Algorithmique du texte

Notations et définitions

Cours basé sur le cours de L3 de Annie Chateau HLIN608

Sèverine Bérard

ISE-M – Université de Montpellier

Plan

- Introduction
- Définitions de base : mots et autres . . .
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*

- Introduction

- Définitions de base : mots et autres . . .

- À partir de mots

- Période et bord

- Recherche de motif ou *pattern matching*

- Le texte est l'une des représentations de l'information la plus simple et naturelle

- Le texte est l'une des représentations de l'information la plus simple et naturelle

- Les données à traiter se présentent souvent comme une suite de caractères (fichiers textes par exemple)

- Le texte est l'une des représentations de l'information la plus simple et naturelle

- Les données à traiter se présentent souvent comme une suite de caractères (fichiers textes par exemple)

- Les textes sont l'objet central du traitement de texte sous toutes ses formes

- L'algorithmique du texte a des champs d'application dans la plupart des sciences et à chaque fois qu'il y a du traitement de l'information à réaliser

- L'algorithmique du texte a des champs d'application dans la plupart des sciences et à chaque fois qu'il y a du traitement de l'information à réaliser

- La plupart des éditeurs de texte et des langages de programmation proposent des outils pour manipuler les textes (ou chaînes de caractères)

- L'algorithmique du texte a des champs d'application dans la plupart des sciences et à chaque fois qu'il y a du traitement de l'information à réaliser

- La plupart des éditeurs de texte et des langages de programmation proposent des outils pour manipuler les textes (ou chaînes de caractères)

- La complexité des algorithmes de traitement de texte est un domaine de recherche très actif, où la théorie et la pratique sont très proches !

- Introduction
- Définitions de base : mots et autres . . .
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*

Définition 1 (Alphabet) *Un alphabet Σ est un ensemble, non vide, fini ou infini de symboles*

Exemple 1 *Pour $k \geqslant 2$, l'alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$. $\Omega = \{a, b\}$ est encore un autre alphabet*

Mot (1)

Définition 1 (Alphabet) *Un alphabet Σ est un ensemble, non vide, fini ou infini de symboles*

Exemple 1 *Pour $k \geqslant 2$, l'alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$. $\Omega = \{a, b\}$ est encore un autre alphabet*

Définition 2 (Mot) *Un mot (ou chaîne de caractères) de l'alphabet Σ est une liste de symboles issus de Σ*

Exemple 2 *15423 est un mot de l'alphabet Σ_6, de l'alphabet Σ_{10}, mais pas de l'alphabet Σ_3. Les mots aaa, b, $ababbb$ sont des mots de Ω*

Remarques :

1. Un mot peut être fini ou infini
2. Un mot fini de longueur n peut être vu comme une application de $\{1, \ldots, n\}$ vers Σ
3. Un mot de longueur $n = 0$ est le mot vide, noté ϵ

Mot (2)

Remarques :

1. Un mot peut être fini ou infini
2. Un mot fini de longueur n peut être vu comme une application de $\{1, \ldots, n\}$ vers Σ
3. Un mot de longueur $n = 0$ est le mot vide, noté ϵ

Définition 3 (Ensemble des mots d'un alphabet) *L'ensemble des mots finis d'un alphabet Σ est noté Σ^*. L'ensemble des mots finis non vides de l'alphabet Σ est noté Σ^+*

Exemple 3 $\Omega^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$.
$\Sigma_3^* = \{\epsilon, 0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22, 000, 001, \ldots\}$

Mot (3)

Définition 4 (Longueur d'un mot) *Si w est un mot fini, sa longueur (i.e. le nombre de symboles contenus dans w) est notée $|w|$*

Exemple 4 *Le mot six est de longueur 3 et $anticonstitutionnellement$ de longueur 25. On a $|15423| = 5$ et $|\epsilon| = 0$*

Définition 4 (Longueur d'un mot) *Si w est un mot fini, sa longueur (i.e. le nombre de symboles contenus dans w) est notée $|w|$*

Exemple 4 *Le mot six est de longueur 3 et $anticonstitutionnellement$ de longueur 25. On a $|15423| = 5$ et $|\epsilon| = 0$*

Définition 5 (Occurrence d'un symbole) *Si $a \in \Sigma$ et $w \in \Sigma^*$, alors $|w|_a$ désigne le nombre d'occurrences du symbole a dans le mot w*

Exemple 5 $|anticonstitutionnellement|_t = 5$, $|anticonstitutionnellement|_a = 1$, $|anticonstitutionnellement|_w = 0$

Définition 6 (Concaténation de deux mots) *La concaténation de deux mots finis w et x est la juxtaposition des symboles de w et des symboles de x, notée wx ou $w.x$*

Exemple 6 *Si $w = anti$ et $x = pasti$ alors $wx = antipasti$*

Définition 6 (Concaténation de deux mots) *La concaténation de deux mots finis w et x est la juxtaposition des symboles de w et des symboles de x, notée wx ou $w.x$*

Exemple 6 *Si $w = anti$ et $x = pasti$ alors $wx = antipasti$*

Remarque : La concaténation n'est pas commutative $wx \neq xw$, mais elle est associative : $(xy)z = x(yz)$

Définition 6 (Concaténation de deux mots) *La concaténation de deux mots finis w et x est la juxtaposition des symboles de w et des symboles de x, notée wx ou $w.x$*

Exemple 6 *Si $w = anti$ et $x = pasti$ alors $wx = antipasti$*

Remarque : La concaténation n'est pas commutative $wx \neq xw$, mais elle est associative : $(xy)z = x(yz)$

Remarque : La concaténation est notée comme la multiplication, c'est-à-dire que w^n désigne $www\ldots w$ (n fois)

Définition 6 (Concaténation de deux mots) *La concaténation de deux mots finis w et x est la juxtaposition des symboles de w et des symboles de x, notée wx ou $w.x$*

Exemple 6 *Si $w = anti$ et $x = pasti$ alors $wx = antipasti$*

Remarque : La concaténation n'est pas commutative $wx \neq xw$, mais elle est associative : $(xy)z = x(yz)$

Remarque : La concaténation est notée comme la multiplication, c'est-à-dire que w^n désigne $www\ldots w$ (n fois)

L'ensemble Σ^* muni de la concaténation est un monoïde, avec comme élément identité le mot vide ϵ

- Introduction
- Définitions de base : mots et autres . . .
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*

Facteur, préfixe, suffixe

Définition 7 (Facteur) *On dit qu'un mot x est un facteur d'un mot w s'il existe des mots y et z tels que $w = yxz$*
Un synonyme de facteur est sous-mot

Définition 8 (Préfixe) *Le mot x est un préfixe du mot w s'il existe un mot y tel que $w = xy$*

Définition 9 (Suffixe) *Le mot x est un suffixe du mot w s'il existe un mot y tel que $w = yx$*

Facteur, préfixe, suffixe

Définition 7 (Facteur) *On dit qu'un mot x est un facteur d'un mot w s'il existe des mots y et z tels que $w = yxz$*
Un synonyme de facteur est sous-mot

Définition 8 (Préfixe) *Le mot x est un préfixe du mot w s'il existe un mot y tel que $w = xy$*

Définition 9 (Suffixe) *Le mot x est un suffixe du mot w s'il existe un mot y tel que $w = yx$*

Remarque : Les facteurs, préfixes et suffixes d'un mot w sont dits propres s'ils sont différents de w et de ϵ

Définition 7 (Facteur) *On dit qu'un mot x est un facteur d'un mot w s'il existe des mots y et z tels que $w = yxz$*
Un synonyme de facteur est sous-mot

Définition 8 (Préfixe) *Le mot x est un préfixe du mot w s'il existe un mot y tel que $w = xy$*

Définition 9 (Suffixe) *Le mot x est un suffixe du mot w s'il existe un mot y tel que $w = yx$*

Remarque : Les facteurs, préfixes et suffixes d'un mot w sont dits propres s'ils sont différents de w et de ϵ

Exemple 7 *On considère $w = algorithmique$. Le mot $x = algo$ est un préfixe de w, $y = ue$ est un suffixe de w, et $z = rithmi$ est un facteur de w*

Sous-séquence

Notations :

- Si $w = a_1 a_2 \dots a_n$ alors pour $i \in \{1, \dots, n\}$, on définit : $w[i] = a_i$
- Si $i \in \{1, \dots, n\}$ et $i \leqslant j \leqslant n$, on définit : $w[i..j] = a_i \dots a_j$

Remarque : $w[i..i] = a_i$ et si $j < i$ on définit $w[i..j] = \epsilon$

Sous-séquence

Notations :

- Si $w = a_1 a_2 \dots a_n$ alors pour $i \in \{1, \dots, n\}$, on définit : $w[i] = a_i$
- Si $i \in \{1, \dots, n\}$ et $i \leqslant j \leqslant n$, on définit : $w[i..j] = a_i \dots a_j$

Remarque : $w[i..i] = a_i$ et si $j < i$ on définit $w[i..j] = \epsilon$

Définition 10 (Sous-séquence) *Le mot x est une sous-séquence du mot w s'il existe $|x| + 1$ mots $y_0, y_1, \dots, y_{|x|}$ tels que $w = y_0 x[1] y_1 x[2] \dots y_{|x|-1} x[|x|] y_{|x|}$*

Exemple 8 *Toujours avec $w = algorithmique$. Les mots $x = amie$ et $y = aoiie$ sont des sous-séquences de w*

Attention : Sous-séquence $\neq$ sous-mot !

- Introduction
- Définitions de base : mots et autres . . .
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*

Définition 11 (Période) *Une période d'un mot w est un entier $0 < p \leqslant |w|$ tel que*

$$\forall i \in \{1, \ldots, |w| - p\}\, w[i] = w[i + p]$$

On note $period(w)$ la plus petite période de w

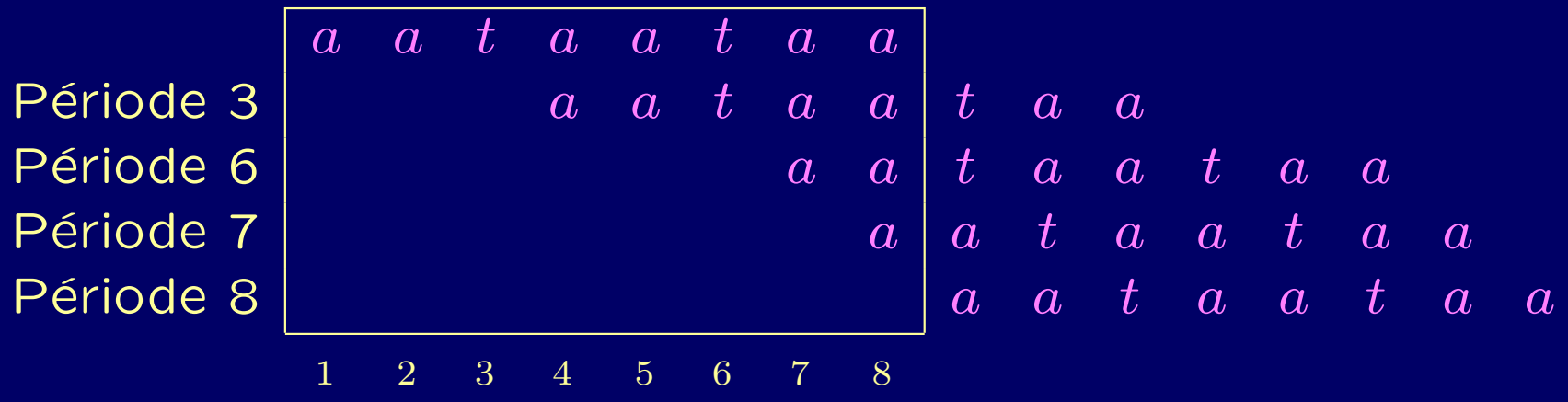
Exemple 9 *Les périodes de $aataataa$ (de longueur 8) sont 3, 6, 7 et 8*

Définition 11 (Période) *Une période d'un mot w est un entier $0 < p \leqslant |w|$ tel que*

$$\forall i \in \{1, \ldots, |w| - p\}\, w[i] = w[i + p]$$

On note $period(w)$ la plus petite période de w

Exemple 9 *Les périodes de $aataataa$ (de longueur 8) sont 3, 6, 7 et 8*



Proposition 1 *Soient x un mot non vide et p un entier tel que $0 < p \leqslant |x|$. Alors les 5 propriétés suivantes sont équivalentes :*

1. *L'entier p est une période de x*
2. *Il existe deux mots u et $v \neq \epsilon$ et un entier $k > 0$ tels que $x = (uv)^k u$ avec $|uv| = p$*
3. *Il existe un mot t et un entier $k > 0$ tels que x est un préfixe de t^k avec $|t| = p$*
4. *Il existe trois mots u, v et w tels que $x = uw = wv$ et $|u| = |v| = p$*
5. *Il existe un mot t tel que x est préfixe de tx et $|t| = p$*

Définition 12 (Bord) *Un bord du mot x est un facteur de x, différent de x, qui est à la fois un préfixe et un suffixe de x*

Exemple 10 *Les bords du mot $aataataa$ sont $aataa$, aa, a et ϵ*

Définition 12 (Bord) *Un bord du mot x est un facteur de x, différent de x, qui est à la fois un préfixe et un suffixe de x*

Exemple 10 *Les bords du mot $aataataa$ sont $aataa$, aa, a et ϵ*

Remarque : Bord et période sont des notions duales

Périodes	Bords
3	$aataa$
6	aa
7	a
8	ϵ

Périodes et bords du mot $aataataa$ de longueur 8

Bord maximal

Définition 13 (Bord maximal) *Soit x un mot non vide. On note $border(x)$ le plus grand bord de x*

On dit que x est sans bord si $border(x) = \epsilon$

Définition 13 (Bord maximal) *Soit x un mot non vide. On note $border(x)$ le plus grand bord de x*
On dit que x est sans bord si $border(x) = \epsilon$

Remarque : Le bord maximal d'un mot est le plus long chevauchement non trivial quand on essaye de faire coïncider x avec lui-même

Définition 13 (Bord maximal) *Soit x un mot non vide. On note $border(x)$ le plus grand bord de x*
On dit que x est sans bord si $border(x) = \epsilon$

Remarque : Le bord maximal d'un mot est le plus long chevauchement non trivial quand on essaye de faire coïncider x avec lui-même

Remarque : Puisque $border(x)$ est strictement plus petit que x, si on itère le processus on finit par tomber sur le mot vide ϵ.

Proposition 2 *Soit x un mot et $k \geqslant 0$ le plus petit entier tel que $border^k(x)$ est vide. Alors*

$$(border(x), border^2(x), \ldots, border^k(x))$$

est la suite de tous les bords de x ordonnés par longueur décroissante, et

$$(|x| - |border(x)|, |x| - |border^2(x)|, \ldots, |x| - |border^k(x)|)$$

est l'ensemble de toutes les périodes de x en ordre croissant

Proposition 2 *Soit x un mot et $k \geqslant 0$ le plus petit entier tel que $border^k(x)$ est vide. Alors*

$$(border(x), border^2(x), \ldots, border^k(x))$$

est la suite de tous les bords de x ordonnés par longueur décroissante, et

$$(|x| - |border(x)|, |x| - |border^2(x)|, \ldots, |x| - |border^k(x)|)$$

est l'ensemble de toutes les périodes de x en ordre croissant

Remarque : On a exactement $period(x) = |x| - |border(x)|$

Les résultats suivants sont utilisés dans les preuves combinatoires sur les mots.

Lemme 1 (Périodicité faible) *Soient p et q deux périodes d'un mot x. Si $p+q \leqslant |x|$, alors $pgcd(p,q)$ est aussi une période de x.*

Lemme 2 (Périodicité) *Soient p et q deux périodes d'un mot x. Si $p+q-pgcd(p,q) \leqslant |x|$, alors $pgcd(p,q)$ est aussi une période de x.*

- Introduction
- Définitions de base : mots et autres . . .
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*

La recherche de motif ou *pattern matching*

- Le problème le plus fondamental en algorithmique du texte est le *pattern matching*, ou recherche de motif dans un texte

La recherche de motif ou *pattern matching*

- Le problème le plus fondamental en algorithmique du texte est le *pattern matching*, ou recherche de motif dans un texte

- Il est nécessaire pour accéder à l'information

La recherche de motif ou *pattern matching*

- Le problème le plus fondamental en algorithmique du texte est le *pattern matching*, ou recherche de motif dans un texte

- Il est nécessaire pour accéder à l'information

- Il est comparable, en terme d'utilité, aux tris sur les structures de données, ou aux opérations arithmétiques . . .

La recherche de motif ou *pattern matching*

- Le problème le plus fondamental en algorithmique du texte est le *pattern matching*, ou recherche de motif dans un texte

- Il est nécessaire pour accéder à l'information

- Il est comparable, en terme d'utilité, aux tris sur les structures de données, ou aux opérations arithmétiques . . .

- L'expression la plus simple du problème : on recherche une chaîne de caractères de longueur m, le motif ou *pattern*, dans une autre chaîne de caractères de longueur $n \geqslant m$, le texte

La recherche de motif ou *pattern matching*

- Le problème le plus fondamental en algorithmique du texte est le *pattern matching*, ou recherche de motif dans un texte

- Il est nécessaire pour accéder à l'information

- Il est comparable, en terme d'utilité, aux tris sur les structures de données, ou aux opérations arithmétiques . . .

- L'expression la plus simple du problème : on recherche une chaîne de caractères de longueur m, le motif ou *pattern*, dans une autre chaîne de caractères de longueur $n \geqslant m$, le texte

- La longueur du motif et celle du texte peuvent être très grandes $\Rightarrow$ il est impératif de considérer la complexité des algorithmes

Quels types de recherche ?

Le motif peut être décrit simplement (motif exact) ou de manière à désigner plusieurs chaînes de caractères (motif approché)

Quels types de recherche ?

Le motif peut être décrit simplement (motif exact) ou de manière à désigner plusieurs chaînes de caractères (motif approché)

- Recherche d'expression régulière (regexp) (`grep`)

Quels types de recherche ?

Le motif peut être décrit simplement (motif exact) ou de manière à désigner plusieurs chaînes de caractères (motif approché)

- Recherche d'expression régulière (regexp) (`grep`)
- Recherche de répétitions ou plus généralement de régularité (`sed`)

Quels types de recherche ?

Le motif peut être décrit simplement (motif exact) ou de manière à désigner plusieurs chaînes de caractères (motif approché)

- Recherche d'expression régulière (regexp) (`grep`)
- Recherche de répétitions ou plus généralement de régularité (`sed`)
- Recherche de motifs multiples (`awk`)

Quels types de recherche ?

Le motif peut être décrit simplement (motif exact) ou de manière à désigner plusieurs chaînes de caractères (motif approché)

- Recherche d'expression régulière (regexp) (`grep`)
- Recherche de répétitions ou plus généralement de régularité (`sed`)
- Recherche de motifs multiples (`awk`)
- Recherche de motifs arborescents (`lex`)

Quels types de recherche ?

Le motif peut être décrit simplement (motif exact) ou de manière à désigner plusieurs chaînes de caractères (motif approché)

- Recherche d'expression régulière (regexp) (`grep`)
- Recherche de répétitions ou plus généralement de régularité (`sed`)
- Recherche de motifs multiples (`awk`)
- Recherche de motifs arborescents (`lex`)
- Comparaison de textes : plus longue sous-séquence commune (PLSC) (`diff`) ou calcul de la distance d'édition (`diff -ed`)

Recherche de motif exact

Voici le problème que nous allons aborder lors des prochaines séances :

Trouver toutes les occurrences d'un motif P de longueur m à l'intérieur d'un texte T de longueur n

Voici le problème que nous allons aborder lors des prochaines séances :

Trouver toutes les occurrences d'un motif P de longueur m à l'intérieur d'un texte T de longueur n

1. Algorithme naïf
2. Algorithmes de Morris-Pratt (MP) et Knuth-Morris-Pratt (KMP)
3. Algorithme de Boyer-Moore
4. Tour d'horizon des autres algorithmes de recherche exacte de motif