ENSAM
ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS
UNIVERSITÉ HASSAN II DE CASABLANCA

Algorithmique Avancée

Recherche de Motif dans une chaîne de caractères

Animé par : Dr. ibrahim GUELZIM
Email : ib.guelzim@gmail.com

Sommaire

- Rappels
  - Introduction et notions générales
  - Analyse et conception d'algorithmes
  - Complexité d'algorithmes classiques : 3 Tris de tableaux, 2 recherches dans un tableau, Schéma de Hörner
  - Preuves d'algorithmes
- Autres algorithmes de tri :
  - Tri par fusion
  - Tri par Tas
- Complexité moyenne :
  - Application au Tri rapide
  - Structures de Données Probabilistes :
    - Notions sur les Tables de Hachage et Fonctions de Hachage,
    - Bloom Filter,
    - Count Min Sketch
  - Programmation dynamique
- Traitements de chaines de Caractères :
  - **Recherche de motif dans une chaine de caractères**
  - Compression de données

Recherche de chaîne de caractères : Algorithme naif

- Soit un motif **P** (En : Pattern) de taille **m**,
- Soit **ch** une chaine de caractères de taille **n**,

- Question : Ecq P apparait dans ch ?

- Exemple :
  - P = 'GATTACA' (m = 7)
  - Ch = 'GTAGA … GCGATTACA …'

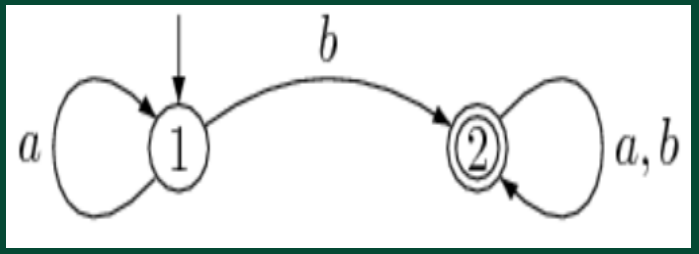
- Réponse 1 : Algorithme naif :
  - Pour chaque s = 1, 2 , … , n – m + 1 : vérifier si P est sous chaine de ch
  - ch[s..s+m-1] = P ?
  - Complexité : Θ(m x (n-m+1)) = Θ(mn)

Recherche de chaîne de caractères via fonctions de Hashage

- Réponse 2 : utilisation d'une Fonction de Hashage h (cf précèdent chap du cours) pour hasher le motif P
  - h(P) = r
  - Pour chaque s = 1, 2 , … , n – m + 1 :
    - Calculer q = h(ch [s..s+m-1])
    - Si r = q : comparer ch[s..s+m-1] à P
  - Complexité pire des cas : $O(m \times (n-m+1)) = \Theta(mn)$ n'est pas amélioré
- Réponse 3 : ≠ lors du calcul de q
  - itération j :
    - ch = '… $s_m s_{m-1} \dots s_2 s_1 s_0$ …'
    - $q = h(p, s_1) = p^{m-1}s_m + p^{m-2}s_{m-1} \dots + ps_2 + s_1$ [M] // prendre M nbr premier
  - itération j+1 :
    - ch = '… $s_m s_{m-1} \dots s_2 s_1 s_0$ …'
    - $q' = h(p, s_0) = p^{m-1}s_{m-1} + p^{m-2}s_{m-2} \dots + ps_1 + s_0$ [M]
    - $= p * (h(s) - p^{m-1}s_m)) + s_0$ [M]
    - Tps : 2 multiplications , 1 soustraction , 1 addition

Recherche de chaîne de caractères via un Automate Fini

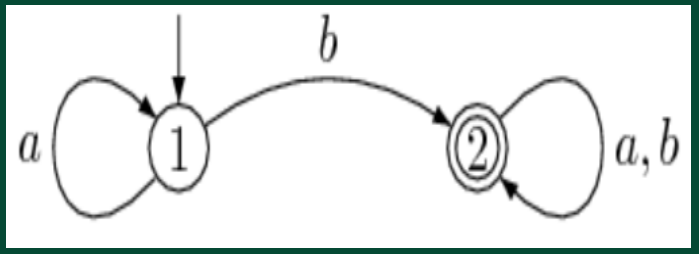
- Soit le schéma suivant représentant un traitement d'une chaine de caractères :
  - Les lettres sur les arcs représentent des actions à mener :
    - En l'occurrence :
      - a : **concaténer** la lettre **a** à une chaine de caractère
      - b : **concaténer** la lettre **b** à une chaine de caractère
    - Ces actions sont appelées : **transitions**



  - a,b représentent l'**alphabet** des actions qu'on peut mener
  - Les sommets représentent des situations particulières qu'on nomme : **états**
  - La flèche verticale sur l'état 1 signifie que c'est l'état de commencement : **état initial**
  - L'état 2 entouré deux fois (ou avec une flèche sortante) :
    - Signifie que la chaine de caractère qui s'arrête dessus, est une **chaine validée** par le schéma
    - Appelé **état final**
  - Ce schéma est appelé **automate fini**
  - Il valide (**reconnait**) tous les mots contenant au moins un "*b*"

Recherche de chaîne de caractères via un Automate Fini

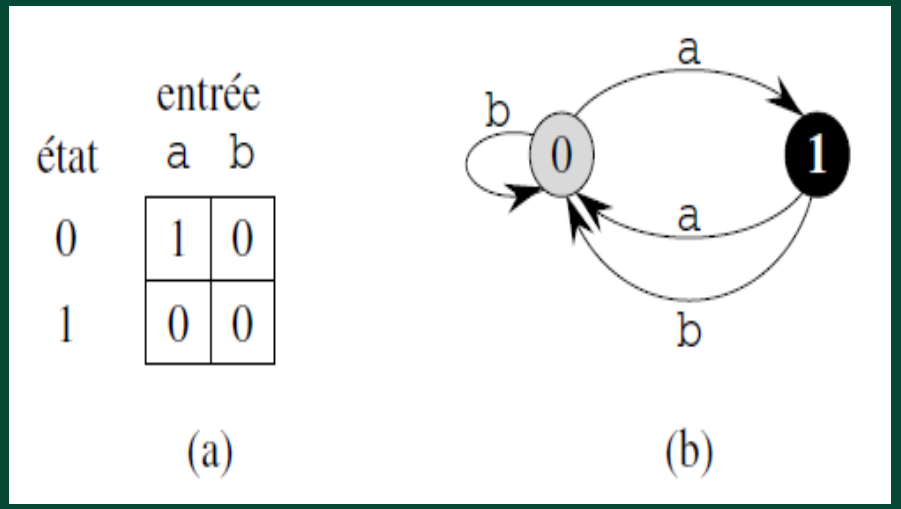
- Automate fini
  - Un automate fini M est un quintuplet : $(Q , q_0 , A , \Sigma , \delta)$
    - Q est un ensemble fini **d'états**,
    - $q_0 \in Q$ est l'état initial,
    - $A \subseteq Q$ est un ensemble distingué d'états **terminaux**,
    - Σ est un alphabet fini,
    - δ est une fonction de $Q \times \Sigma$ vers Q, appelée fonction de transition de M.



  - L'automate fini démarre à l'état q_0 et lit les caractères de la chaîne d'entrée un par un.
  - Si l'automate se trouve dans l'état q et lit le caractère a, il passe (« effectue une transition ») de l'état q à l'état $\delta(q, a)$.
  - Chaque fois que l'état courant q appartient à A, on dit que la machine M a accepté la chaîne lue jusqu'à cet endroit.
  - On dit d'une entrée qui n'est pas acceptée qu'elle est rejetée.
  - La figure 1 illustre ces définitions à l'aide d'un automate simple à deux états.

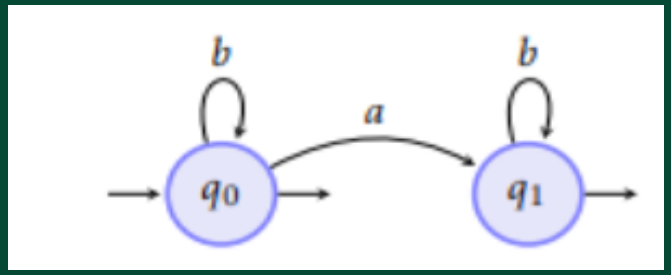
Recherche de chaîne de caractères via un Automate Fini

- Automate fini : exemple
  - Un automate fini simple à deux états pour l'alphabet Σ = {a, b},
  - l'ensemble d'états Q = {0, 1}
  - l'état initial q_0 = 0
  - (a) Une représentation tabulaire de la fonction de transition δ.
  - (b) Un diagramme équivalent de transitions d'état.
    - L'état 1 est le seul état d'acceptation (représenté en noir).
    - Les arcs représentent les transitions.
    - Par exemple, l'arc allant de l'état 1 à l'état 0 et étiqueté b indique δ(1, b) = 0.
    - Cet automate reconnaît les chaînes qui se terminent par un nombre impair de a.
    - Plus précisément, une chaîne x est acceptée SSI x = yz avec y = ε ou y se termine par b, et z = a^k où k est impair.
    - Par exemple, la séquence d'états dans lesquels entre cet automate pour l'entrée abaaa (état initial compris) est 0, 1, 0, 1, 0, 1. Cette entrée est donc acceptée.
    - Pour l'entrée abbaa, la séquence d'états est 0, 1, 0, 0, 1, 0 et elle est donc rejetée.

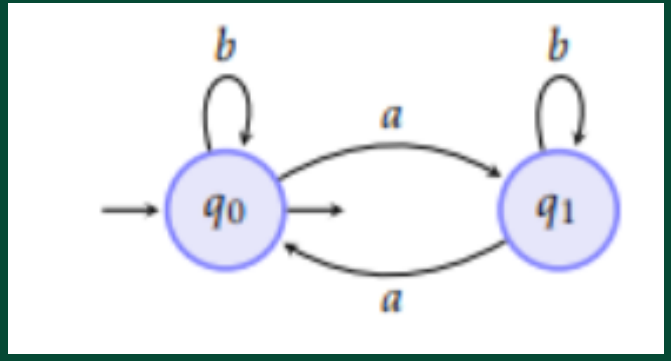


Recherche de chaîne de caractères via un Automate Fini

- Exemples
  - Langage des mots contenant au plus une fois la lettre a



  - Langage des mots contenant un nombre pair de fois la lettre a



Recherche de chaîne de caractères via un Automate Fini

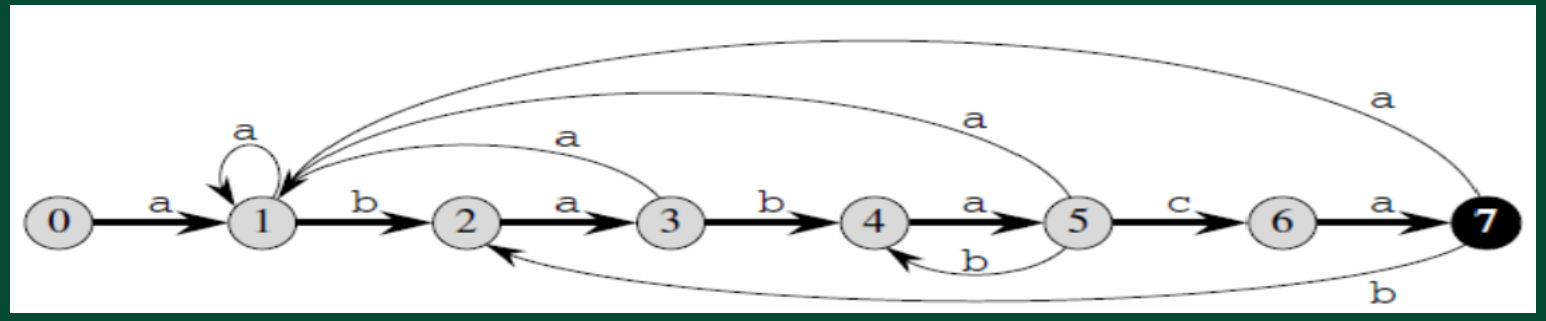
- Un automate fini M induit une fonction $\mathbf{f}$, appelée ***fonction d'état final***, de Σ^* vers Q telle que $\mathbf{f}(w)$ est l'état dans lequel est M après avoir traité la chaîne w.

- Donc, M reconnaît une chaîne w si et seulement si $\mathbf{f}(w) \in A$.

- La fonction $\mathbf{f}$ est définie par la relation récursive :
  - $\mathbf{f}(\varepsilon) = q_0$,
  - $\mathbf{f}(wa) = \delta(\mathbf{f}(w), a)$ pour $w \in \Sigma^*$, $a \in \Sigma$

Recherche de chaîne de caractères via un Automate Fini

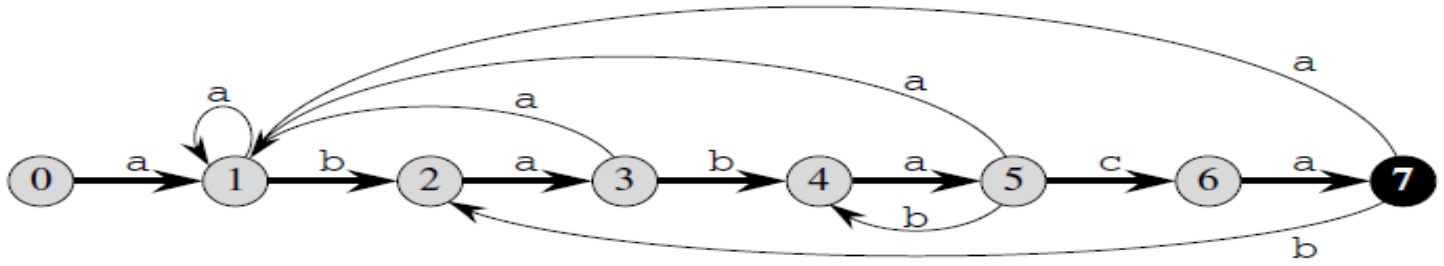
- Automates de recherche de motif :
  - Il existe un automate de recherche pour chaque motif *P* ;
  - Cet automate doit être construit à partir du motif lors d'une étape de pré traitement, avant de pouvoir être utilisé pour chercher le motif dans une chaîne textuelle.
  - La figure suivante illustre cette construction pour le motif *P* = ababaca.



- Dorénavant, nous supposerons que *P* est une chaîne donnée fixée à l'avance
- Pour alléger la notation, nous omettrons les références à *P*

Recherche de chaîne de caractères via un Automate Fini

- Automate de recherche du motif : ababaca



(a)

état	entrée a	b	c	P
0	1	0	0	a
1	1	2	0	b
2	3	0	0	a
3	1	4	0	b
4	5	0	0	a
5	1	4	6	c
6	7	0	0	a
7	1	2	0	

(b)

i	—	1	2	3	4	5	6	7	8	9	10	11
$T[i]$	—	a	b	a	b	a	b	a	c	a	b	a
état $\phi(T_i)$	0	1	2	3	4	5	4	5	6	**7**	2	3

(c)

Recherche de chaîne de caractères via un Automate Fini

- (a) : Diagramme de transition d'état pour l'automate de recherche de chaîne qui accepte toutes les chaînes finissant par ababaca.
  - L'état 0 est l'état initial et l'état 7 (en noir) est le seul état d'acceptation.
  - Un arc étiqueté *a*, partant de l'état *i* et arrivant à l'état *j*, représente $\delta(i, a) = j$.
  - Les arcs dirigés vers la droite forment le « squelette » de l'automate (dessiné en trait épais sur la figure) et correspondent aux comparaisons réussies entre le motif et les caractères d'entrée.
  - Les arcs dirigés vers la gauche correspondent aux comparaisons ayant échoué.
  - Certains de ces arcs ne sont pas représentés ;
  - Par convention, si un état *i* ne possède pas d'arc sortant étiqueté *a* pour un certain $a \in \mathbf{S}$, alors $\delta(i, a) = 0$.
- **(b)** La fonction de transition **d** correspondante et le motif *P* = ababaca.
  - Les entrées correspondant à des comparaisons réussies entre le motif et les caractères d'entrée sont représentées en gris.
- **(c)** L'action de l'automate sur le texte *T* = abababacaba
  - Sous chaque caractère *T*[*i*] du texte, on donne l'état **f**(*Ti*) de l'automate après traitement du préfixe *Ti*
  - Une occurrence du motif est trouvée, qui se termine à la position 9.

Recherche de chaîne de caractères via un Automate Fini

- Notations et terminologie:
  - Soit Σ^* , l'ensemble de toutes les chaînes de longueur finie utilisant les caractères de l'alphabet Σ (On ne s'intéresse qu'aux chaînes de longueur finie).
  - La **chaîne vide** de longueur zéro, notée ε, appartient à Σ^*.
  - La longueur d'une chaîne x est notée $|x|$.
  - La concaténation de deux chaînes x et y, notée xy, a pour longueur $|x| + |y|$ et est composée des caractères de x suivis des caractères de y.
  - On dit que la chaîne w est un **préfixe** de la chaîne x, notée $w \subset x$, si $\exists\, y \in \Sigma^*, x = wy$
    - Notez que si $w \subset x$, alors $| w | \leq | x |$
  - On dit que la chaîne w est un **suffixe** de la chaîne x, notée $w \supset x$, si $\exists\, y \in \Sigma^*, x = yw$
    - On déduit que $| w | \leq | x |$
  - La chaîne vide ε est à la fois suffixe et préfixe de toute chaîne.
- Exemples :
  - $ab \subset abcca$; $cca \supset abcca$
- À noter que, pour toutes chaînes x et y et pour tout caractère a, $x \supset y$ ssi $xa \supset ya$
- Notez également que $\subset$ et $\supset$ sont des relations transitives.

Recherche de chaîne de caractères via un Automate Fini

- Automates de recherche de motif :
  - Pour spécifier l'automate correspondant à une chaîne $P[1 . . m]$ donnée, on commence par définir une fonction auxiliaire σ, appelée ***fonction suffixe*** associée à P.
  - La fonction σ est une application de Σ^* vers $\{0, 1, . . . , m\}$ telle que $\sigma(x)$ est la **longueur du plus long préfixe de P qui est un suffixe de x**

- La fonction suffixe σ est bien définie, car la chaîne vide $P_0 = \varepsilon$ est un suffixe de n'importe quelle chaîne.

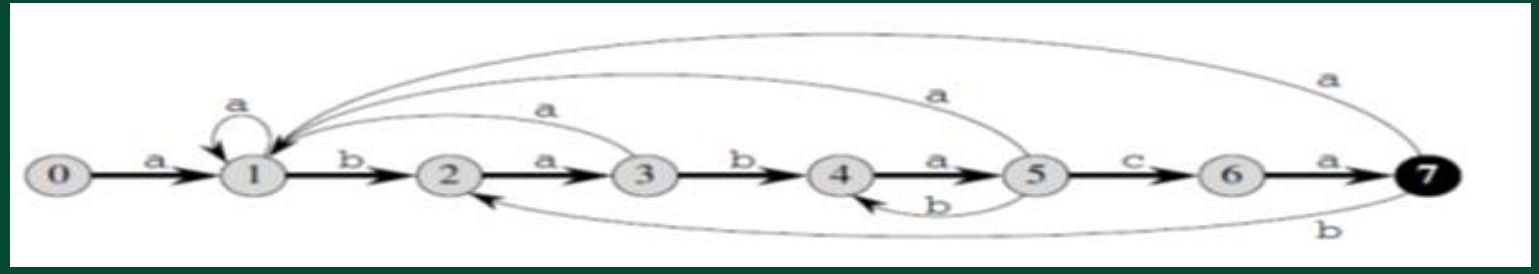
- Exemple, pour la chaîne P = ab on a :
  - σ(ε) = 0,
  - σ(ccaca) = 1
  - σ(ccab) = 2

Recherche de chaîne de caractères via un Automate Fini

- Notation et terminologie: Pour alléger la notation,
  - On note P_k le préfixe de longueur k, P[1..k] de la chaîne de caractères P[1..m], d'où
    - $P_0 = \varepsilon$
    - $P_m = P = P[1..m]$
  - On note T_k le préfixe de longueur k du texte T
  - Le problème de la recherche de caractères revient à trouver tous les décalages s dans l'intervalle $0 \leq s \leq n - m$, tq $P \supset T_{s+m}$,

Recherche de chaîne de caractères via un Automate Fini

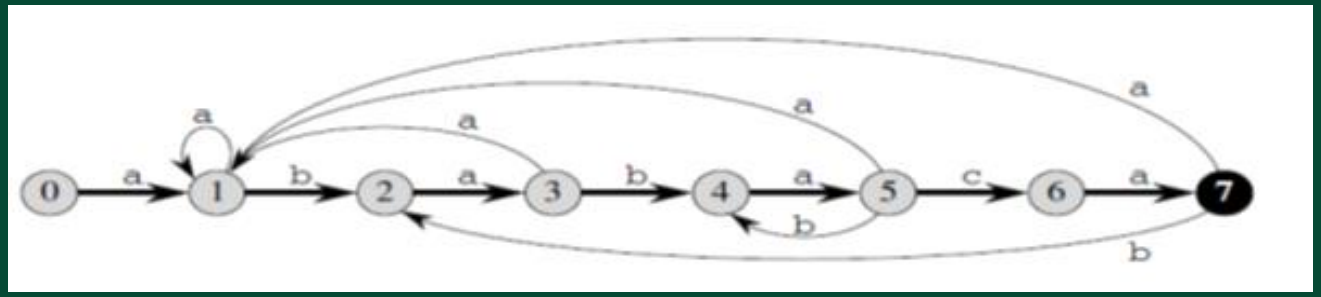
- Automates de recherche de motif :
  - On définit comme suit l'automate de recherche qui correspond à une chaîne $P[1 .. m]$ donnée.
  - L'ensemble des états Q est {0, 1, . . . ,m}.
  - L'état initial q_0 est l'état 0 et l'état m est le seul état d'acceptation.
  - La fonction de transition **δ** est définie par l'équation suivante : pour tout état q et tout caractère a : $\delta(q, a) = \sigma(P_q a)$
- *Exemple :*
  - Dans l'automate de recherche du motif P = 'ababaca', on a **δ**(5, b) = 4.
  - On fait cette transition car, si l'automate lit un b dans l'état $q = 5$,
  - alors $P_q b$ = ababab et le plus long préfixe de P qui est aussi un suffixe de ab**abab** est P_4 = abab.



Recherche de chaîne de caractères via un Automate Fini

- RECHERCHE-AUTOMATE-FINI(T, δ, m)

```
1 n ← longueur[T]
2 q ← 0
3 pour i ← 1 à n
4         faire q ← δ(q, T[i])
5             si q = m alors
6               s ← i – m
7 afficher « Le motif apparaît à la position » s
```



- CALCUL-FONCTION-TRANSITION δ (P, Σ)

```
1 m ← longueur[P]
2 pour q ← 0 à m
3         Pour chaque caractère x ∈ Σ faire
4                 k ← min(m + 1, q + 2)
5                     répéter k ← k – 1
6                     jusqu'à P_k ⊃ P_q x
7                 δ(q, x) ← k
8 retourner δ
```

- *Complexité : $O(m^3 |\Sigma|)$;*

Recherche de chaîne de caractères via un Automate Fini

- Exercice 1:
  - Construire l'automate de recherche du motif *P* = aabab et illustrer son action sur le texte
    T = aaababaabaababaab

- Exercice 2:
  - Écrire une fonction qui dit si une chaîne de caractères est suffixe d'une seconde chaîne de caractères

  - Ecrire un programme qui :
    - demande la longueur du motif, la longueur du texte, le motif (sur l'alphabet {a, b, c }) puis
    - génère un texte de façon aléatoire,
    - affiche les cent premiers caractères de ce texte puis,
    - indique le nombre d'occurrences du motif dans le texte, pour la méthode naïve d'abord et pour la méthode avec automate ensuite,
    - calcule le temps d'exécution en tics d'horloge

FIN