

Système d'automates communicants

soutenance de stage M2 - PENSUNS - (Mars-Août 2012)

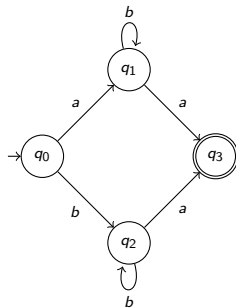
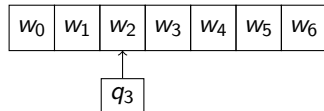
Bruno Guillon,
dirigé par Christian Choffrut

L.I.A.F.A, Université Paris VII, 2 pl. Jussieu, 75251 Paris, France

7 septembre 2012

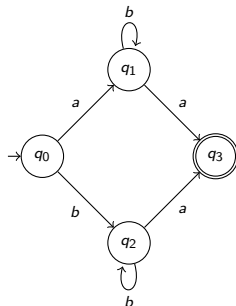
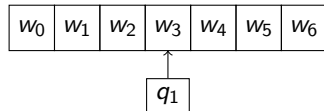
Encore des automates finis

On sait ce qu'est un automate fini



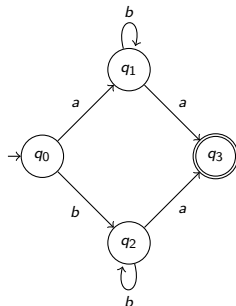
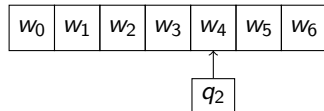
Encore des automates finis

On sait ce qu'est un automate fini



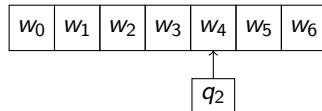
Encore des automates finis

On sait ce qu'est un automate fini

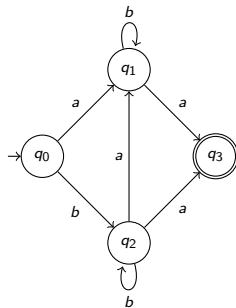


Encore des automates finis

On sait ce qu'est un automate fini

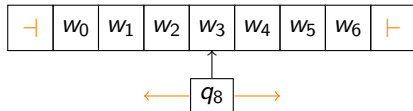


- ▶ déterministe/non-déterministe

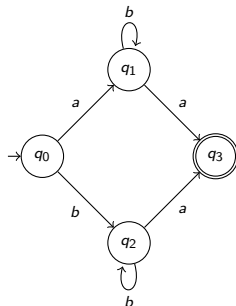


Encore des automates finis

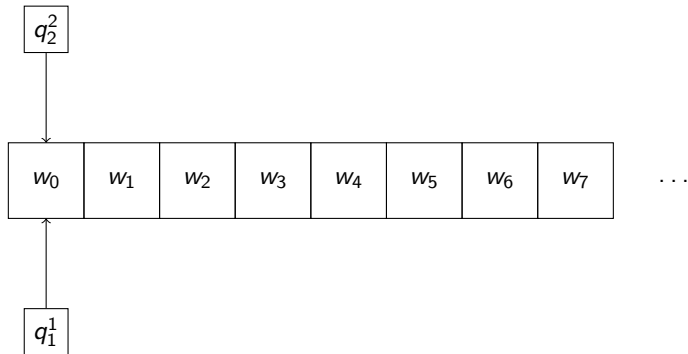
On sait ce qu'est un automate fini



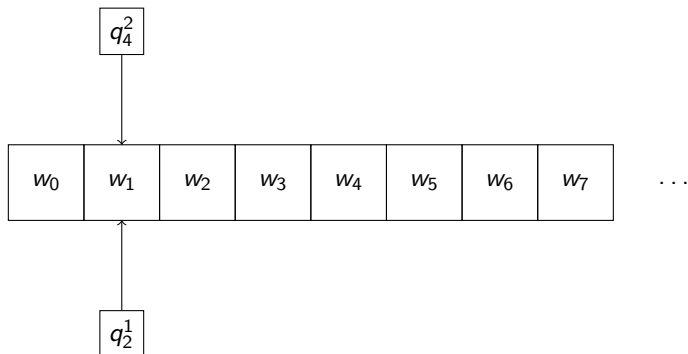
- ▶ déterministe/non-déterministe
- ▶ uni/bi-directionnel



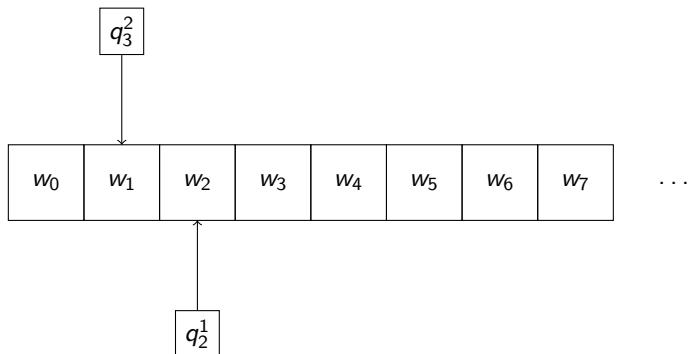
Plusieurs automates



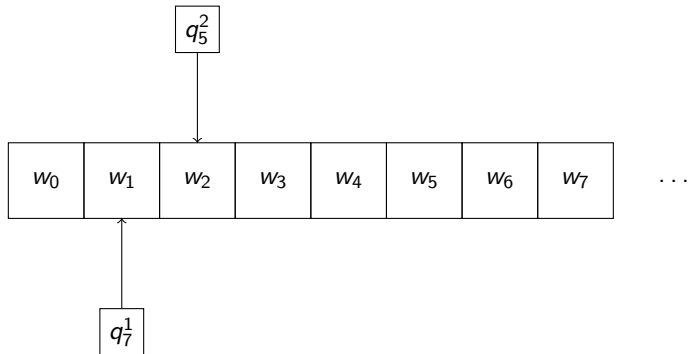
Plusieurs automates



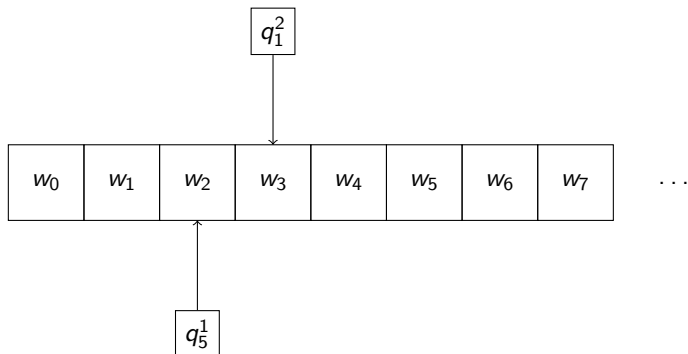
Plusieurs automates



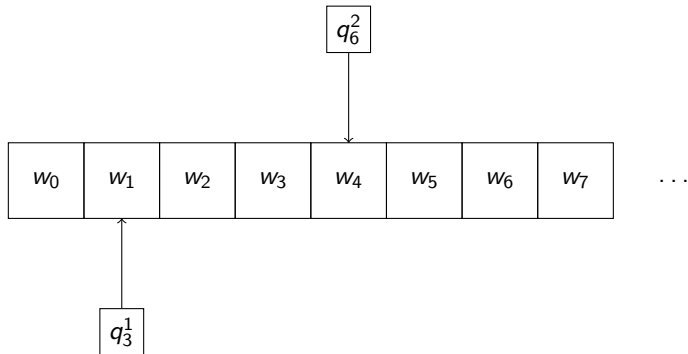
Plusieurs automates



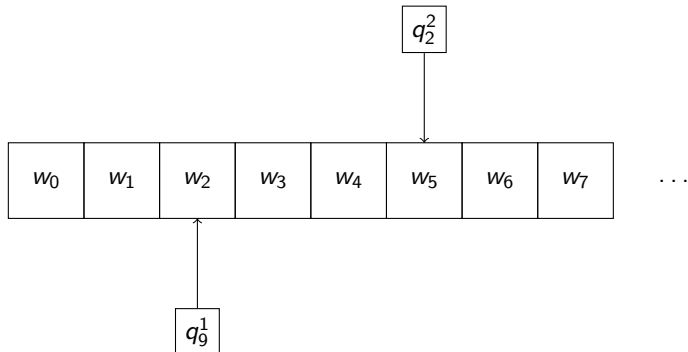
Plusieurs automates



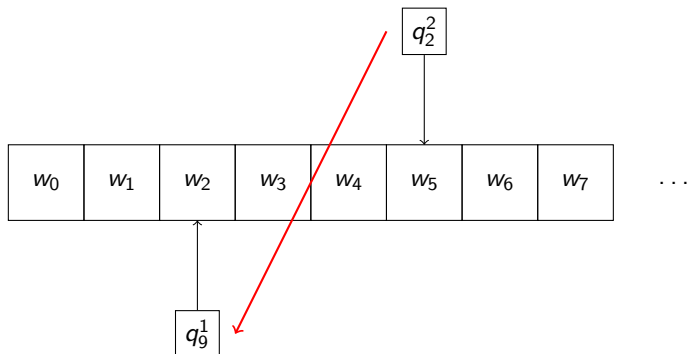
Plusieurs automates



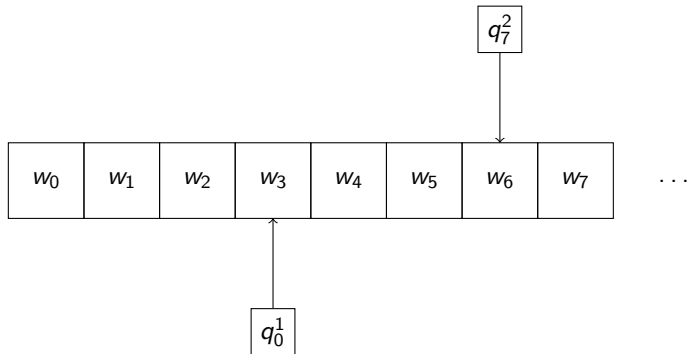
Plusieurs automates



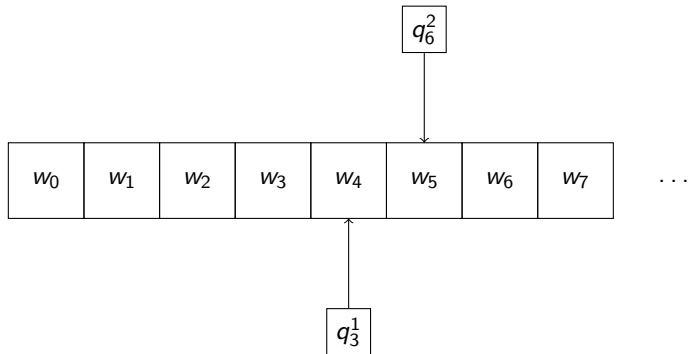
Plusieurs automates



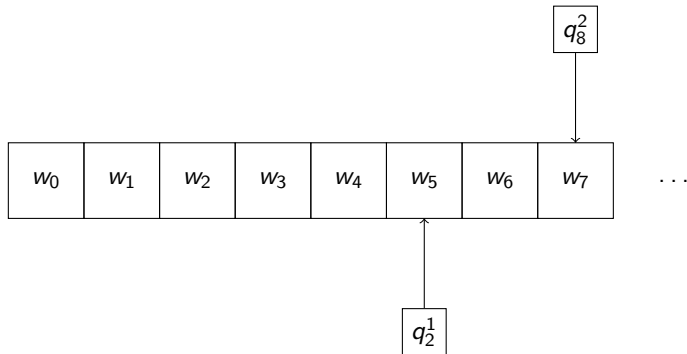
Plusieurs automates



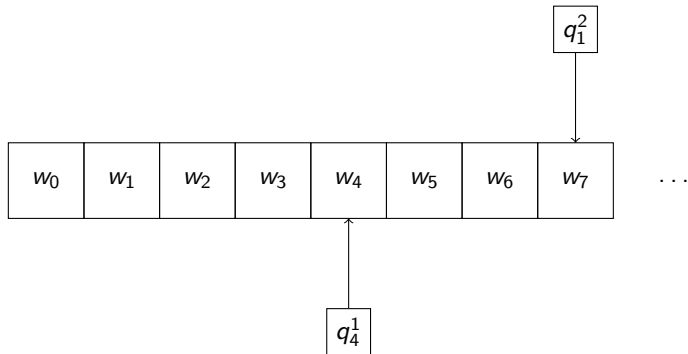
Plusieurs automates



Plusieurs automates



Plusieurs automates



Formellement

Chaque automate a un état. Un pas de calcul :

Formellement

Chaque automate a un état. Un pas de calcul :

- ▶ décision de l'envoi d'un message par une fonction binaire ν

Formellement

Chaque automate a un état. Un pas de calcul :

- ▶ décision de l'envoi d'un message par une fonction binaire ν
 - ▶ envoi (broadcast) éventuel du message (état, caractère lu)

Formellement

Chaque automate a un état. Un pas de calcul :

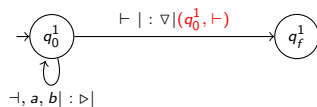
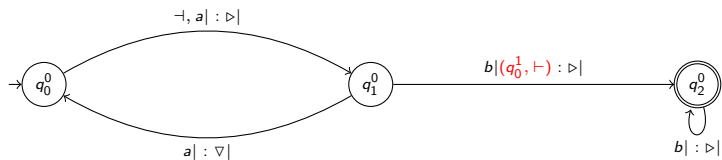
- ▶ décision de l'envoi d'un message par une fonction binaire ν
 - ▶ envoi (broadcast) éventuel du message (état, caractère lu)
- ▶ réception des messages

Formellement

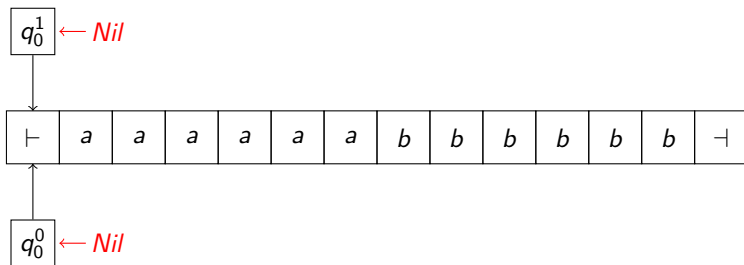
Chaque automate a un état. Un pas de calcul :

- ▶ décision de l'envoi d'un message par une fonction binaire ν
 - ▶ envoi (broadcast) éventuel du message (état, caractère lu)
- ▶ réception des messages
- ▶ transition selon une fonction δ
(état, caractère lu, message) \rightarrow (état, déplacement)

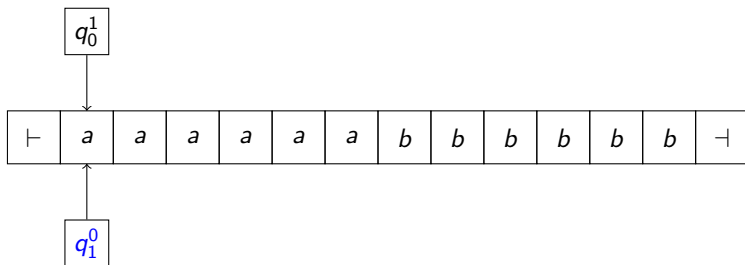
Exemple



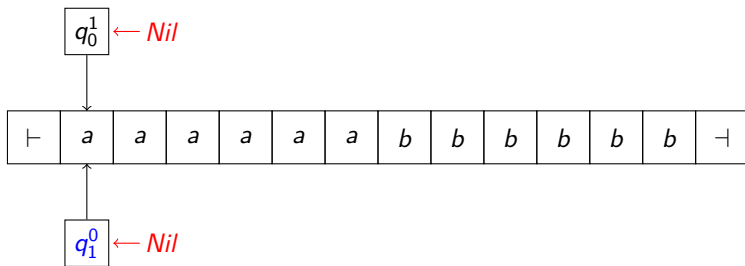
Example



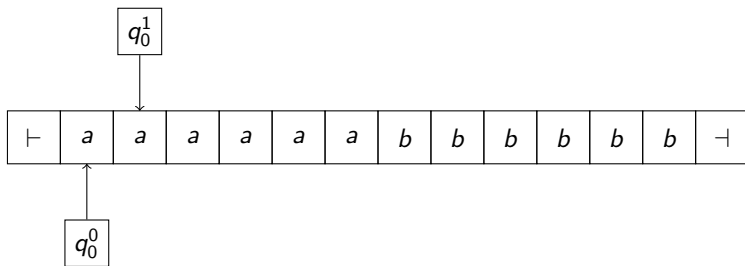
Example



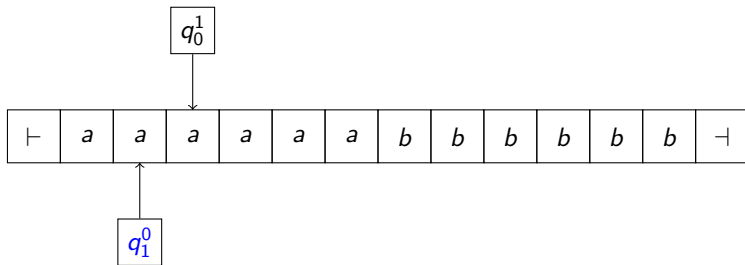
Example



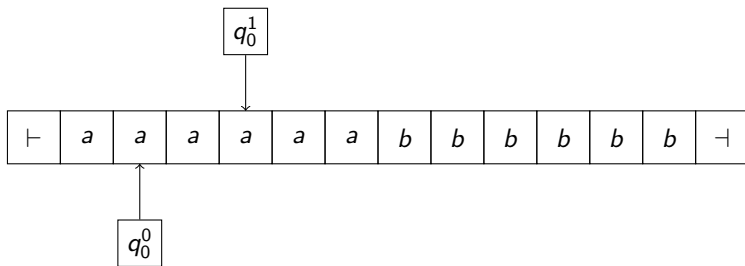
Exemple



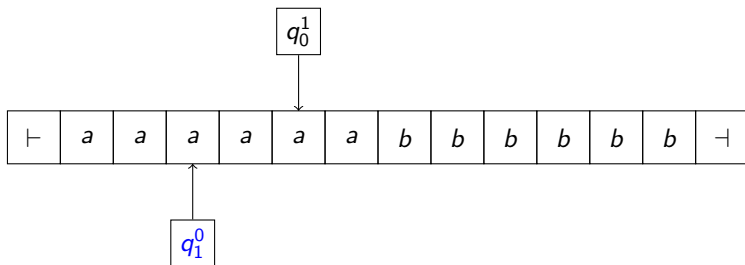
Exemple



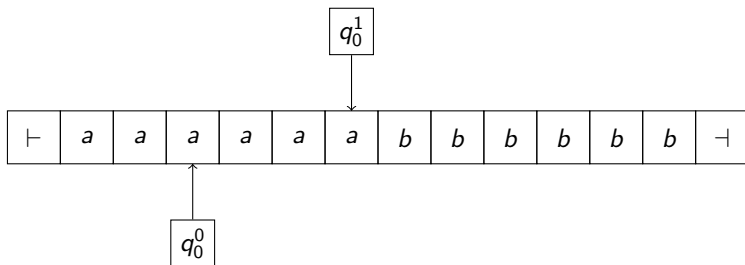
Exemple



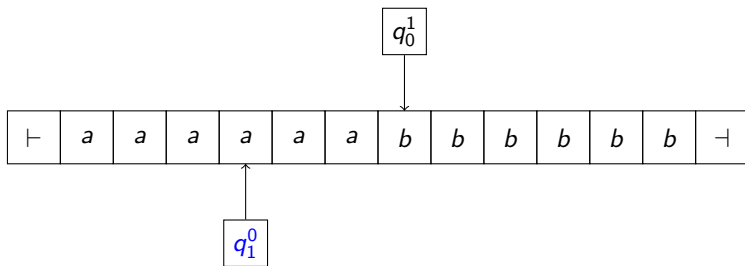
Exemple



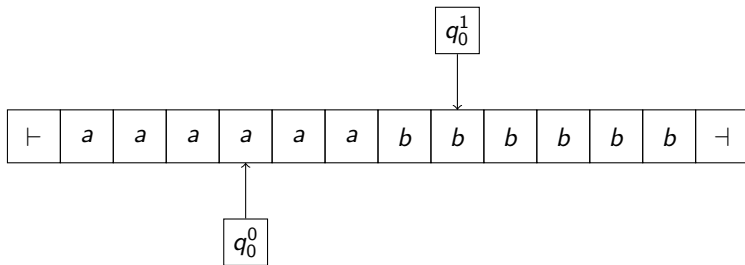
Exemple



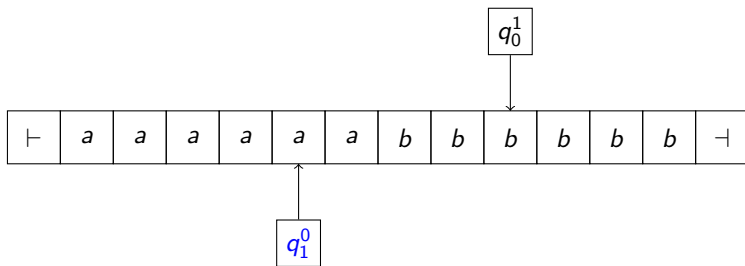
Exemple



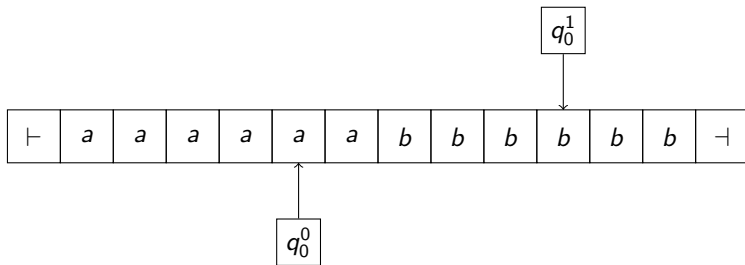
Exemple



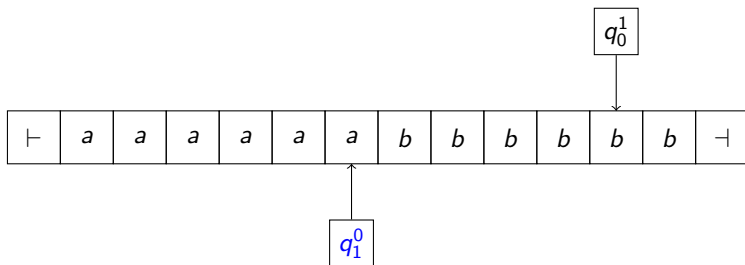
Exemple



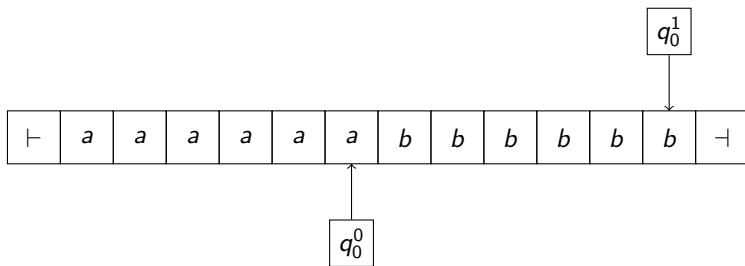
Exemple



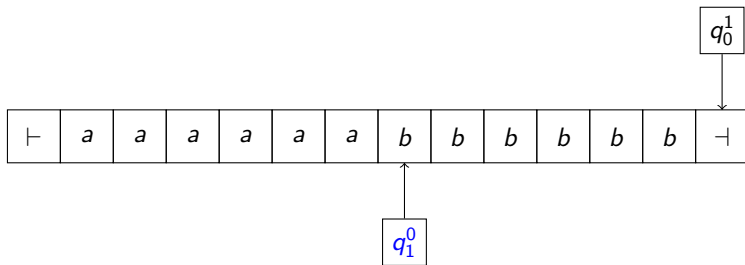
Exemple



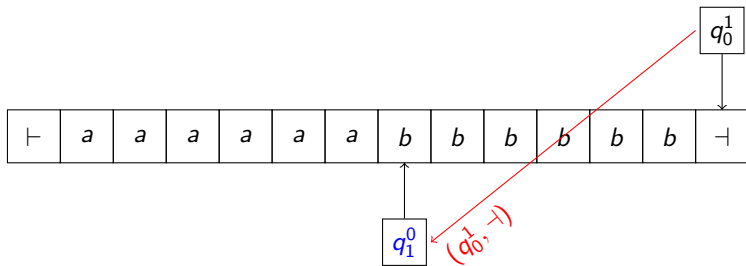
Example



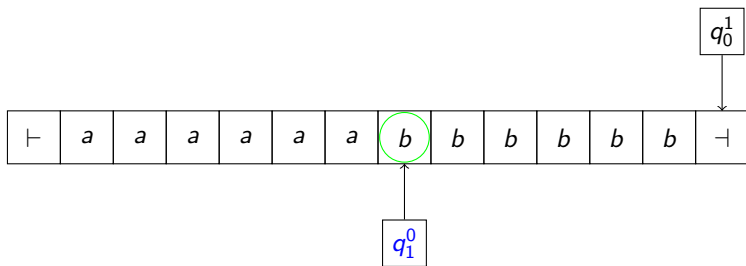
Example



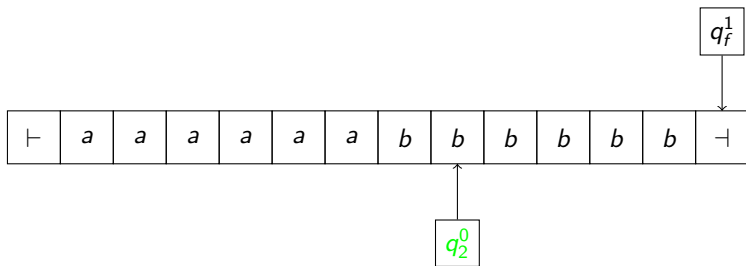
Example



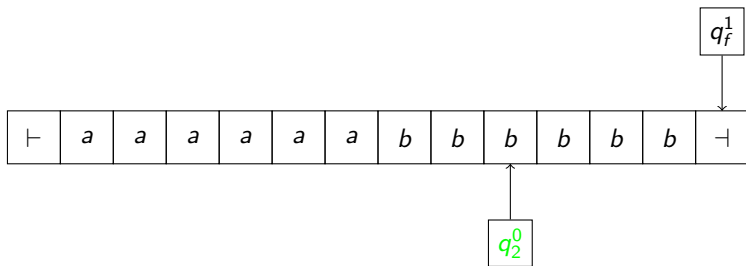
Example



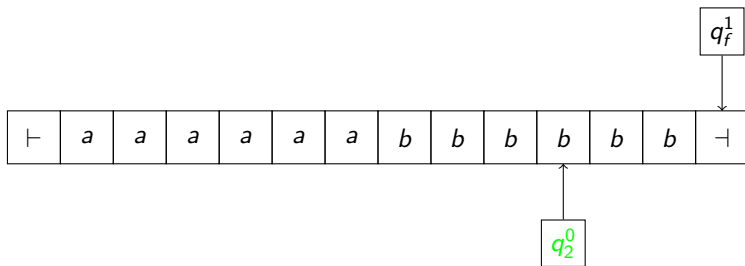
Example



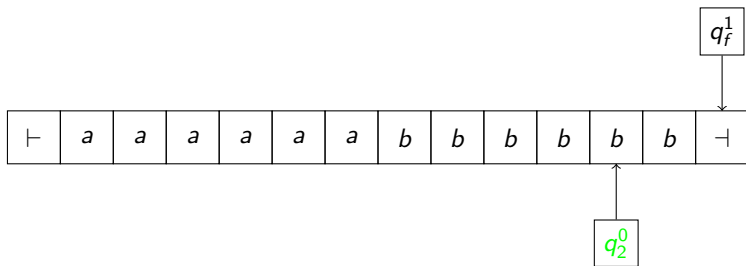
Exemple



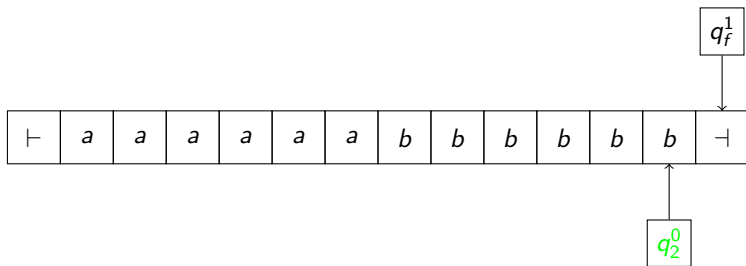
Example



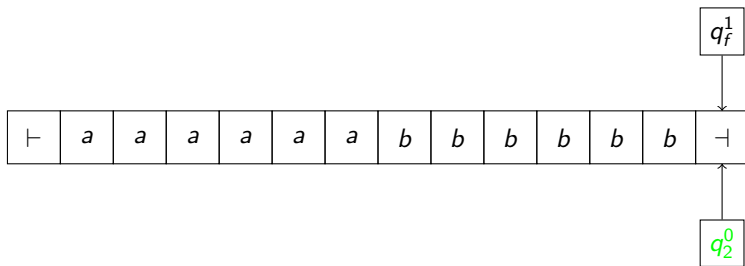
Example



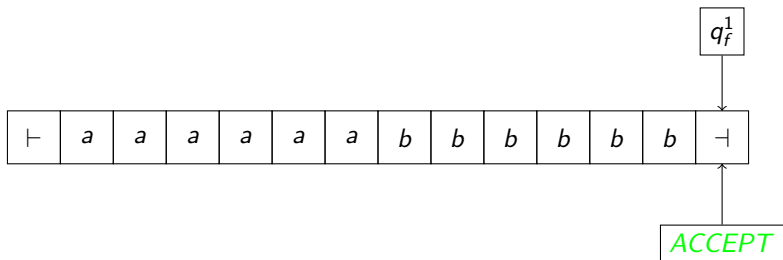
Example



Example

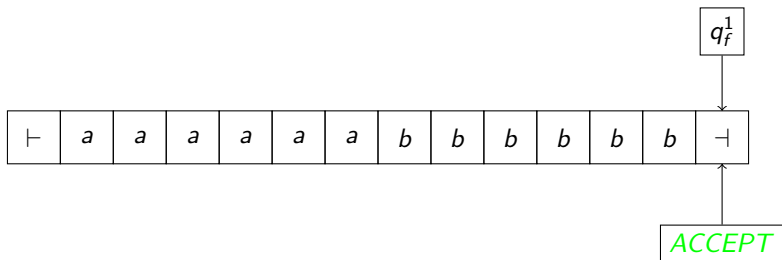


Example



Exemple

langage accepté : $\{a^n b^n\}$

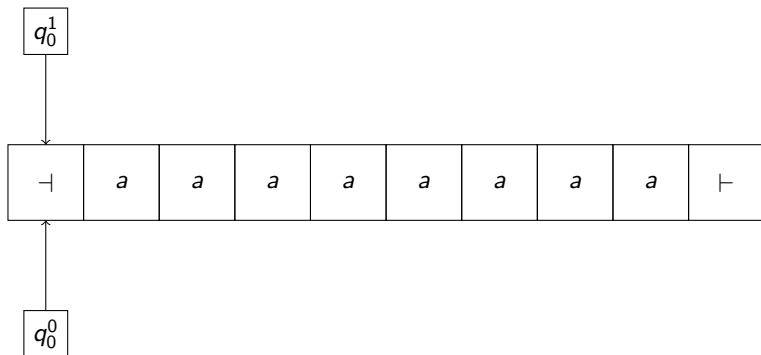


1 message

Et Bidirectionnels? Et sur un alphabet unaire?

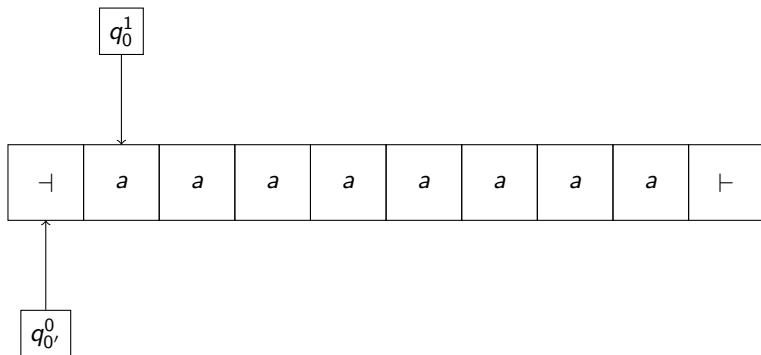
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



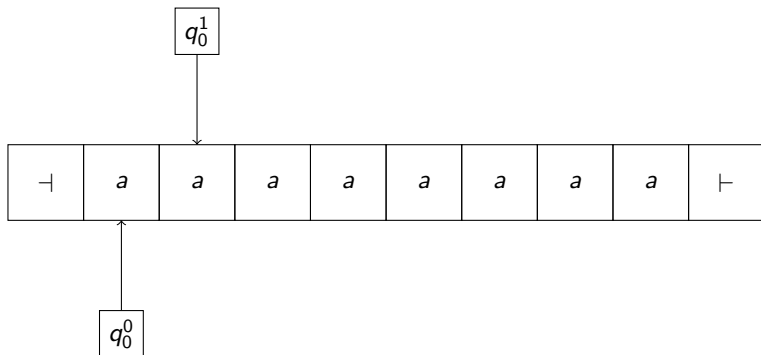
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



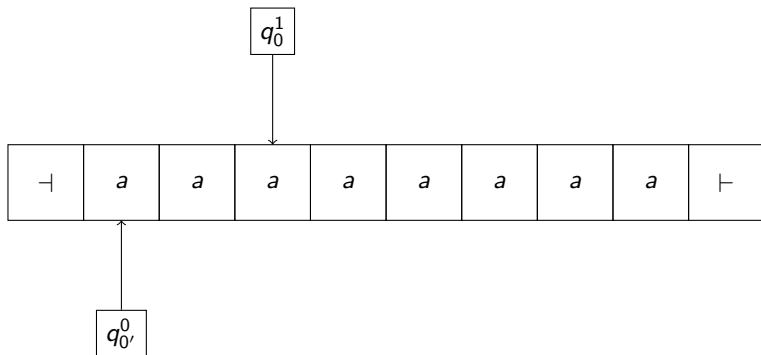
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



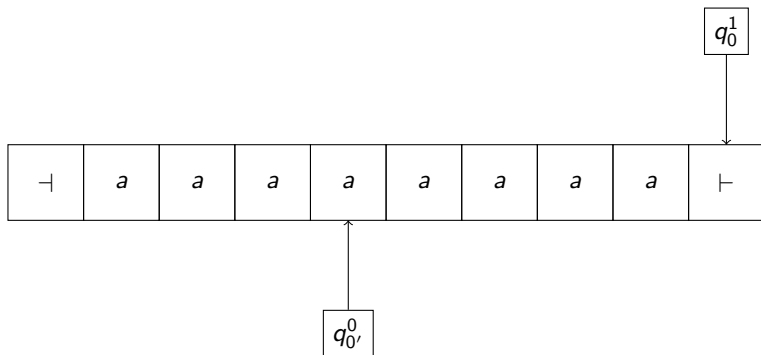
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



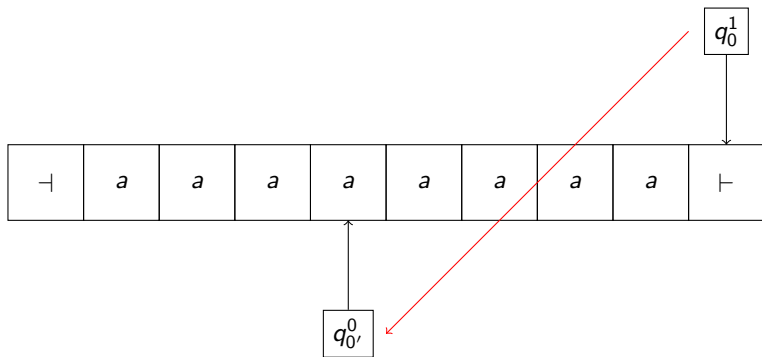
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



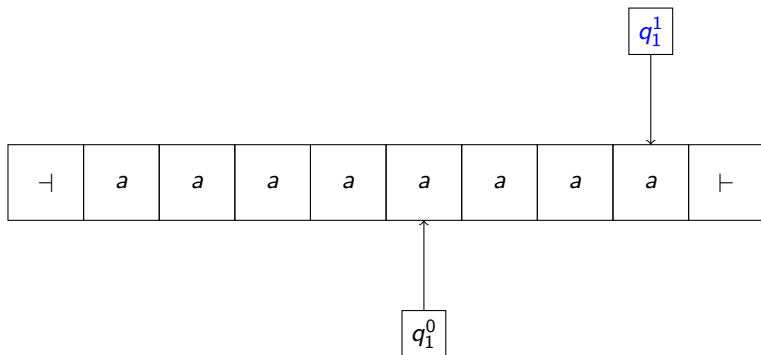
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



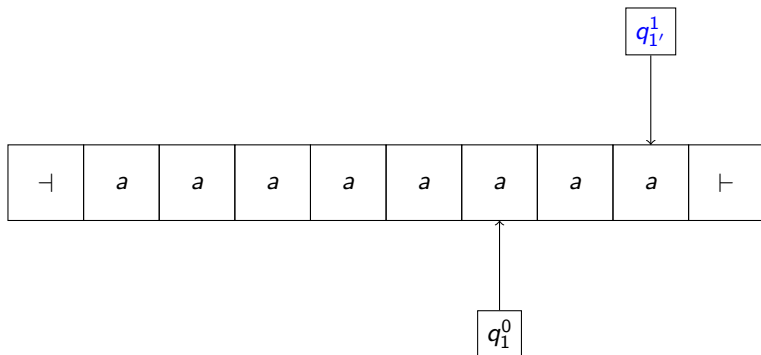
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



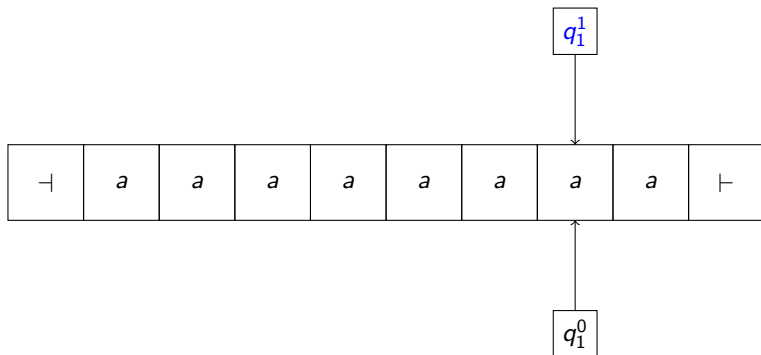
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



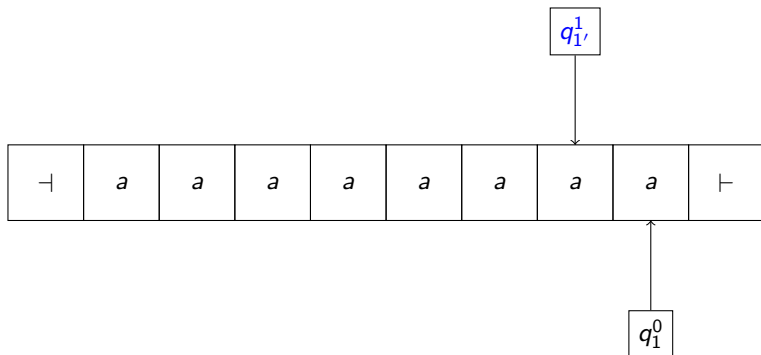
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



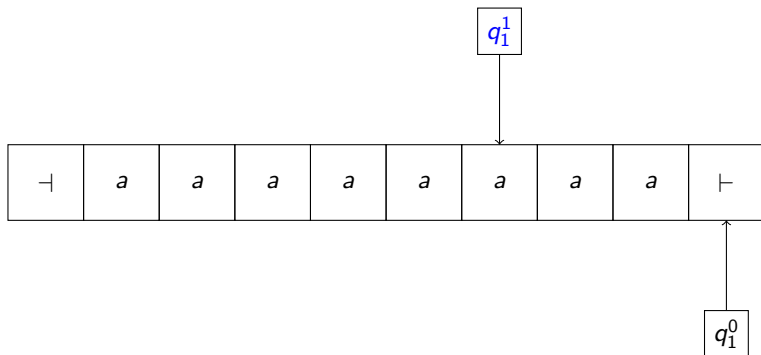
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



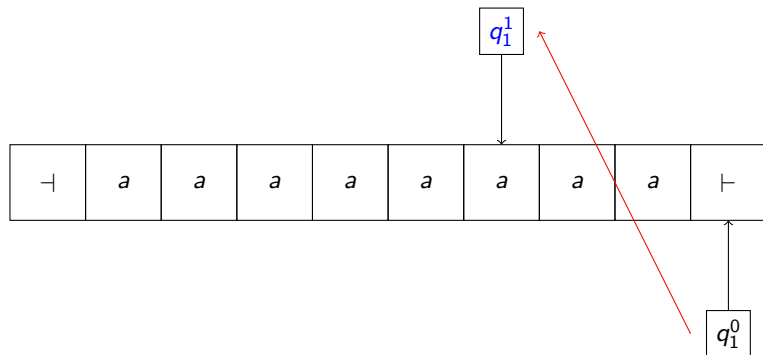
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



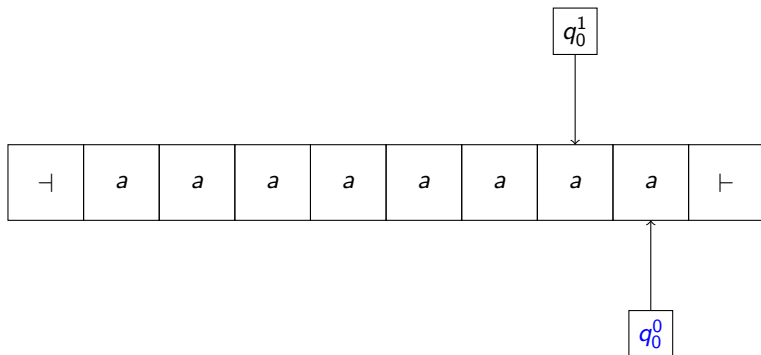
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



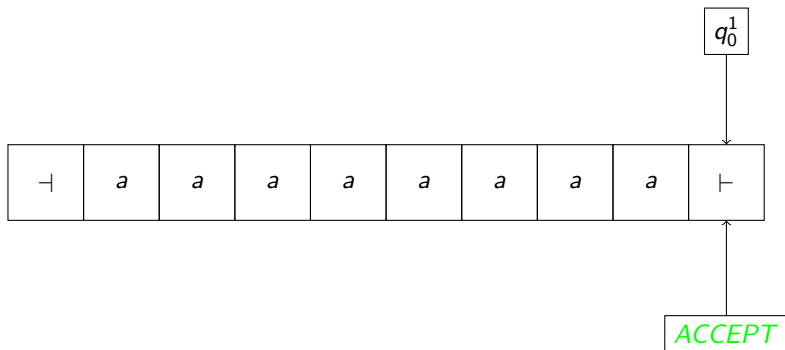
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



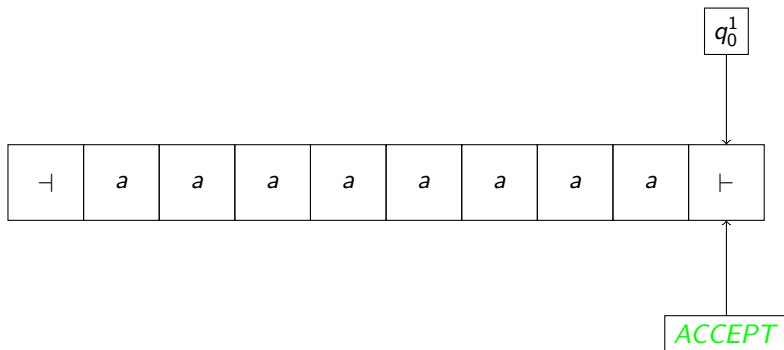
Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



Et Bidirectionnels? Et sur un alphabet unaire?

On reconnait $\{1^{2^n}, n \in \mathbb{N}\}$:



$\log(n)$ messages

Influence du nombre de message

Théoremes (Tomasz Jurdzinski)

- ▶ $1Mes_k(m) \subsetneq 2Mes_k(m+1)$

Influence du nombre de message

Théorèmes (Tomasz Jurdzinski)

- ▶ $1Mes_k(m) \subsetneq 2Mes_k(m+1)$
- ▶ $2Mes_2(m) \subsetneq 2Mes_k(m-1)$

Influence du nombre de message

Théorèmes (Tomasz Jurdzinski)

- ▶ $1Mes_k(m) \subsetneq 2Mes_k(m+1)$
- ▶ $2Mes_2(m) \subsetneq 2Mes_k(m-1)$
- ▶ *si $f(n) = o(\log n)$ et $f(n) = \omega(1)$ alors il n'existe pas de système **unidirectionnel** qui utilise $f(n)$ messages.*

Influence du nombre de message

Théorèmes (Tomasz Jurdzinski)

- ▶ $1Mes_k(m) \subsetneq 2Mes_k(m+1)$
- ▶ $2Mes_2(m) \subsetneq 2Mes_k(m-1)$
- ▶ si $f(n) = o(\log n)$ et $f(n) = \omega(1)$ alors il n'existe pas de système *unidirectionnel* qui utilise $f(n)$ messages.
- ▶ il existe une constante c telle que si $f(n) = o((\log \log \log n)^c)$ et $f(n) = \omega(1)$ alors il n'existe pas de système *bidirectionnel* qui utilise $f(n)$ messages.

Conjecture

Question

Qu'acceptent les FAS

Conjecture

Question

*Qu'acceptent les FAS
communications ?*

*avec un nombre **constant** de*

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

les langages rationnels (unaire).

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

les langages rationnels (unaire).

- ▶ unidirectionnel/bidirectionnel

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

les langages rationnels (unaire).

- ▶ unidirectionnel/bidirectionnel
- ▶ déterministe/non-déterministe

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

les langages rationnels (unaire).

- ▶ unidirectionnel/bidirectionnel
- ▶ déterministe/non-déterministe

Théoreme (Harrison & Ibarra)

Si un langage est accepté par un automate unaire unidirectionnel à tête multiple alors il est régulier.

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

les langages rationnels (unaire).

- ▶ unidirectionnel/**bidirectionnel**
- ▶ **déterministe/non-déterministe**

Conjecture

Question

Qu'acceptent les FAS *unaire* avec un nombre *constant* de communications ?

Conjecture

les langages rationnels (unaire).

- ▶ unidirectionnel/**bidirectionnel**
- ▶ **déterministe**/**non-déterministe**

Plan de la preuve

Plan de la preuve

- ▶ on introduit un modèle simplifié : *TSUDFAS*

Plan de la preuve

- ▶ on introduit un modèle simplifié : *TSUDFAS*
- ▶ on simule n'importe quel système unaire unidirectionnel déterministe par un *TSUDFAS* (sans trop augmenter le nombre de message).

Plan de la preuve

- ▶ on introduit un modèle simplifié : *TSUDFAS*
- ▶ on simule n'importe quel système unaire unidirectionnel déterministe par un *TSUDFAS* (sans trop augmenter le nombre de message).
- ▶ on montre que les *TSUDFAS* avec un nombre constant de messages reconnaissent exactement les langages rationnels

Comportements déterministes sur un alphabet unaire

La seule chose qui compte :

Comportements déterministes sur un alphabet unaire

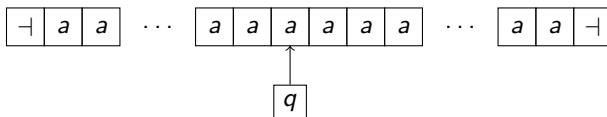
La seule chose qui compte :

la taille du mot.

Comportements déterministes sur un alphabet unaire

La seule chose qui compte :

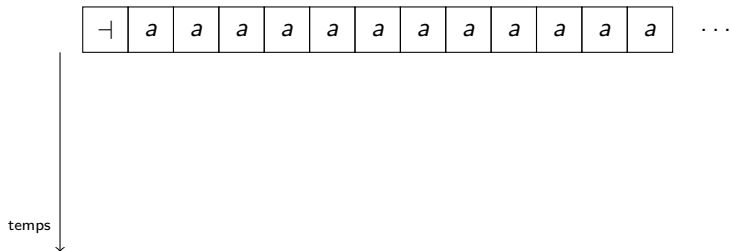
la taille du mot.



Comportements déterministes sur un alphabet unaire

La seule chose qui compte :

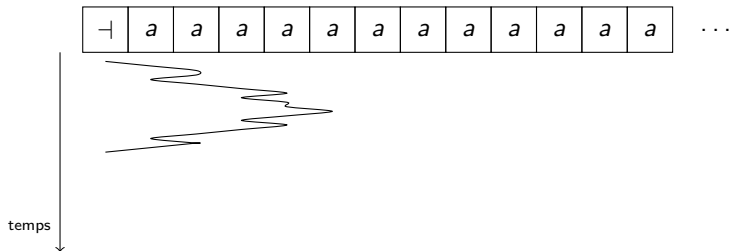
la taille du mot.



Comportements déterministes sur un alphabet unaire

La seule chose qui compte :

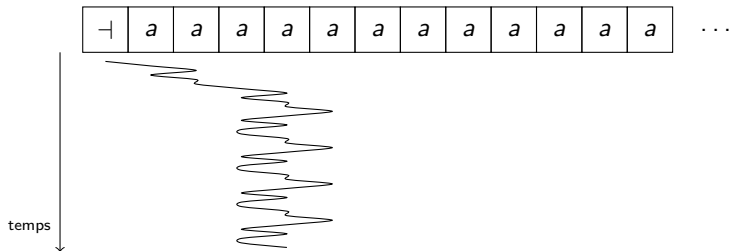
la taille du mot.



Comportements déterministes sur un alphabet unaire

La seule chose qui compte :

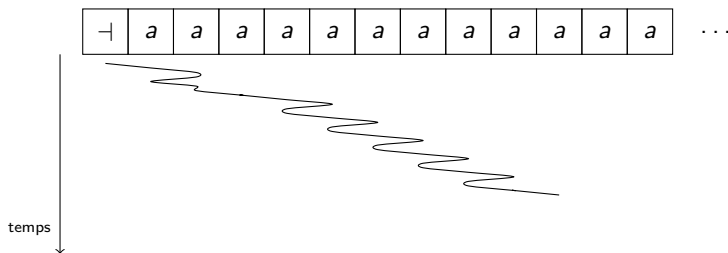
la taille du mot.



Comportements déterministes sur un alphabet unaire

La seule chose qui compte :

la taille du mot.



Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System

Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary **Deterministic Finite Automata System**

Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping **Unary** Deterministic Finite Automata System

Simulation des 2UDFAS par TSUDFAS

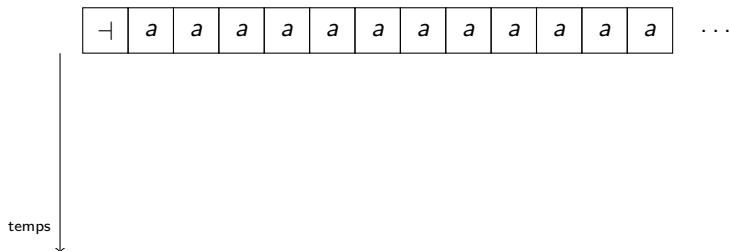
Tri-Phase **Sweeping** Unary Deterministic Finite Automata System

Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System

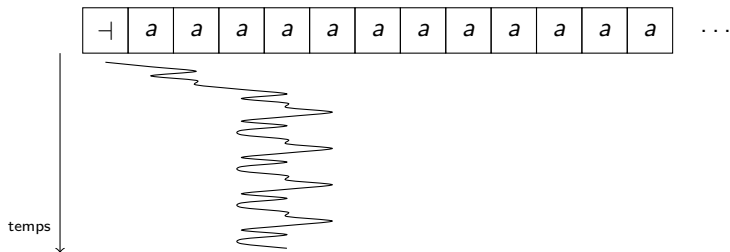
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



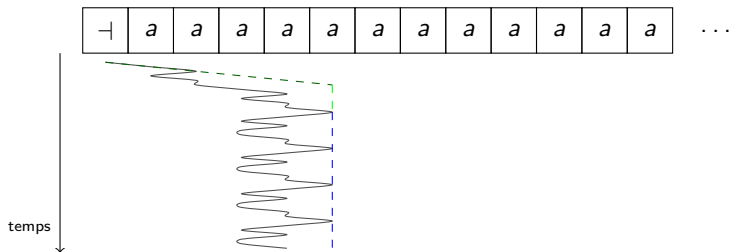
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



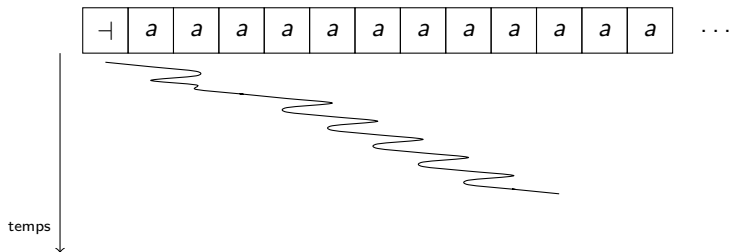
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



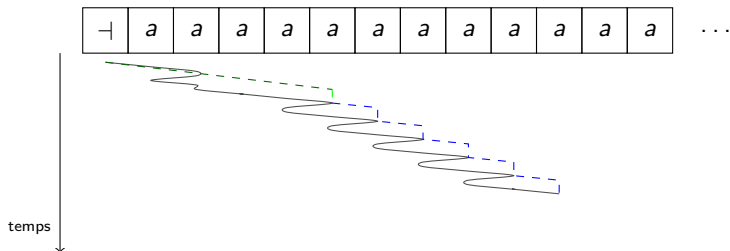
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



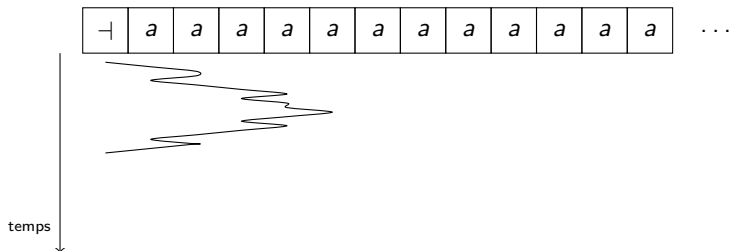
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



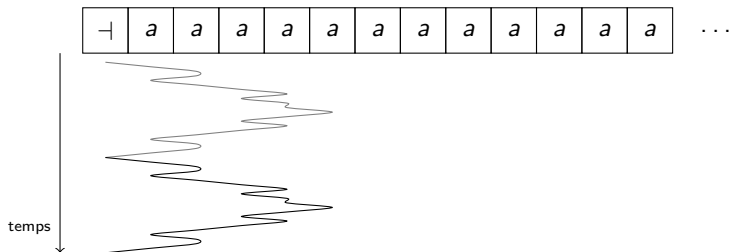
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



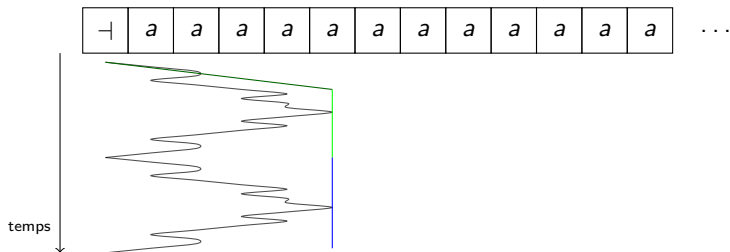
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



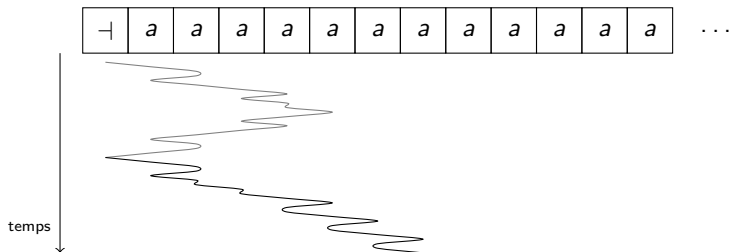
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System



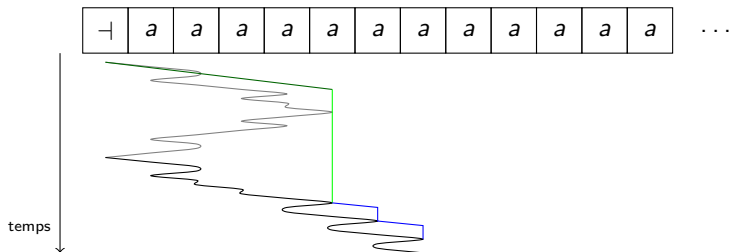
Simulation des 2UDFAS par TSUDFAS

Tri-Phase Sweeping Unary Deterministic Finite Automata System

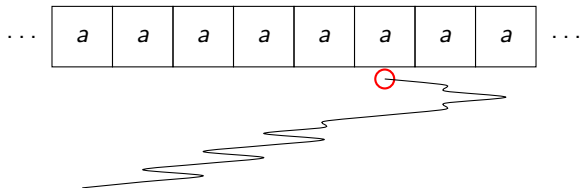


Simulation des 2UDFAS par TSUDFAS

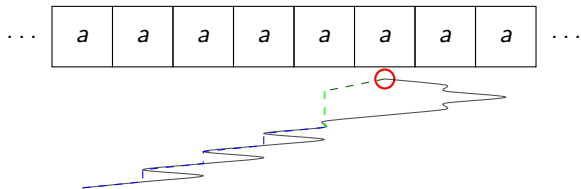
Tri-Phase Sweeping Unary Deterministic Finite Automata System



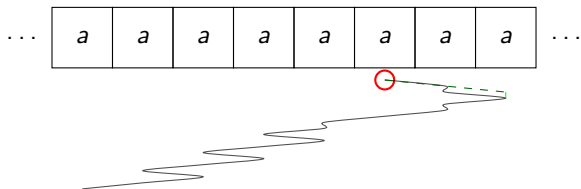
Simulation à partir du milieu



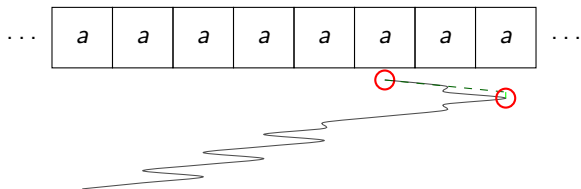
Simulation à partir du milieu



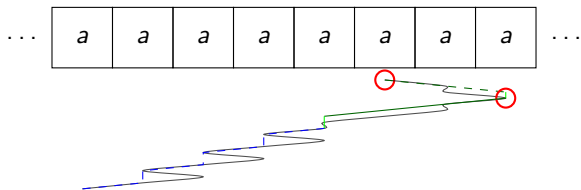
Simulation à partir du milieu



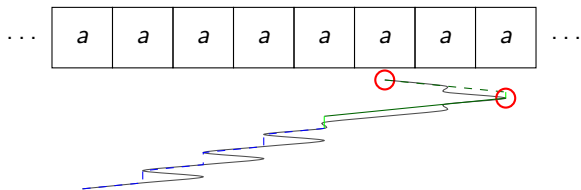
Simulation à partir du milieu



Simulation à partir du milieu

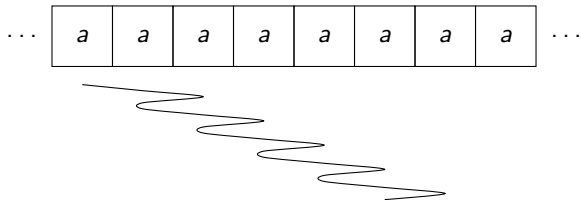


Simulation à partir du milieu

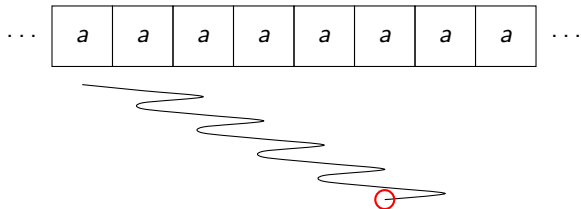


ajoute 1 message

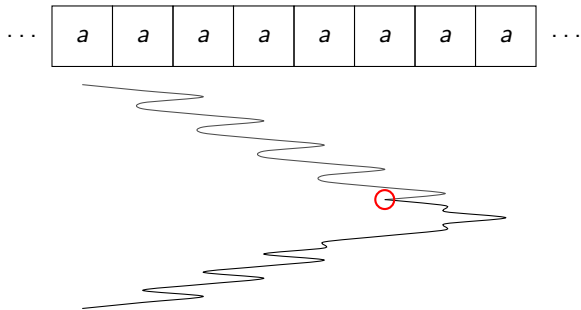
Réception d'un message



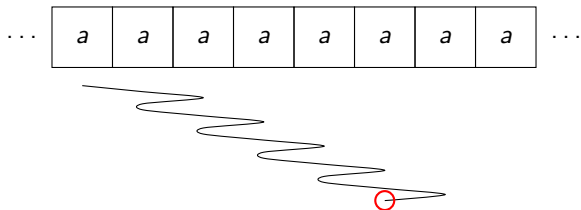
Réception d'un message



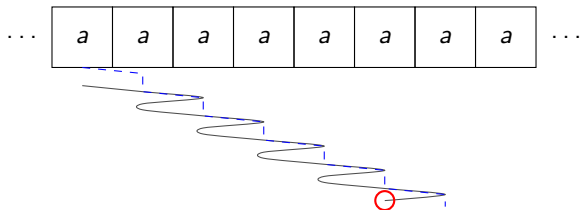
Réception d'un message



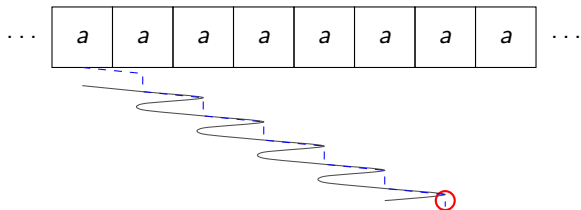
Réception d'un message



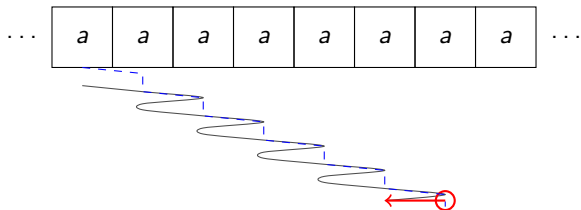
Réception d'un message



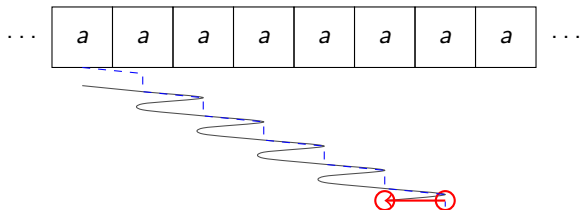
Réception d'un message



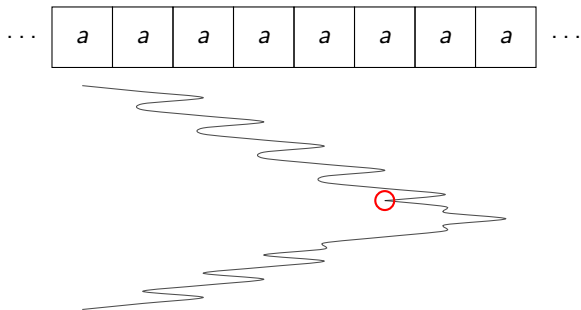
Réception d'un message



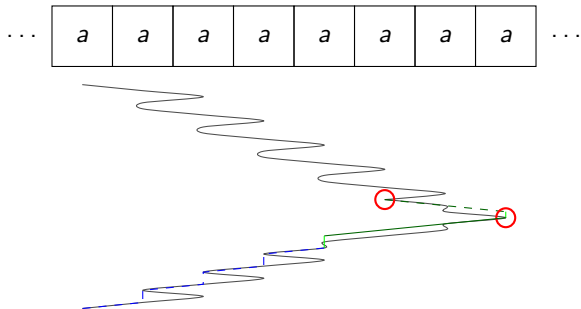
Réception d'un message



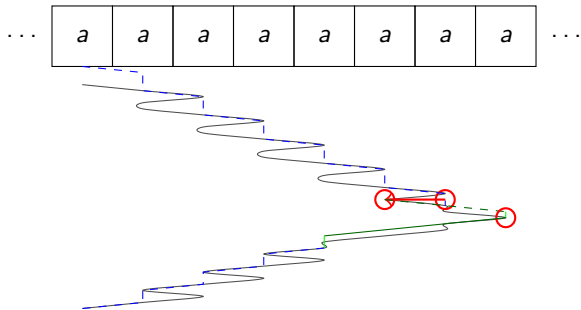
Réception d'un message



Réception d'un message



Réception d'un message



ajoute 2 messages

À quoi ça sert ?

Théoreme

*\mathcal{L} reconnu par un TSUDFAS
avec un nombre constant de messages* \Rightarrow *\mathcal{L} rationnel*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

Il existe $\pi_1 < \pi_2$

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

*Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que
si A fait s pas de calcul depuis une position p*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

*Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que
si A fait s pas de calcul depuis une position p
sans recevoir de messages*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

*Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que
si A fait s pas de calcul depuis une position p
sans recevoir de messages et sans atteindre un bord*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que

si A fait s pas de calcul depuis une position p

sans recevoir de messages et sans atteindre un bord, alors :

- ▶ *si $s < \pi_1$ alors sa position est $p + d * s$*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que

si A fait s pas de calcul depuis une position p

sans recevoir de messages et sans atteindre un bord, alors :

- ▶ *si $s < \pi_1$ alors sa position est $p + d * s$*
- ▶ *si $\pi_1 < s < \pi_2$ alors sa position est $p + \pi_1 * d$*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

*Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que
si A fait s pas de calcul depuis une position p
sans recevoir de messages et sans atteindre un bord, alors :*

- ▶ *si $s < \pi_1$ alors sa position est $p + d * s$*
- ▶ *si $\pi_1 < s < \pi_2$ alors sa position est $p + \pi_1 * d$*
- ▶ *si $\pi_2 < s$ alors sa position est
 $p + (\pi_1 + \left\lceil (s - \pi_2) * \frac{\Delta}{\omega} \right\rceil) * d$*

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

*Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que
si A fait s pas de calcul depuis une position p
sans recevoir de messages et sans atteindre un bord, alors :*

- ▶ *si $s < \pi_1$ alors sa position est $p + d * s$*
- ▶ *si $\pi_1 < s < \pi_2$ alors sa position est $p + \pi_1 * d$*
- ▶ *si $\pi_2 < s$ alors sa position est
 $p + (\pi_1 + \lceil (s - \pi_2) * \frac{\Delta}{\omega} \rceil) * d$*

Corollaire

Combien d'étapes avant le prochain marqueur de fin ?

- ▶ $d = +1 \Rightarrow \alpha * (n - p) + \beta$

Tout est plus simple

Lemme

Soit A une composante d'un TSUDFAS.

Soit q un état et d sa direction associée.

Il existe $\pi_1 < \pi_2$ et $\Delta < \omega$ tels que si A fait s pas de calcul depuis une position p sans recevoir de messages et sans atteindre un bord, alors :

- ▶ *si $s < \pi_1$ alors sa position est $p + d * s$*
- ▶ *si $\pi_1 < s < \pi_2$ alors sa position est $p + \pi_1 * d$*
- ▶ *si $\pi_2 < s$ alors sa position est $p + (\pi_1 + \left\lceil (s - \pi_2) * \frac{\Delta}{\omega} \right\rceil) * d$*

Corollaire

Combien d'étapes avant le prochain marqueur de fin ?

- ▶ $d = +1 \Rightarrow \alpha * (n - p) + \beta$
- ▶ $d = -1 \Rightarrow \alpha * p + \beta$

Des événements particuliers

Lemme

Entre deux messages il y a un nombre fini de configuration au bords.

Des événements particuliers

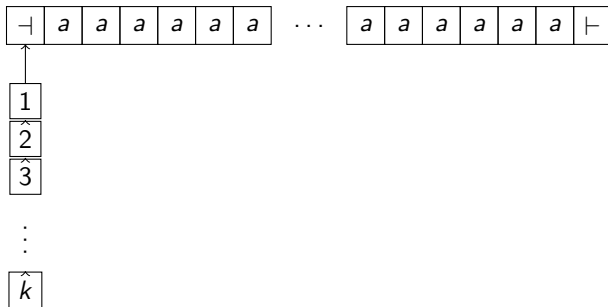
Lemme

Entre deux messages il y a un nombre fini de configuration au bords.

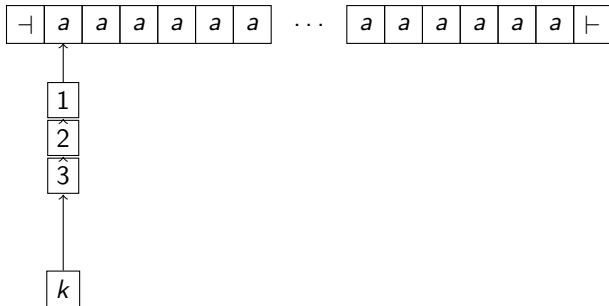
Corollaire

Nombre fini de message \Rightarrow nombre fini d'événement particulier

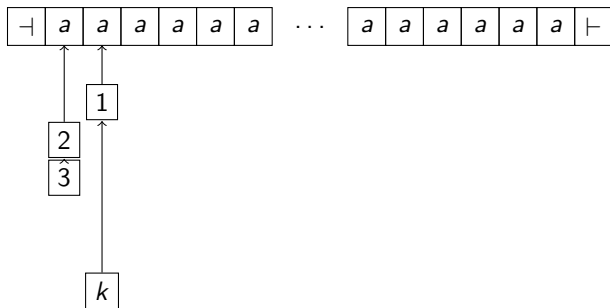
Tout est affine



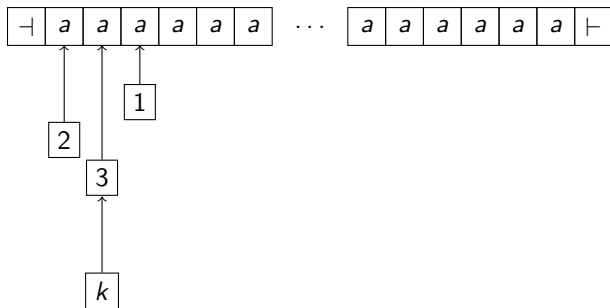
Tout est affine



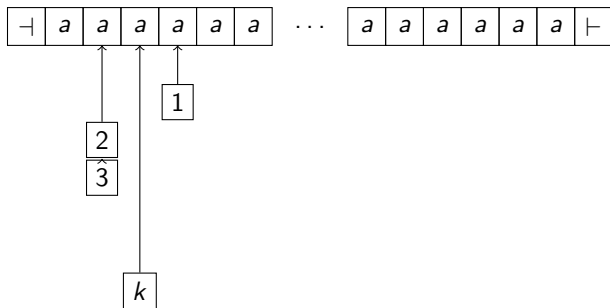
Tout est affine



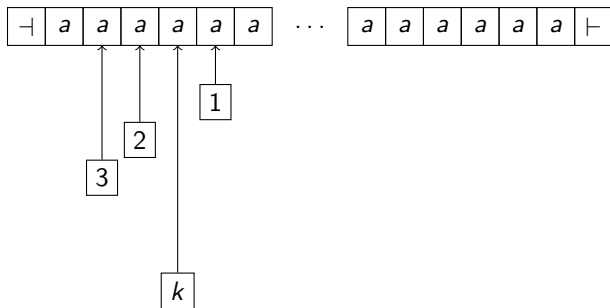
Tout est affine



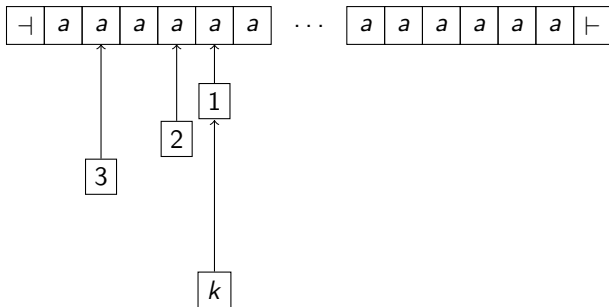
Tout est affine



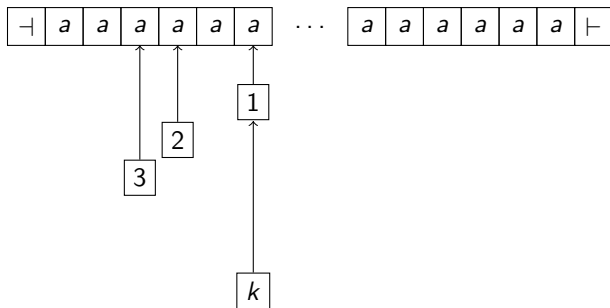
Tout est affine



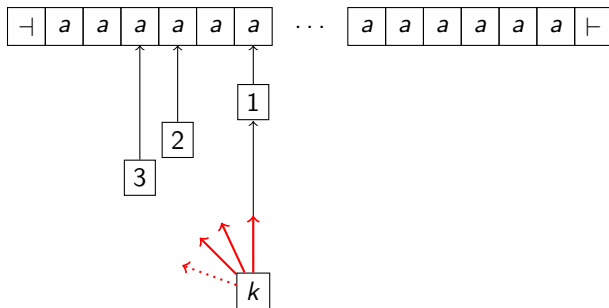
Tout est affine



Tout est affine

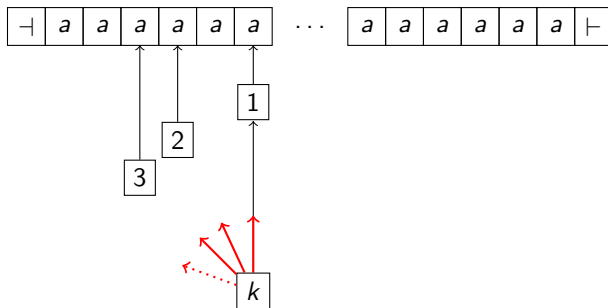


Tout est affine



après un nombre constant de pas de calcul

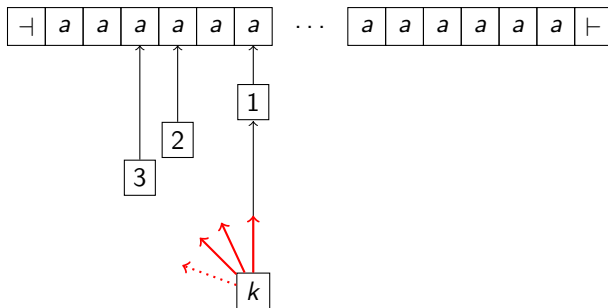
Tout est affine



après un nombre constant de pas de calcul

expression affine des positions de chaque automate

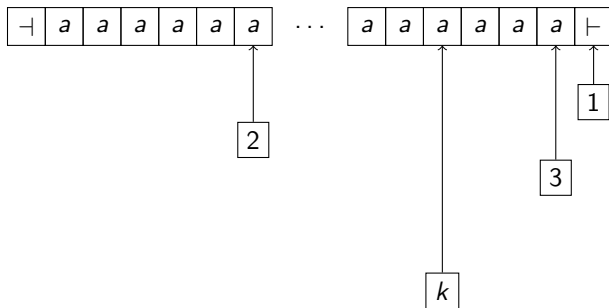
Tout est affine



après un nombre constant de pas de calcul

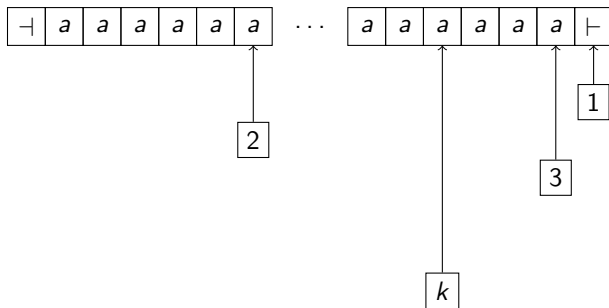
expression affine des positions de chaque automate
états connus

Tout est affine



après $\alpha * n + \beta$ pas de calcul

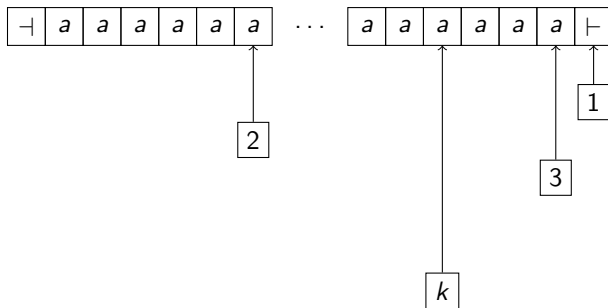
Tout est affine



après $\alpha * n + \beta$ pas de calcul

expression affine des positions de chaque automate

Tout est affine



après $\alpha * n + \beta$ pas de calcul

expression affine des positions de chaque automate
états connus selon congruence de n

Conclusion

Les $2UDFAS$ acceptent exactement les langages rationnels.

Conclusion

Les $2UDFAS$ acceptent exactement les langages rationnels.
Ce n'est pas encore le résultat espéré.

Conclusion

Les $2UDFAS$ acceptent exactement les langages rationnels.

Ce n'est pas encore le résultat espéré.

Objets simples, mais comportement complexes. . .

Questions ouvertes

Questions ouvertes

- ▶ Le cas non-déterministe ?

Questions ouvertes

- ▶ Le cas non-déterministe ? (conjecture : rationnel)

Questions ouvertes

- ▶ Le cas non-déterministe ?
 - ▶ Seul l'envoi des messages est non-déterministe ?

Questions ouvertes

- ▶ Y-a-t-il des langages unaires qui requiert $o(\log n)$ et $\omega(1)$?

Questions ouvertes

- ▶ Y-a-t-il des langages unaires qui requiert $o(\log n)$ et $\omega(1)$?
(conjecture non)

Questions ouvertes

- ▶ Influence du nombre d'automates dans le cas unaire ?

Merci pour votre attention

Avez-vous des questions ?