddot{q}\triangleright\dot{q}}), \mathrm{AS}_{\pi}^{(k)}(\Gamma_{\dot{q}\triangleright\ddot{q}})\}$$

An additional result...

Staying at the current position after a mismatch cannot improve the asymptotic speed...

Theorem

Being given a pattern w, an iid model π and an order n, there exists an optimal w-matching machine Γ such that Q is in bijection with a subset of partial functions f from $\{0, \ldots, n\}$ to $\mathcal{A}$, which are such that if $f(i)$ is defined and $i < |w|$ then $f(i) = w_i$.

Finding the best w-matching machine

For w-matching machine of order $|w| - 1$, there is a brute force algorithm with complexity

$$O\left(\prod_{k=1}^{|w|-1} k^{\binom{|w|}{k}} 2^{3|w|}\right)$$

ℓ-*Heuristic* based on the greatest speed expectation for the ℓ first reads

Theoretical evaluation

Asymptotic speeds for $\pi_a = 0.5$, $\pi_b = 0.5$

	Naive	Morris-Pratt	Knuth-Morris-Pratt	Quicksearch	Horspool	FJS	TVSBS	EBOM	Hashq	1-Heuristic	2-Heuristic	3-Heuristic	Fastest
aaaa	0.53	0.70	1.00	0.98	1.18	0.78	0.72	0.93	0.52	1.50	1.69	1.80	**1.83**
aaab	0.53	0.76	0.94	0.51	1.18	0.69	0.39	0.73	0.52	1.37	1.52	**1.60**	**1.60**
aaba	0.53	0.76	0.89	0.51	0.73	0.59	0.54	0.62	0.53	1.19	1.33	1.35	**1.37**
aabb	0.53	0.76	0.84	0.69	0.73	0.71	0.50	0.69	0.53	1.30	1.43	1.54	**1.56**
abaa	0.53	0.73	0.80	0.63	0.73	0.66	0.61	0.62	0.54	1.23	1.34	**1.38**	**1.38**
abab	0.53	0.70	0.80	0.50	0.73	0.56	0.50	0.73	0.54	1.22	1.33	1.36	**1.43**
abba	0.53	0.70	0.73	0.55	0.94	0.56	0.39	0.67	0.53	1.27	1.31	**1.34**	**1.34**
abbb	0.53	0.70	0.70	0.75	0.94	0.76	0.62	0.73	0.53	1.47	1.59	1.64	**1.69**
baaa	0.53	0.70	0.70	0.75	0.94	0.76	0.62	0.73	0.53	1.47	1.59	1.64	**1.69**
baab	0.53	0.70	0.73	0.55	0.94	0.56	0.39	0.67	0.53	1.27	1.31	**1.34**	**1.34**
baba	0.53	0.70	0.80	0.50	0.73	0.56	0.50	0.73	0.54	1.22	1.33	1.36	**1.43**
babb	0.53	0.73	0.80	0.63	0.73	0.66	0.61	0.62	0.54	1.23	1.34	**1.38**	**1.38**
bbaa	0.53	0.76	0.84	0.69	0.73	0.71	0.50	0.69	0.53	1.30	1.43	1.54	**1.56**
bbab	0.53	0.76	0.89	0.51	0.73	0.59	0.54	0.62	0.53	1.19	1.33	1.35	**1.37**
bbba	0.53	0.76	0.94	0.51	1.18	0.69	0.39	0.73	0.52	1.37	1.52	**1.60**	**1.60**
bbbb	0.53	0.70	1.00	0.98	1.18	0.78	0.72	0.93	0.52	1.50	1.69	1.80	**1.83**

Theoretical evaluation

Asymptotic speeds for $\pi_a = 0.1$, $\pi_b = 0.9$

	Naive	Morris-Pratt	Knuth-Morris-Pratt	Quicksearch	Horspool	FJS	TVSBS	EBOM	Hashq	1-Heuristic	2-Heuristic	3-Heuristic	Fastest
aaaa	0.90	0.91	1.00	2.18	3.30	1.97	1.68	1.49	0.67	3.02	3.47	**3.50**	**3.50**
aaab	0.90	0.91	1.00	0.52	1.77	0.53	0.27	1.37	0.66	2.43	2.60	2.60	**2.61**
aaba	0.89	0.92	1.00	0.92	0.91	0.87	1.55	1.24	0.66	1.77	**2.19**	**2.19**	**2.19**
aabb	0.89	0.92	1.00	0.56	0.38	0.53	0.29	0.70	0.63	1.74	1.79	**1.80**	**1.80**
abaa	0.83	0.91	0.98	1.19	1.67	1.24	1.63	1.24	0.66	1.80	2.15	**2.18**	**2.18**
abab	0.83	0.92	0.99	0.50	0.85	0.54	0.28	1.17	0.63	1.42	1.80	1.80	**1.81**
abba	0.79	0.92	0.98	0.92	0.93	0.87	0.82	0.63	0.61	1.30	1.73	**1.80**	**1.80**
abbb	0.79	0.97	0.97	0.56	0.33	0.57	0.31	0.31	0.46	1.08	1.10	1.14	**1.15**
baaa	0.50	0.53	0.53	1.19	2.50	1.49	1.33	1.37	0.66	2.44	2.60	**2.61**	**2.61**
baab	0.50	0.53	0.53	0.37	1.34	0.37	0.22	1.18	0.65	**1.75**	**1.75**	**1.75**	**1.75**
baba	0.48	0.53	0.55	0.58	0.91	0.80	1.16	1.17	0.63	1.09	1.78	**1.84**	**1.84**
babb	0.48	0.57	0.57	0.39	0.38	0.39	0.21	0.55	0.54	1.04	1.04	1.04	**1.05**
bbaa	0.36	0.55	0.55	0.75	1.67	1.10	1.06	0.70	0.63	1.09	1.72	**1.84**	**1.84**
bbab	0.36	0.57	0.58	0.29	0.85	0.31	0.23	0.55	0.54	1.01	**1.08**	**1.08**	**1.08**
bbba	0.29	0.59	0.60	0.42	1.00	0.90	0.73	0.31	0.36	1.08	1.16	**1.24**	**1.24**
bbbb	0.29	0.80	1.00	0.29	0.35	0.27	0.24	0.27	0.19	1.03	1.03	**1.05**	**1.05**

Practical evaluation

Average speeds on E. Coli genome and the Bible

	Naive	Morris-Pratt	Knuth-Morris-Pratt	Quicksearch	Horspool	FJS	TVSBS	EBOM	Hashq	1-Heuristic	2-Heuristic	3-Heuristic	Fastest
gcga	0.73	0.80	0.84	1.11	1.96	1.03	1.08	1.21	0.62	1.90	2.24	**2.25**	2.23
cgcg	0.74	0.80	0.85	1.27	1.88	1.13	1.16	1.23	0.62	1.96	2.34	2.32	**2.35**
ctta	0.76	0.80	0.80	1.16	2.20	1.10	1.08	1.27	0.63	2.05	2.35	**2.36**	**2.36**
ttca	0.75	0.82	0.94	1.15	1.96	1.03	1.10	1.22	0.64	1.93	2.21	2.21	**2.22**
tacc	0.77	0.81	0.81	1.31	1.78	1.16	1.19	1.28	0.63	2.06	2.36	**2.37**	2.36
ccct	0.76	0.80	0.99	1.34	2.42	1.22	1.14	1.34	0.63	2.05	2.45	**2.53**	2.52
ttgt	0.75	0.82	0.95	1.32	2.06	1.17	1.18	1.22	0.65	2.01	2.36	2.38	**2.39**
gtgg	0.76	0.80	0.83	1.44	2.05	1.23	1.23	1.24	0.64	2.06	2.41	2.42	**2.43**
le t	0.97	0.97	0.97	1.98	2.84	1.78	1.70	1.42	0.66	2.89	**3.07**	**3.07**	**3.07**
hem	0.91	0.94	0.94	1.78	3.04	1.59	1.48	1.45	0.66	2.89	3.05	**3.06**	**3.06**
are	0.83	0.84	0.84	1.69	3.13	1.76	1.60	1.45	0.67	3.01	**3.16**	**3.16**	**3.16**
at d	0.93	0.94	0.94	1.93	3.15	2.02	1.85	1.46	0.66	2.96	**3.22**	**3.22**	**3.22**
f th	0.97	0.98	0.98	2.03	2.72	1.81	1.69	1.35	0.65	2.94	**3.10**	**3.10**	**3.10**
r th	0.95	0.96	0.96	1.99	2.73	1.80	1.69	1.36	0.66	2.94	**3.12**	**3.12**	**3.12**
fede	0.98	0.98	0.98	2.21	3.35	1.97	1.73	1.49	0.66	3.23	**3.53**	**3.53**	**3.53**
ich	0.96	0.96	0.96	1.96	3.11	1.59	1.52	1.47	0.66	3.06	**3.15**	**3.15**	**3.15**

Practical evaluation

Average speeds on E. Coli genome and the Bible

	Naive	Morris-Pratt	Knuth-Morris-Pratt	Quicksearch	Horspool	FJS	TVSBS	EBOM	Hashq	1-Heuristic	2-Heuristic	3-Heuris
ccctgatcgatagcggaacgtccggtatca	0.75	0.80	0.99	1.30	1.93	1.14	2.82	7.36	5.19	2.95	7.12	**10.06**
accacaccgccttccccctggatgggttcg	0.76	0.80	0.81	1.57	2.67	1.31	3.19	7.76	4.74	2.98	7.25	**10.76**
cggccatgttattttcggcgttggtatgat	0.74	0.80	0.81	1.92	2.92	1.54	3.65	7.87	4.48	2.98	7.13	**10.55**
gtagttggagccttgttcagaggcggcgca	0.76	0.80	0.80	2.10	4.25	1.58	2.50	7.86	5.00	2.99	7.13	**10.43**
gagccagagagttcagtgtacgggaattcg	0.76	0.80	0.83	1.15	2.30	1.01	2.52	7.37	4.47	2.99	7.19	**10.28**
caggcgcgatatcgtttaatgcggcggcga	0.74	0.80	0.80	1.82	4.08	1.44	3.01	7.72	5.06	2.93	7.17	**10.26**
cctttgacggcataatccagaaagcaaaag	0.75	0.80	0.94	2.57	4.57	2.03	3.59	7.23	5.30	3.09	7.24	**10.70**
gaatcaaggaagcaagccgccagacgtccc	0.75	0.80	0.80	1.83	2.58	1.41	3.68	7.96	5.58	2.97	7.23	**11.02**
tggcgcggcgagaggttggtcgcatattgg	0.74	0.80	0.80	1.86	2.61	1.43	3.29	8.06	5.20	2.95	7.20	**10.62**
cactccccggttcccgcgccaatattgtcg	0.74	0.80	0.84	1.34	2.15	1.15	2.43	7.63	5.08	3.06	7.24	**10.47**
at the mount called the mount	0.83	0.84	0.84	5.95	11.92	5.87	7.98	11.69	8.56	8.03	17.62	**18.42**
ith Israel, to wit, with all t	0.95	0.96	0.96	7.55	13.34	7.19	8.38	12.65	8.73	9.06	17.97	**18.79**
esus going up to Jerusalem too	0.91	0.91	0.91	6.38	11.90	5.96	8.95	13.18	8.85	9.43	18.05	**18.70**
them, as they were able to hea	0.88	0.93	0.93	6.62	12.37	6.14	8.05	11.69	8.60	7.80	16.57	**17.96**
o in Osee, I will call them my	0.94	0.95	0.95	7.19	14.83	7.52	9.04	12.04	8.54	8.95	18.80	**19.34**
things are come upon thee, the	0.90	0.93	0.93	6.70	10.14	5.78	7.58	11.68	8.64	8.51	16.84	**18.05**
Syria, that dwelt at Damascus,	1.00	1.00	1.00	8.58	16.21	8.40	9.62	12.79	8.77	10.43	19.60	**20.08**
full of darkness. If therefor	0.84	0.84	0.84	6.56	12.80	6.76	8.57	11.94	8.53	9.39	18.74	**19.14**
e it: for there is no other sa	0.88	0.91	0.91	5.91	12.42	5.54	8.02	11.74	8.84	7.72	16.94	**17.86**
g, Syria is confederate with E	0.99	0.99	0.99	6.25	12.10	6.32	9.37	12.81	8.65	9.76	18.61	**18.95**

Références

Gilles Didier.
Optimal pattern matching algorithms.
submitted, https://arxiv.org/abs/1604.08437, 2016.

Gilles Didier and Laurent Tichit.
Designing optimal- and fast-on-average pattern matching algorithms.
Journal of Discrete Algorithms, 42 :45 – 60, 2017.

Position lattice of the pattern *abb*