

Algorithmique du texte

Notations et définitions

Sèverine Bérard



ISE-M – FDS, Université de Montpellier



Introduction

2

- 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

Introduction

3

- Introduction
- Définitions de base : mots et autres ...
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*
- 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 $k \geq 2$, l'alphabet $\Sigma_k = \{0, 1, \dots, k-1\}$. $\Omega = \{a, b\}$ 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 $\{1, \dots, n\}$ 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 $\Omega^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$.
 $\Sigma_3^* = \{\epsilon, 0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22, 000, 001, \dots\}$

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 $|\text{anticonstitutionnellement}|_t = 5$,
 $|\text{anticonstitutionnellement}|_a = 1$, $|\text{anticonstitutionnellement}|_w = 0$

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 $wx = antipasti$

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 $www \dots w$ (n fois)

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

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

Notations :

- Si $w = a_1a_2 \dots a_n$ alors pour $i \in \{1, \dots, n\}$, on définit : $w[i] = a_i$
- Si $i \in \{1, \dots, n\}$ et $i \leq j \leq n$, 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 $w = y_0x[1]y_1x[2] \dots y_{|x|-1}x[|x|]y_{|x|}$

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

Attention : Sous-séquence \neq sous-mot !

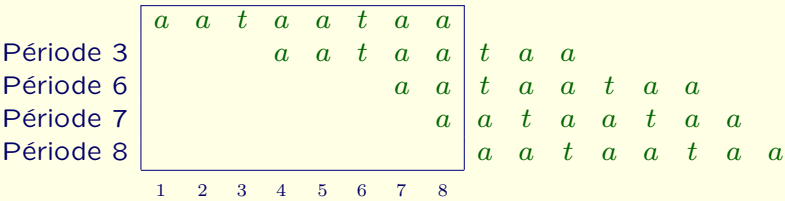
- Introduction
- Définitions de base : mots et autres ...
- À partir de mots
- Période et bord
- Recherche de motif ou *pattern matching*

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

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

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 $0 < p \leq |x|$. 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 $x = uw = vw$ 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

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

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 .

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 $p + q \leq |x|$, alors $\text{pgcd}(p, q)$ est aussi une période de x .

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

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

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

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

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

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

Remarque : On a exactement $\text{period}(x) = |x| - |border(x)|$.

- Introduction
- Définitions de base : mots et autres ...
- À partir de mots
- Période et bord
- 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 $n \geq m$, le texte
- La longueur du motif et celle du texte peuvent être très grandes \Rightarrow il est impératif de considérer la complexité des algorithmes

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

22

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