1. Langages non réguliers

1. Pour démontrer qu'un langage n'est pas régulier, nous utilisons le lemme de gonflement. Par sa forme, ce lemme énonce une propriété des langages réguliers,

$$L \text{ régulier} \Rightarrow P(L)$$

où $P(L)$ est la propriété. Cette propriété est nécessaire et par les règles de la logique propositionnelle nous savons que $(B \Rightarrow A) \Leftrightarrow (\neg A \Rightarrow \neg B)$. Dès lors, la contraposée de la proposition ci-dessus est

$$\neg P(L) \Rightarrow \neg(L \text{ régulier})$$

La propriété est donc *nécessaire* dans le sens où si elle n'est pas satisfaite, le langage n'est pas régulier[1]. Dans notre cas, la proposition $P(L)$ est,

$$\exists n \in \mathbb{N} : \forall w \in L :$$
$$|w| \geq n \Rightarrow \big(\exists x, y, z : (w = xyz) \wedge (y \neq \epsilon) \wedge (|xy| \leq n) \wedge (\forall k \geq 0 : xy^k z \in L)\big)$$

Sa négation $\neg P(L)$ est donc[2],

$$\forall n \in \mathbb{N} : \exists w \in L :$$
$$(|w| \geq n) \wedge \big(\forall x, y, z : \neg\big((w = xyz) \wedge (y \neq \epsilon) \wedge (|xy| \leq n)\big) \vee \neg(\forall k \geq 0 : xy^k z \in L)\big)$$
$$\Leftrightarrow$$
$$\forall n \in \mathbb{N} : \exists w \in L :$$
$$(|w| \geq n) \wedge \Big(\forall x, y, z : (w = xyz) \wedge (y \neq \epsilon) \wedge (|xy| \leq n) \Rightarrow (\exists k \geq 0 : xy^k z \notin L)\Big)$$

Une stratégie pour démontrer $\neg P(L)$ est la suivante. Prendre un n quelconque (de façon à pouvoir généraliser sur tout $n \in \mathbb{N}$), construire un $w \in L$ particulier dont la taille est au moins n, votre construction doit donc dépendre de n. Prenez une décomposition de $w = xyz$ quelconque (pour pouvoir généraliser) mais qui satisfait $y \neq \epsilon$ et $|xy| \leq n$. Montrez qu'il existe un k tel que $xy^k z \notin L$.
Pour les trois langages ci-dessous nous appliquons cette stratégie.

(a) **Proposition 1.1** *Le langage $L = \{w \in \{0,1\}^* \mid \#0(w) = \#1(w)\}$ n'est pas régulier.* □

Preuve. Suivant la stratégie énoncée ci-dessus, nous prenons n quelconque et construisons la chaîne $w = 0^n 1^n \in L$.

Prenons une décomposition $w = xyz$ quelconque telle que $y \neq \epsilon$ et $|xy| \leq n$. Étant donné que $w = xyz = 0^n 1^n$, $xy \leq n$ et que et que $y \neq \epsilon$, nous savons que $xy = 0^i$ avec $\#0(y) \geq 1$

Prenons $k = 0$ et formons $xy^0 z = xz$, nous montrons que xz n'est pas dans L. Étant donné que $\#1(xy) = \#1(0^i) = 0$, nous avons $\#1(xz) = \#1(xyz) =$

[1] Par contre si elle est satisfaite pour un language L elle ne nous dit rien sur la régularité de celui-ci.
[2] Rappelez vous $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$ et la dualité des quantificateurs et connecteurs logiques.

$\#0(xyz) = m$. Par ailleurs, $\#0(xz) = \#0(xyz) - \#0(y) = m - \#0(y)$ et nous savons que $\#0(y) \geq 1$, cela signifie que $\#0(xz) = m - \#0(y) < m = \#1(xz)$, d'où l'on tire que $xz \notin L$.

L n'est donc pas régulier. ■

Prenez le temps de comparer ce langage avec celui, régulier, que nous vous avons proposé dans la série 5 à l'exercice 1.1(e).

Intuitivement, dans le langage L non régulier donné ici nous devons « compter » un nombre arbitrairement grand de zéros et de uns pour vérifier la propriété $\#0(w) = \#1(w)$ — pensez à la chaîne 0^n1^n. Ces nombres étant arbitrairement grands il n'est pas possible de « mémoriser » leur valeur dans un nombre *fini* d'états.

Ceci n'est pas le cas pour le langage régulier de la série 5 car la propriété additionnelle sur les préfixes satisfaite par les chaînes du langage permet de vérifier $\#0(w) = \#1(w)$ en vérifiant l'égalité sur les sous chaînes de longueur — fixe et finie — 2.

(b) **Proposition 1.2** *Le langage* $L = \{w \in \{a, b\}^* \mid w$ *palindrome*$\}$ *n'est pas régulier.* □

Preuve. Suivant la stratégie énoncée ci-dessus, nous prenons n quelconque et construisons la chaîne $w = b^naab^n \in L$.

Prenons une décomposition $w = xyz$ quelconque telle que $y \neq \epsilon$ et $|xy| \leq n$. Nous savons donc que $y = b^j$ avec $j > 0$.

Prenons $k = 0$ et formons $xy^0z = xz$, nous montrons que xz n'est pas dans L. Nous avons $xz = b^{n-j}aab^n = l_1 \cdot \ldots \cdot l_{2n+2-j}$ avec $l_i \in \{a, b\}$. $l_{n-j+1} = a$ mais $l_{2n+2-j-(n-j+1-1)} = l_{n+2} = b$, ceci car $j > 0$. La propriété des palindromes n'est donc pas satisfaite par xz.

L n'est donc pas régulier. ■

(c) **Proposition 1.3** *Le langage* $L = \{w \in \{(,), a\}^* \mid w$ *équilibré*$\}$ *n'est pas régulier.* □

Preuve. Suivant la stratégie énoncée ci-dessus, nous prenons n quelconque et construisons la chaîne $w = (^n)^n \in L$.

Prenons une décomposition $w = xyz$ quelconque telle que $y \neq \epsilon$ et $|xy| \leq n$. Étant donné que $w = xyz = (^n)^n$, $xy \leq n$ et que et que $y \neq \epsilon$, nous savons que $xy = (^i$ avec $\#((y) \geq 1$

Prenons $k = 0$ et formons $xy^0z = xz$, nous montrons que xz n'est pas dans L. Étant donné que $\#)(xy) = \#)((^i) = 0$, nous avons $\#)(xz) = \#)(xyz) = \#((xyz) = m$. Par ailleurs, $\#((xz) = \#((xyz) - \#((y) = m - \#((y)$ et nous savons que $\#((y) \geq 1$, cela signifie que $\#((xz) = m - \#((y) < m = \#)(xz)$, d'où l'on tire que $xz \notin L$.

L n'est donc pas régulier. ■

2. Dans les programmes JAVA, les instructions sont structurées par blocs. Un bloc est délimité par des accolades, le caractère '{' ouvre un bloc et '}' le referme. Or les blocs peuvent être imbriqués les uns dans les autres, et les accolades doivent obligatoirement être équilibrées. Or nous avons vu dans l'exercice 1.1.(c) que ce genre de langage n'est pas régulier.

   Notons que JAVA utilise aussi des parenthèses pour délimiter les expressions ou pour les appels de méthode et encore des crochets pour les tableaux. Ces parenthèses et crochets doivent également être équilibrées, ce qui ne fait que renforcer notre soupçon.

   L'ensemble des programmes JAVA n'étant en fait pas régulier, nous ne pouvons pas reconnaître les programmes JAVA syntaxiquement correct à l'aide d'un automate fini, ni décrire leur syntaxe à l'aide d'une expression régulière.

2. Langages réguliers infinis

L'algorithme suivant permet de décider la finitude d'un langage régulier L sur un alphabet Σ. L'algorithme *accepte* un langage L si et seulement si il est fini.

1. Trouver un nombre n satisfaisant la condition du lemme de gonflement pour L (le plus petit par exemple).
2. Si pour tout $w \in \{w \in \Sigma^* \mid n \leq |w| \leq 2n-1\}$, $w \notin L$ alors L est accepté.
   Sinon L est rejeté.

Pour le point 1, L étant régulier, l'existence de n est garantie par le lemme de gonflement. Afin de prouver la validité de notre algorithme, il nous reste donc à démontrer la proposition suivante qui est implicitement contenue dans le point 2 de l'algorithme[3] :

Proposition 2.4 *Soit L un langage régulier et n un nombre satisfaisant la condition du lemme de gonflement pour L. Alors nous avons*

$$L \text{ infini} \Leftrightarrow \exists w \in L : n \leq |w| \leq 2n-1$$

□

Preuve. Nous démontrons d'abord l'implication de droite à gauche, puis de gauche à droite.

1. ($\Leftarrow$)
   Supposons qu'il existe $w \in L$ tel que $n \leq |w| \leq 2n-1$. L étant régulier nous savons, par le lemme de gonflement, que nous pouvons décomposer w en xyz tel que $y \neq \epsilon$, $|xy| \leq n$ et — ce qui nous intéresse vraiment — $\forall k \geq 0 : xy^kz \in L$. De cette dernière propriété il est possible de déduire que $\forall m \in \mathbb{N} : \|L\| \neq m$, ce qui signifie que L est infini.
2. ($\Rightarrow$)
   Nous démontrons la contraposée. Supposons qu'il n'existe pas de $w \in L$ tel que $n \leq |w| \leq 2n-1$, nous devons démontrer que L est fini. Pour cela nous prouvons par l'absurde qu'il n'existe pas de chaîne $w \in L$ telle que $|w| \geq 2n$.
   Supposons qu'il existe des chaînes $w \in L$ telles que $|w| \geq 2n$. Soit S l'ensemble de ces chaînes et w l'un des plus petits éléments de S, dans le sens où $\forall w' \in S : |w| \leq |w'|$. L étant régulier nous savons, par le lemme de gonflement, que w peut être décomposée en $w = xyz$ avec $y \neq \epsilon$ et $|xy| \leq n$ et $xy^0z = xz \in L$. Étant donné que $y \neq \epsilon$, xz est strictement plus petite que w et w étant une des plus petite chaîne dans L dont la taille est plus grande que $2n$, nous savons donc que $xz < 2n$. Il existe donc une chaîne $w' = xz \in L$ telle que $n \leq |w'| \leq 2n-1$, ce qui contredit l'hypothèse initiale de la contraposée. Il n'existe donc pas de chaîne $w \in L$ telle que $|w| \geq 2n$. Il existe donc un $m < 2n$ tel que $\|L\| = m$, ce qui signifie que L est fini.

De (1) et (2) nous déduisons la proposition. ■

3. Langages finis et langages réguliers

Théorème 3.5 *Tous les langages finis sont réguliers.* □

Preuve.
Soit L un langage fini quelconque. Nous avons les deux cas suivant,

[3] Cette proposition est en fait la contraposée du point 2. Nous passons d'une proposition constructive, algorithmique — un « pour tout » sur un ensemble fini facilement constructible — à une proposition existentielle sur un ensemble potentiellement infini.

1. $L = \emptyset$.

   Prenons l'expression régulière $\alpha = \boldsymbol{\emptyset}$, nous avons $L(\alpha) = \emptyset = L$. Il existe donc une expression régulière α telle que $L(\alpha) = L$. L est donc régulier (cf. théorème leçon 5, p.21).

2. $L \neq \emptyset$

   L étant fini, nous pouvons énumérer ses chaînes. Supposons que $L = \{w_1, w_2, \ldots, w_n\}$ avec $n = \|L\|$. Construisons, l'expression régulière suivante,

   $$w_1 + w_2 + \ldots + w_n$$

   Nous avons $L(w_1 + w_2 + \ldots + w_n) = \bigcup_{i=1}^{n} L(w_i) = \bigcup_{i=1}^{n} \{w_i\} = L$. Il existe donc une expression régulière α telle que $L(\alpha) = L$. L est donc régulier (cf. théorème leçon 5, p.21).

De (1) et (2) et en généralisant sur L, nous déduisons le théorème. ■