# 1 Introduction

On cherche des algorithmes simples qui permettent de répondre à des questions de type oui/non. Le mot « simple » signifie ici « à mémoire bornée » : le nombre de cases mémoires utilisé est indépendant des entrées de l'algorithme et leur contenu est lui-même borné indépendamment des entrées. On étudie quelques exemples :

— Soit $m$ un mot de la langue française (sans accent) connu par la liste de ses lettres : $m = (c, a, m, l)$ par exemple. Donnez un algorithme simple qui permet de savoir si en lisant le mot de gauche à droite, le mot contient trois consonnes consécutives (comme $(e, n, c, r, e)$).

— De même, donnez un algorithme simple qui permet de savoir si en lisant le mot de gauche à droite, l'ordre de première apparition des voyelles est l'ordre alphabétique.
Par exemple, le mot $(b, a, t, e, a, u)$ est acceptable, car l'ordre de première apparition de ses voyelles est $a, e, u$ ;
le mot $(a, g, i, l, e)$ ne l'est pas, car le premier $i$ est apparu avant le premier $e$.

— Soit $n$ un entier naturel connu par son écriture en base 2 : $n = \overline{c_k \dots c_0}$. Donnez deux algorithmes simples qui déterminent si $n$ est de la forme $3k + 1$, suivant qu'on lise les chiffres de gauche à droite ou de droite à gauche.

# 2 Mots et langages

## 2.1 Définitions et vocabulaire

Soit $A$ un ensemble fini, appelé alphabet. Les éléments de $A$ sont appelés des lettres.

Un mot sur $A$ est une suite finie de lettres de $A$, éventuellement vide. On note en général $\varepsilon$ le mot vide. On note les mots sous la forme de lettres accolées $u = a_1 \dots a_n$ où $(a_1, \dots, a_n) \in A^n$.

**Exemples**

a) $A$ est l'alphabet français en minuscule : `mot`, `langagecaml`, `zerumbert`, `bgftttaaaaajdga` sont des mots sur ce langage ;

b) $A$ est l'ensemble des mots de la langue française : une phrase au sens classique comme `["je";"suis";"un";"vieux";"croulant"]` est un mot sur $A$ ;

c) $A$ est l'ensemble des entiers $[\![0, 9]\!]$ : 123, 000057 sont des mots sur $A$ ;

d) $A$ est l'ensemble des quatre bases $A, C, G, T$ : une molécule d'ADN est un mot sur $A$ ;

e) $A$ est l'ensemble des symboles $\square, \blacksquare, \backslash n$ ($\backslash n$ symbolise le retour à la ligne) : une grille de mots croisés est un mot sur $A$.

L'ensemble des mots sur $A$ est noté $A^*$.

**Définition.** Si $m$ est un mot, on note $|m|$ la longueur du mot, *i.e.* le nombre de lettres du mots (donc $|\varepsilon| = 0$). De même, si $a$ est une lettre, on note $|m|_a$ la longueur en $a$ du mot $m$, c'est-à-dire le nombre de lettres $a$ dans le mot $m$.

Si $n$ est un entier naturel, $A^n$ est l'ensemble des mots de longueur $n$ sur $A$, donc $A^* = \bigcup_{n \in \mathbb{N}} A^n$.

$A^0 = \{\varepsilon\}$ : on note $A^+ = \bigcup_{n \in \mathbb{N}^*} A^n$, ensemble des mots non vides.

## 2.2 Opération sur les mots

Si $u, v$ sont deux mots sur $A$, on note $u.v$ ou plus simplement $uv$ la concaténation des deux mots :

$$\text{si } u = a_1 \dots a_n \text{ et } v = b_1 \dots b_p, \text{ alors } uv = a_1 \dots a_n b_1 \dots b_p.$$

On munit ainsi $A^*$ d'une loi de composition interne associative, ayant pour neutre $\varepsilon$ (on dit que $A^*$ est un monoïde unitaire). Dans cet ensemble, tous les éléments sont réguliers :

**Proposition 1** *Soit $u, v, w$ sont trois mots sur $A$, alors*

$$\begin{aligned} uv = uw &\Rightarrow v = w \\ vu = wu &\Rightarrow v = w \end{aligned}$$

Plus généralement, on a le résultat presque évident suivant :

**Proposition 2** *Soit $u, v, w, x$ quatre mots tels que $uv = wx$. Alors*
- *soit il existe un mot $t$ tel que $w = ut$ et $v = tx$*
- *soit il existe un mot $t$ tel que $u = wt$ et $x = tv$.*

La loi étant associative, on définit sans ambiguïté la puissance d'un mot :

$$u^n = \begin{cases} \varepsilon & \text{si } n = 0 \\ u^{n-1}.u & \text{si } n \in \mathbb{N}^* \end{cases}$$

En revanche, $A^*$ n'est pas un monoïde commutatif (dès que $A$ a plus de deux éléments).

**Proposition 3** *Soit $u, v$ deux mots sur $A$.*
*Alors $uv = vu$ si et seulement si il existe un mot $w$ et deux entiers naturels $p, q$ tels que $u = w^p$ et $v = w^q$.*

**Définition.** Soit $u, v$ deux mots, on dit que
- $u$ est un préfixe de $v$ quand il existe $w \in A^*$ tel que $v = uw$ (préfixe strict ou propre quand $w \neq \varepsilon$)
- $u$ est un suffixe de $v$ quand il existe $w \in A^*$ tel que $v = wu$ (suffixe strict ou propre quand $w \neq \varepsilon$)
- $u$ est un facteur de $v$ quand il existe $w, w' \in A^*$ tels que $v = wuw'$ (facteur strict ou propre quand $w \neq \varepsilon$ ou $w' \neq \varepsilon$).

# 3 Langages

## 3.1 Généralités

**Définition.** Un langage sur $A$ est un ensemble de mots sur $A$, donc une partie de $A^*$.

**Exemples**

a) L'ensemble des mots contenant trois consonnes consécutives est le langage défini dans le premier point de l'introduction ;

b) L'ensemble des représentations binaires des entiers divisibles par 3 est le langage défini dans le troisième point.

Les langages étant des ensembles, on peut leur appliquer les opérations classiques : union, intersection, complémentaire, différence.

On ajoute l'opération de concaténation :

$$\text{si } L, L' \text{ sont deux langages sur } A, \text{ on note } L.L' = LL' = \{uv \;/\; u \in L, v \in L'\}$$

On peut alors définir les puissances d'un langage par récurrence :

$$\text{si } L \text{ est un langage sur } A \text{ et } n \text{ un entier naturel, } L^n = \begin{cases} \{\varepsilon\} & \text{si } n = 0 \\ L^{n-1}.L & \text{si } n \in \mathbb{N}^* \end{cases}$$

### 3.2 Étoile d'un langage

Enfin, on définit l'itéré (ou l'étoile) d'un langage : $L^* = \bigcup_{n\in\mathbb{N}} L^n$.

On définit de même l'itéré strict (ou l'étoile stricte) de $L$ : $L^+ = \bigcup_{n\in\mathbb{N}^*} L^n$.

***Remarque.*** $L^* = \{\varepsilon\} \cup L^+$, donc si $\varepsilon \in L$, alors $L^* = L^+$ (et réciproquement).
$L^*$ est appelé aussi la clôture de Kleene de $L$ (ou étoile de Kleene).

Avec ces notations, on peut noter en un nombre fini de symboles des langages potentiellement infinis. Par exemple, si $A$ est l'alphabet français minuscule sans accent et $V = \{a, e, i, o, u, y\}$, $C = A\backslash V$, le langage des mots contenant trois consonnes consécutives peut s'écrire $A^*C^3A^*$, celui des mots commençant par une consonne et terminant par deux voyelles est noté $CA^*V^2$. Il est sans doute plus difficile d'écrire le langage du second exemple traité dans l'introduction : par exemple, le langage des mots où la première voyelle est un $a$, la deuxième nouvelle étant un $e$, les autres apparaissant ensuite dans n'importe quel ordre peut s'écrire $C^*\{a\}(C \cup \{a\})^*\{e\}A^*\ldots$

**Définition.** Un langage $L$ est dit clos par concaténation quand pour tout $(u, v) \in L^2$, $uv \in L$.

**Proposition 4** *Si $L$ est un langage, $L^*$ est le plus petit langage contenant $L$ et clos par concaténation.*

### 3.3 Quelques règles de calcul sur les langages

**Proposition 5** *Soient $L, M, N$ trois langages sur un alphabet $A$ . Alors :*
- $(LM)N = L(MN)$ ;
- $(L \cup M)N = LN \cup MN$ *et* $L(M \cup N) = LM \cup LN$ ;
- $(L \cap M)N \subset LN \cap MN$, *l'inclusion réciproque est fausse en général ;*
- $L \subset M \Rightarrow L^* \subset M^*$ ;
- $(L^*)^* = L^*$ ;
- $(L \cup M)^* = (L^*M^*)^*$.

## 4 Motifs, expressions régulières

Les algorithmes décrits dans l'introduction sont tous des algorithmes de décision : ils permettent de savoir si un mot appartient à un langage donné pas trop compliqué à décrire.

Un objectif de ce cours est de construire un méta-algorithme qui prend en paramètre un langage et qui construit un algorithme qui reconnaît si un mot appartient au langage. Le langage en paramètre n'est pas quelconque, il doit être descriptible dans un formalisme adapté. En effet, si est un alphabet fini, $A^*$ est un ensemble infini dénombrable et donc l'ensemble des langages $\mathscr{P}(A^*)$ est un ensemble infini non dénombrable. Or l'ensemble des langages qu'on peut reconnaître est dénombrable, car l'ensemble des algorithmes est lui-même dénombrable. Il existe donc des langages pour lesquels il n'y a aucun moyen automatique de répondre au problème de décision évoqué ci-dessus.

On va donc restreindre l'étude à certains langages effectivement accessibles.

### 4.1 Expressions régulières

**Définition.** Soit $A$ un alphabet. On définit par induction l'ensemble $ER$ des expressions régulières (standard) sur $A$ :
- $\varnothing$, $\varepsilon$ et les lettres de $A$ sont des expressions régulières ;
- si $e$, $f$ sont deux expressions régulières, alors $(e|f)$, $(e.f)$ et $(e*)$ sont des expressions régulières.

**Exemples**

a) Si $a$, $b$ sont deux lettres de $A$, alors $((\varepsilon|(b.a)).(a*))$, $((a.b)|(b.a))$, $(((a|b.a)*)|((b.a).b))*$ sont des expressions régulières.

Par convention, pour éviter les nombreuses parenthèses qui compliquent les notations, on convient d'un ordre de priorité des opérations :

$*$ est prioritaire sur ., qui est lui-même prioritaire sur $|$, et on omet souvent les points .

Ainsi, $ab|c*$ signifie $((a.b)|(c*))$, $a(b|c)*$ signifie $(a.((b|c)*))$, etc.

Enfin, on pose : $e+ = ee*$. Et bien sûr, $e^n$ désigne $e \dots e$ $n$ fois.

***Remarque.*** L'opérateur $|$ est parfois noté $+$ : dans ce cas, on n'utilise pas le symbole $a+$ pour ne pas confondre...

Les expressions régulières servent à décrire en une suite de symboles un langage.

## 4.2 Langages réguliers

**Définition.** Soit $A$ un alphabet. On associe inductivement à toute expression régulière (standard) $e$ sur $A$ un langage $L(e)$ :

- $L(\varnothing) = \varnothing$ ; $L(\varepsilon) = \{\varepsilon\}$ ; pour tout $a \in A$, $L(a) = \{a\}$ ;
- si $e$, $f$ sont deux expressions régulières, alors
  - $L(ef) = L(e)L(f)$
  - $L(e|f) = L(e) \cup L(f)$
  - $L(e*) = L(e)^*$

On dit que l'expression régulière dénote (ou représente) le langage $L(e)$.

Un langage image par $L$ d'une expression régulière est appelé langage régulier.

Par conséquent : $L(e+) = L(e)^+$

**Exemples**

a) Si $A = \{a, b\}$, $A^*$ est un langage régulier : $A^* = L((a|b)*)$

b) Avec le même alphabet, $L(aab|baa) = \{aab, baa\}$

c) Avec le même alphabet, le langage des mots contenant trois fois consécutivement la lettre $a$ est régulier : il s'agit de $L((a|b)*a^3(a|b)*)$

d) Avec $A = \{a, \dots, z, \bullet, @\}$, on note $N$ l'expression régulière $a|b|\dots|z$ :
le langage $L(N+(\varepsilon|\bullet N+)@N+\bullet(N^2|N^3))$ est l'ensemble des adresses mails sans chiffres de la forme *nom@serveur.ex* ou *prenom.nom@serveur.ex* ou *nom@serveur.ext* ou *prenom.nom@serveur.ext*

Dans ce genre de situation, on omet souvent les accolades pour les langages singletons, ainsi on note $a$ le langage singleton $\{a\}$.

De plus, l'expression régulière $\varnothing$ n'est jamais utilisée en dehors d'elle-même, car $\varnothing L = \varnothing$ et $\varnothing \cup L = L$

***Remarque.*** L'application $L$ précédente n'est pas injective : $L((a|b)*) = \{a, b\}^* = L((a*b*)*))$.

Deux expressions régulières qui dénotent le même langage sont dites équivalentes.

On imagine clairement qu'avec de telles notations, si on dispose de l'algorithme précédent, on peut alors vérifier qu'un mot a une certaine forme, qu'on appelle motif. De nombreux logiciels utilisent ces expressions régulières pour vérifier que la saisie d'un utilisateur est conforme à ce qu'attend la machine (numéro de téléphone, adresse mail, date, code postal, etc). Mais c'est aussi indispensable pour créer un langage informatique, qui suit des règles de syntaxe précise (l'expression régulière est alors singulièrement plus compliquée).

## 4.3 Expressions régulières étendues

Les expressions régulières précédentes permettent de décrire un certain nombre de langages : elles sont dites standard. Toutefois, elles sont parfois un peu lourdes pour préciser d'autres langages formés à partir d'intersection ou de complémentaire.

On ajoute alors aux symboles d'expressions régulières des notations permettant de nier ou d'intersecter des langages. Les notations n'étant pas fixées, on pourrait par exemple noter :

- $\bar{e}$ l'expression régulière telle que $L(\bar{e}) = A^* \setminus L(e)$
- $e \wedge f$ l'expression régulière telle que $L(e \wedge f) = L(e) \cap L(f)$.