Chapitre 6

Langages

Introduction

Un programme ne manipule des objets abstraits que par le biais d'une représentation de ces objets. Les entrées d'un problème d'informatique sont donc les codages d'objets abstraits plutôt que ces objets eux-mêmes. Et un codage n'est rien d'autre qu'un **mot** sur un **alphabet** donné. En somme, pour un programme, une donnée d'un problème est toujours un mot.

Résoudre le problème consiste à décider si cette donnée appartient à l'ensemble des mots qui satisfont une certaine propriété.

Or un ensemble de mots, c'est exactement ce que nous appellerons bientôt un **langage**.

Ainsi, pour un informaticien, résoudre un problème, c'est reconnaitre un langage, c'est-à-dire identifier les mots qui le constituent.

Un problème de décision sera donc toujours associé à un langage L et s'exprimera sous la forme :

Un mot m étant donné, m appartient-il au langage L ?

On introduit dans ce chapitre les définitions et propriétés de base sur les langages. On s'intéresse plus particulièrement aux langages rationnels. Ces derniers vont être définis de deux manières :

- à l'aide d'expressions rationnelles ;
- en utilisant des automates finis (chapitre 7).

1 Généralités sur les langages

1.1 Mot sur un alphabet

Définition.

- Nous appelons **alphabet** tout ensemble Σ de cardinal fini non nul.
- Les éléments de Σ sont appelés des **lettres**.

Exemples.

- $\Sigma = \{0, 1\}$ l'alphabet binaire.
- $\Sigma = \{A, C, T, G\}$ l'alphabet "de la vie" des molécules d'ADN.
- $\Sigma = \{\heartsuit, \diamondsuit, \clubsuit, \spadesuit\}$ l'alphabet des familles de cartes.

Définition.

Soit Σ un alphabet.
Nous appelons **mot** sur Σ toute juxtaposition d'un nombre fini de lettres de Σ,

- dans le cas d'un nombre non nul de lettres, notée :
$$u = a_1 a_2 \dots a_{p-1} a_p \,,$$
où p appartient à $\mathbb{N}^*$ et, pour tout $i \in [\![1, p]\!]$, a_i appartient à Σ,
- dans le cas d'un nombre nul de lettres, notée ε et appelée le **mot vide**.

Notations. L'ensemble des mots sur Σ se note Σ^* et l'ensemble des mots sur Σ distincts du mot vide se note Σ^+. On a donc :
$$\Sigma^* = \Sigma^+ \cup \{\varepsilon\} \,.$$

Définition.

Soit Σ un alphabet.
La **longueur d'un mot** u appartenant à Σ^* est l'entier, noté $|u|$, défini par :

- $|u| = 0$ si $u = \varepsilon$;
- $|u| = p$ si $u = a_1 a_2 \dots a_{p-1} a_p$.

Définition.

Soit Σ un alphabet et u et v deux mots appartenant à Σ^*.
Nous définissons le mot obtenu par **concaténation** de u et v, noté $u.v$, par :

- dans le cas où u et v sont le mot vide : $u.v = \varepsilon$.
- dans le cas où u est le mot vide et v différent du mot vide : $u.v = v$.
- dans le cas où u est différent du mot vide et v est le mot vide : $u.v = u$.
- dans le cas où $u = a_1 a_2 \dots a_{p-1} a_p$ et $v = b_1 b_2 \dots b_{q-1} b_q$ sont différents du mot vide,
$$u.v = a_1 a_2 \dots a_{p-1} a_p b_1 b_2 \dots b_{q-1} b_q.$$

Propriété 1 (de la concaténation)

Soit Σ un alphabet.

(1) En désignant par . la concaténation sur Σ^*, le couple $(\Sigma^*, .)$ est un monoïde, c'est-à-dire que la loi . est une LCI dans Σ^*, associative dans Σ^* et admettant le mot vide comme élément neutre dans Σ^*.

(2) L'application $\Sigma^* \to \mathbb{N}\,,\ u \mapsto |u|$ est un morphisme du monoïde $(\Sigma^*, .)$ dans le monoïde $(\mathbb{N}, +)$, c'est-à-dire :

$$\forall\,(u, v) \in (\Sigma^*)^2, \quad |u.v| = |u| + |v|.$$

Définition.

Soit Σ un alphabet et u un mot sur Σ. Pour tout entier naturel n, on définit u^n par :

- $u^0 = \varepsilon$,
- $u^{n+1} = u.u^n$ pour $n \in \mathbb{N}$.

Exemple. Le mot $a^4.b^2.c^2$ sur l'alphabet $\Sigma = \{a, b, c\}$ est le mot *aaaabbcc* de longueur 8.

Définition.

Soit Σ un alphabet et u appartenant à Σ^*.

- On dit que x est un **préfixe** de u s'il existe un mot $y \in \Sigma^*$ tel que $u = x.y$.
  Dans le cas où y est différent du mot vide, nous parlons de **préfixe propre** ou **préfixe strict**.
- On dit que y est un **suffixe** de u s'il existe un mot $x \in \Sigma^*$ tel que $u = x.y$.
  Dans le cas où x est différent du mot vide, nous parlons de **suffixe propre** ou **suffixe strict**.
- On dit que y est un **facteur** de u s'il existe deux mots $x, z \in \Sigma^*$ tels que $u = x.y.z$.
  Dans le cas où x ou z est différent du mot vide, nous parlons de **facteur propre** ou **facteur strict**.

Remarque. Tout mot est à la fois préfixe, suffixe et facteur de lui-même car :

$$u = \varepsilon.u = u.\varepsilon = \varepsilon.u.\varepsilon.$$

Exemple. Soit $\Sigma = \{a, b\}$. Donner l'ensemble des préfixes, des suffixes et des facteurs du mot *abba* :

Définition.

Soit Σ un alphabet et $u = a_1a_2 \dots a_{p-1}a_p$ un mot non vide sur Σ.
On dit que v est un **sous-mot** de u s'il existe des indices $1 \leq i_1 < i_2 < \dots < i_k \leq p$ tels que :

$$v = a_{i_1}a_{i_2} \dots a_{i_k}.$$

Le mot vide peut être considéré comme un sous-mot de u.

Remarques.

1. De façon informelle, un sous-mot d'un mot u est le mot u dans lequel on a effacé certaines lettres. Par exemple, aa, $abab$ et cac sont des sous-mots du mot $abcabc$.
2. Nous pouvons également définir les sous-mots du mot vide : ε est le seul sous-mot du mot vide.

Attention.

Ne pas confondre facteur et sous-mot : un facteur est un sous-mot mais la réciproque n'est pas vraie !

1.2 Premières propriétés

Propriété 2 (Égalité de deux mots)

Soit Σ un alphabet, $u = a_1a_2 \dots a_p$ et $v = b_1b_2 \dots b_q$ deux mots non vides sur Σ. Alors :

$$u = v \quad \Leftrightarrow \quad \begin{cases} p = q, \\ \forall i \in [\![1, p]\!],\ a_i = b_i. \end{cases}$$

Autrement dit, deux mots sont **égaux** si et seulement s'ils sont de **même longueur** et constitués des **mêmes lettres** dans le **même ordre**.

Propriété 3 (Lemme de Levi)

Soit Σ un alphabet, x, y, u et v appartenant à Σ^* tels que : $x.y = u.v$. Alors :

- dans le cas où $|x| \geq |u|$, il existe un unique mot w tel que :

$$x = uw \quad \text{et} \quad v = wy.$$

- dans le cas où $|x| < |u|$, il existe un unique mot w tel que :

$$u = xw \quad \text{et} \quad y = wv.$$

Preuve.

$\square$

Propriété 4 (Corollaire du lemme de Levi)

Soit Σ un alphabet et u un mot sur Σ.

(1) Si x et y sont deux préfixes de u, alors x est un préfixe de y ou y est un préfixe de x.

(2) Si x et y sont deux suffixes de u, alors x est un suffixe de y ou y est un suffixe de x.

Preuve.

$\square$

1.3 Langages

Définition.

Nous appelons **langage** sur un alphabet Σ toute partie L de Σ^*.

Exemples.

1. Soit Σ un alphabet. L'ensemble des mots sur Σ de longueur 2 est un langage sur Σ.
2. Soit l'alphabet $\Sigma = [\![0, 9]\!]$. L'ensemble des entiers naturels multiples de 3 est un langage sur Σ.
3. Soit l'alphabet $\Sigma = \{a, b\}$. L'ensemble des mots sur Σ tels qu'aucun a ne se situe après un b est un langage sur Σ et :

$$L =$$

Cas particuliers. Soit Σ un alphabet.

- $\emptyset$ est un langage sur Σ appelé le **langage vide**, à ne pas confondre avec le langage $\{\varepsilon\}$.
- Σ^* est un langage sur Σ appelé le **langage plein**.

Les langages étant des ensembles, ils sont susceptibles d'être soumis aux opérations ensemblistes.

Définition.

Soit Σ un alphabet et L_1 et L_2 deux langages sur Σ. Nous définissons :

- le **complémentaire** de L_1 :
$$\overline{L_1} = \{u \mid u \in \Sigma^* \setminus L_1\}.$$
- l'**union** de L_1 et L_2 :
$$L_1 \cup L_2 = \{u \mid u \in L_1 \text{ ou } u \in L_2\}.$$
- l'**intersection** de L_1 et L_2 :
$$L_1 \cap L_2 = \{u \mid u \in L_1 \text{ et } u \in L_2\}.$$

Remarques.

1. Même si L_1 et L_2 sont définis sur des alphabets différents Σ_1 et Σ_2, il est possible de les considérer comme deux langages sur le même alphabet $\Sigma = \Sigma_1 \cup \Sigma_2$. Dans la suite, quand nous ferons l'union et l'intersection de deux langages, on supposera sans perte de généralité qu'ils sont définis sur le même alphabet.
2. A partir du complémentaire, de l'union et de l'intersection, on peut aussi définir :
   - la **différence** de L_1 par L_2 :
   $$L_1 \setminus L_2 = L_1 \cap \overline{L_2}.$$
   - la **différence symétrique** de L_1 et L_2 (correspondant au ou exclusif) :
   $$L_1 \Delta L_2 = (L_1 \cup L_2) \setminus (L_1 \cap L_2).$$

Il est aussi possible de faire la concaténation de deux langages :

Définition.

Soit Σ un alphabet, L_1 et L_2 deux langages sur Σ.
Le **langage concaténation** ou **produit** de L_1 et L_2 est l'ensemble $L_1.L_2$ défini par :

$$L_1.L_2 = \{w \mid \exists u \in L_1,\ \exists v \in L_2,\ w = u.v\}.$$

Exemple. Si $L_1 = \{aa, ab, bc\}$ et $L_2 = \{\varepsilon, a, abc\}$, alors :

$$L_1.L_2 =$$

Remarque. Soit Σ un alphabet, L un langage sur Σ. Alors :

$$\{\varepsilon\}.L = L.\{\varepsilon\} = \qquad\qquad \emptyset.L = L.\emptyset =$$

Propriété 5 (Opérations sur les langages)

Soit Σ un alphabet et L_1, L_2 et L_3 trois langages sur Σ. Alors :

$$(L_1 \cup L_2).L_3 = (L_1.L_3) \cup (L_2.L_3).$$

Preuve.

□

Remarque. On a également :

$$(L_1 \cap L_2).L_3 \subset (L_1.L_3) \cap (L_2.L_3).$$

En effet,

Par contre, l'inclusion réciproque est fausse en général. Considérons l'alphabet $\Sigma = \{a, b\}$ et les langages $L_1 = \{\varepsilon, a\}$, $L_2 = \{b\}$ et $L_3 = \{\varepsilon, b\}$ sur Σ. Alors :

- $(L_1 \cap L_2).L_3 =$
- $(L_1.L_3) \cap (L_2.L_3) =$

Comme dans le cas des mots, la concaténation peut être itérée pour produire la puissance d'un langage :

Définition.

Soit Σ un alphabet et L un langage sur Σ. Pour tout entier naturel n, on définit L^n par :

- $L^0 = \{\varepsilon\}$,
- $L^{n+1} = L.L^n$ pour $n \in \mathbb{N}$.

Exemple. Soit $L = \{a, b, ac\}$ un langage sur $\Sigma = \{a, b, c\}$. Le langage L^3 est :

Attention.

Ne pas confondre L^n et le langage plus petit $\{u^n \mid u \in L\}$. Par exemple, pour $L = \{ab, ba\}$,

$$\begin{array}{rcl} L^2 = \{u \mid u \in L\}^2 & = & \{abab, abba, baba, baab\}, \\ \{u^2 \mid u \in L\} & = & \{abab, baba\}. \end{array}$$

Remarque. Soit Σ un alphabet, L un langages sur Σ et $p, q \in \mathbb{N}$. Alors :

$$L^p.L^q = L^{p+q}.$$

Définition.

Soit Σ un alphabet et L un langages sur Σ.

- L'**étoile** (de Kleene) de L est le langage noté L^* défini par :

$$L^* = \bigcup_{n\in\mathbb{N}} L^n.$$

- L'**étoile strict** de L est le langage noté L^+ défini par :

$$L^+ = \bigcup_{n\in\mathbb{N}^*} L^n.$$

Remarque. On a :

$$L^* = \{\varepsilon\} \cup L^+ \quad \text{et} \quad L.L^* = L^+.$$

De plus,

$$L^+ = \begin{cases} L^* - \{\varepsilon\} & \text{si } \varepsilon \notin L, \\ L^* & \text{si } \varepsilon \in L. \end{cases}$$

Exemple. Soit $\Sigma = \{a, b\}$ un alphabet.

- $\emptyset^* =$
- $\{a\}^* =$
- $\{a\}^*.\{b\}^* =$
- $\{a, b\}^* =$
- Désignons par L_1 le langage sur Σ constitué des mots de longueur paire :

$$L_1 =$$

- Désignons par L_2 le langage sur Σ constitué des mots contenant un nombre pair de a.

$$L_2 =$$

- Désignons par L_3 le langage sur Σ constitué des mots de longueur non divisible par 3 :

$$L_3 =$$

Propriété 6 (Croissance de l'étoile de Kleene)

Soit Σ un alphabet et L et M deux langages sur Σ. Alors :

$$L \subset M \quad \Rightarrow \quad L^* \subset M^*.$$

Preuve.

$\square$

Propriété 7 (de l'étoile de Kleene d'un langage)

Soit Σ un alphabet et L un langage sur Σ.

(1) L^* est le plus petit langage sur Σ contenant L et le mot vide ε, stable pour la concaténation.

(2) $(L^*)^* = L^*$.

Preuve.

$\square$

Exercice 1 (★)
Déterminer les mots sur $\Sigma = \{a, b\}$ dont tous les facteurs sont des préfixes.

Exercice 2 (★)
Montrer que deux mots qui ont le même ensemble de préfixes sont égaux.
Est-ce encore vrai si l'on considère les ensembles de préfixes propres ?

Exercice 3 (★)
Soit $a, b \in \Sigma$ et $u \in \Sigma^*$ tels que $au = ub$.
Montrer que $a = b$ et que $u \in \{a\}^*$.

Exercice 4 (★)
Soit L un langage non vide sur un alphabet Σ. Montrer que :

$$\varepsilon \in L \quad \Leftrightarrow \quad L \subset L^2.$$

Exercice 5 (★)
Notons, pour tout langage L, $pref(L)$ le langage des préfixes des mots de L.
Déterminer, pour tous langages L_1 et L_2, le langage $pref(L_1.L_2)$.

Exercice 6 (★★★)
Soit u et v deux mots non vides sur un alphabet Σ.
Montrer qu'il existe deux mots x et y tels que $u = xy$ et $v = yx$ si et seulement s'il existe un mot w tel que $uw = wv$.

Exercice 7 (★★)

Soit Σ un alphabet et L et M deux langages sur Σ.

1. (a) Montrer que $(L^* \cup M^*) \subset (L \cup M)^*$.
   (b) L'inclusion réciproque est-elle vraie ?
2. (a) Montrer que $(L^*.M^*) \subset (L \cup M)^*$.
   (b) L'inclusion réciproque est-elle vraie ?
3. Montrer que $(A \cup B)^* = (A^*.B^*)^*$.

Exercice 8 (Lemme d'Arden - ★★)

1. Soit $\Sigma = \{a, b\}$. On considère l'équation $L = (\{a\}.L) \cup \{b\}$ d'inconnue le langage $L \subset \Sigma^*$.
   (a) Montrer que $L = \{a\}^*.\{b\}$ est solution de cette équation.
   (b) Montrer qu'il s'agit de l'unique solution.
2. Soit Σ un alphabet quelconque et A et B deux langages sur Σ. On considère l'équation $L = (A.L) \cup B$ d'inconnue le langage $L \subset \Sigma^*$.
   (a) Montrer que $L = A^*B$ est solution de cette équation.
   (b) Montrer que si de plus $\varepsilon \notin A$, cette solution est unique.

Exercice 9 (Racine carrée d'un langage - ★★)

Soit Σ un alphabet et L un langage sur Σ.
On appelle racine carrée de L le langage, noté $\sqrt{L}$, défini par :

$$\sqrt{L} = \{u \in \Sigma^* \mid u.u \in L\}.$$

1. Considérons l'alphabet $\Sigma = \{a, b\}$. Déterminer $\sqrt{L}$ lorsque :
   (a) $L = \{a\}.\{b\}^*$,
   (b) $L = \{a\}.\{b\}^*.\{a\}$,
   (c) $L = \{a.b\}^*$.
2. (a) Montrer que $L \subset \sqrt{L^2}$.
   (b) Donner un contre-exemple pour l'inclusion réciproque.

Exercice 10 (Code - ★★★)

Un code sur un alphabet Σ est un langage L tel que l'égalité $x_1x_2\dots x_p = y_1y_2\dots y_q$ avec $x_1, x_2, \dots, x_p, y_1, y_2, \dots, y_q \in L$ entraîne $p = q$ et $x_i = y_i$ pour tout $i \in [\![1, p]\!]$.

1. Parmi les langages finis suivants, déterminer lesquels sont des codes.
   - $L_1 = \{ab, baa, abba, aabaa\}$
   - $L_2 = \{b, ab, baa, abaa, aaaa\}$
   - $L_3 = \{aa, ab, aab, bba\}$
   - $L_4 = \{a, ba, bba, baab\}$
2. Soit $u \in \Sigma^*$. Montrer que $\{u\}$ est un code si et seulement si $u \neq \varepsilon$.
3. Soit u et $v \in \Sigma^*$ distincts. Montrer que $\{u, v\}$ est un code si et seulement si $uv \neq vu$.
4. Soit $L \subset \Sigma^*$ un langage ne contenant pas ε et tel qu'aucun mot de L ne soit préfixe d'un autre mot de L. Montrer que L est un code.

---

2 Expressions rationnelles et langages rationnels

2.1 Définitions

Définition.

Soit Σ un alphabet. On définit inductivement l'ensemble des **langages rationnels** (ou **réguliers**) sur Σ, noté $\text{Rat}(\Sigma)$, par :

- $\emptyset \in \text{Rat}(\Sigma)$;
- pour tout $a \in \Sigma$, $\{a\} \in \text{Rat}(\Sigma)$;
- si $L, L' \in \text{Rat}(\Sigma)$, alors $L \cup L' \in \text{Rat}(\Sigma)$;
- si $L, L' \in \text{Rat}(\Sigma)$, alors $L.L' \in \text{Rat}(\Sigma)$;
- si $L \in \text{Rat}(\Sigma)$, alors $L^* \in \text{Rat}(\Sigma)$.

Exemples. Les langages suivants sont rationnels :

- $L_1 = \{\varepsilon\}$, car
- $L_2 = \Sigma$, car
- L_3 l'ensemble des mots de longueur impaire, car
- L_4 l'ensemble des mots commençant par un a et terminant par un b, car

Définition.

Soit Σ un alphabet.
Les **expressions rationnelles** (ou **régulières**) sont définies inductivement de la manière suivante :

- Cas de base :
  - $\emptyset$ et ε sont des expressions rationnelles,
  - pour tout a appartenant à Σ, a est une expression rationnelle.
- Étape d'induction :
  - pour toutes expressions rationnelles e_1 et e_2, la réunion $(e_1 + e_2)$ de e_1 et e_2 est une expression rationnelle,
  - pour toutes expressions rationnelles e_1 et e_2, la concaténation $(e_1.e_2)$ de e_1 et e_2 est une expression rationnelle,
  - pour toute expression rationnelle e, l'itération (aussi appelée l'étoile) (e^*) de e est une expression rationnelle.

Exemple. Si $\Sigma = \{a, b\}$, alors $(a.(b^*)) + b$ est une expression rationnelle.

Notation. Nous pouvons nous affranchir de certaines parenthèses en respectant un ordre de priorité : $*$ puis . puis $+$. Ainsi, $(a.(b^*)) + b$ se note plus simplement $a.b^* + b$.

Définition.

On définit l'opération d'**interprétation** qui, à une expression rationnelle e sur un alphabet Σ, associe un langage $\mathscr{L}(e)$ dit **dénoté** par e sur Σ de manière inductive avec les règles naturelles suivantes :

- Cas de base :
  - $\emptyset$ dénote le langage $\emptyset$: $\mathscr{L}(\emptyset) = \emptyset$,
  - ε dénote le langage $\{\varepsilon\}$: $\mathscr{L}(\varepsilon) = \{\varepsilon\}$,
  - pour tout a appartenant à Σ, a dénote le langage $\{a\}$: $\mathscr{L}(a) = \{a\}$.
- Étape d'induction :
  - pour toutes expressions rationnelles e_1 et e_2, $e_1 + e_2$ dénote la réunion du langage dénoté par e_1 et du langage dénoté par e_2 :
  $$\mathscr{L}(e_1 + e_2) = \mathscr{L}(e_1) \cup \mathscr{L}(e_2),$$
  - pour toutes expressions rationnelles e_1 et e_2, $(e_1.e_2)$ dénote la concaténation du langage dénoté par e_1 et du langage dénoté par e_2 :
  $$\mathscr{L}(e_1.e_2) = \mathscr{L}(e_1).\mathscr{L}(e_2),$$
  - pour toute expression rationnelle e, (e^*) dénote l'étoile de Kleene du langage dénoté par e :
  $$\mathscr{L}(e^*) = \mathscr{L}(e)^*.$$

Exemple. Considérons l'alphabet $\Sigma = \{a, b\}$ et le langage L sur Σ défini par :

$$L = \{a \,.\, b^k \,.\, a \mid k \in \mathbb{N}\}.$$

Alors $L = \mathscr{L}(a.b^*.a)$.

De par la construction des langages rationnels, on en déduit directement la proposition :

Propriété 8 (Langages rationnels et expressions rationnelles)

Un langage $L \subset \Sigma^*$ est **rationnel** si et seulement s'il est l'interprétation d'une **expression rationnelle** sur l'alphabet Σ (c'est-à-dire s'il est dénoté par une expression rationnelle).

Remarque. Lorsqu'il n'y a pas d'ambiguïté, on assimile une expression rationnelle avec son langage interprété.

Exemples. Considérons l'alphabet $\Sigma = \{a, b\}$.

1. Le langage L_1 des mots sur Σ débutant par un certain nombre de a suivi par un certain nombre de b est-il rationnel ?

2. Décrire le langage L_2 dénoté par l'expression rationnelle $(a.b)^*$.

3. Le langage L_3 des mots sur Σ dont aaa est un facteur est-il rationnel ?

4. Le langage L_4 des mots sur Σ dans lesquels chaque b est suivi d'un a est-il rationnel ?

5. Le langage L_5 formé du mot vide et de tous les mots sur Σ se terminant par un b est-il rationnel ?

2.2 Arbre associé à une expression rationnelle

Nous pouvons naturellement associer à une expression rationnelle, un arbre binaire où les feuilles sont des éléments de $\Sigma \cup \{\emptyset, \varepsilon\}$ et les noeuds internes sont étiquetés par les opérations

$$+ \text{ (réunion)}, \quad . \text{ (concaténation)} \quad \text{et} \quad * \text{ (étoile de Kleene)}.$$

Par exemple, la représentation arborescente de l'expression $(a+b)^*.\, c$ est :

Aussi, des propriétés relatives aux expressions rationnelles pourront se prouver par induction sur la hauteur de l'arbre, appelée **profondeur de l'expression rationnelle**.

2.3 Expressions rationnelles équivalentes

Attention.

Une expression rationnelle dénote un unique langage, mais un langage donné peut être dénoté par des expressions rationnelles différentes.

Par exemple, les expressions rationnelles $a+bb^*a$ et b^*a s'interprètent toutes les deux en le langage $\{b\}^*\{a\}$.

Définition.

Deux expressions rationnelles e_1 et e_2 sont dites **équivalentes** si et seulement si elles dénotent le même langage. On note alors $e_1 \equiv e_2$.

Exemple. On a donc $a + bb^*a \equiv b^*a$.

Propriété 9 (Expressions rationnelles équivalentes)

Soient e_1, e_2, e_3 trois expressions rationnelles sur un alphabet Σ. On a :

(1) $\emptyset + e_1 \equiv e_1 + \emptyset \equiv e_1$	(2) $e_1.\varepsilon \equiv \varepsilon.e_1 \equiv e_1$
(3) $e_1.\emptyset \equiv \emptyset.e_1 \equiv \emptyset$	(4) $e_1 + e_2 \equiv e_2 + e_1$
(5) $(e_1.e_2).e_3 \equiv e_1.(e_2.e_3)$	(6) $(e_1 + e_2) + e_3 \equiv e_1 + (e_2 + e_3)$
(7) $e_1.(e_2 + e_3) \equiv e_1.e_2 + e_1.e_3$	(8) $(e_1 + e_2).e_3 \equiv e_1.e_3 + e_2.e_3$
(9) $e_1 + e_1 \equiv e_1$	(10) $\emptyset^* \equiv \varepsilon$
(11) $e_1.e_1^* \equiv e_1^*.e_1$	(12) $e_1.e_1^* + \varepsilon \equiv e_1^*$

Remarque. Déterminer si deux expressions rationnelles sont équivalentes, chercher l'expression rationnelle la plus courte dénotant un langage donné, sont des problèmes difficiles à résoudre algorithmiquement.
Nous aborderons ce problème par le biais des **automates finis**.

Propriété 10

Pour toute expression rationnelle e dénotant un langage non vide, il existe une expression rationnelle e' équivalente à e ne contenant pas le symbole $\emptyset$.

Preuve.

$\square$

Propriété 11

Pour toute expression rationnelle e dénotant un langage non vide, il existe une expression rationnelle e' ne contenant ni le symbole $\emptyset$, ni le symbole ε, telle que e soit équivalente à ε, e' ou $\varepsilon + e'$.

Preuve.

$\square$

Exercice 11 (★)

Donner une expression rationnelle correspondant à chacun des langages suivants, décrits en français.

1. Les mots sur $\Sigma = \{a, b\}$ contenant exactement un a.
2. Les mots sur $\Sigma = \{a, b\}$ contenant toujours un b directement après un a.
3. Les mots sur $\Sigma = \{a, b, c\}$ ne contenant pas de a après le premier b.
4. Les mots sur $\Sigma = \{a, b, c\}$ où chaque paire de a est séparée par exactement 3 caractères.
5. Les mots sur $\Sigma = \{a, b, c\}$ comportant exactement deux b, où tout a est suivi d'au moins deux c, et qui se terminent par b.
6. Les mots sur $\Sigma = \{a, b, c, d\}$ non vide, commençant par b et terminant par d.

Exercice 12 (★)

Le langage $L = \{a^n b^m \mid (n + m) \equiv 0 \bmod 2\}$ est-il rationnel ?

Exercice 13 (★★)

Montrer que tout langage fini est rationnel.

Exercice 14 (★★)

Soit l'alphabet $\Sigma = \{0, 1\}$. Un mot binaire est un mot construit sur Σ, et un mot binaire normalisé est un mot binaire qui soit commence par un 1, soit est exactement le mot 0.

La valeur d'un mot binaire $u = u_{n-1} \ldots u_0$ est définie par $V(u) = \sum_{i=0}^{n-1} 2^i u_i$.

Pour chacun des cas, dire si le langage défini est rationnel, par exemple en exhibant une expression rationnelle qui le dénote :

1. le langage L_1 des mots binaires normalisés ;
2. le langage L_2 des mots binaires dont la valeur est paire ;
3. le langage L_3 des mots binaires normalisés dont la valeur est paire ;
4. le langage L_4 des mots binaires dont la valeur est une puissance de 2.

3 Langages locaux

3.1 Définition et exemples

Remarque. A première vue, l'intérêt de cette notion n'est pas évident, mais nous verrons plus tard que ces langages sont particulièrement bien adaptés à la reconnaissance efficace d'une expression rationnelle par un automate.

Notations. Considérons un langage L sur un alphabet Σ et posons :

- $P(L)$ l'ensemble des premières lettres des mots de L :
$$P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\},$$
- $S(L)$ l'ensemble des dernières lettres des mots de L :
$$S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\},$$
- $F(L)$ l'ensemble des facteurs de longueur 2 des mots de L :
$$F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\},$$
- $N(L) = \Sigma^2 \setminus F(L)$.

Remarque. De manière évidente nous avons l'inclusion :

$$L \setminus \{\varepsilon\} \subset (P(L)\Sigma^* \cap \Sigma^* S(L)) \setminus (\Sigma^* N(L) \Sigma^*)$$

ce qui signifie simplement que tout mot non vide de L a sa première lettre dans $P(L)$, sa dernière lettre dans $S(L)$ et qu'aucun de ses facteurs de longueur 2 n'est dans $N(L)$.

Nous dirons qu'un langage L sur l'alphabet Σ est local lorsque cette inclusion est en fait une égalité :

Définition.

Un langage L est dit **local** si et seulement s'il existe deux parties P et S de Σ et une partie N de Σ^2 (l'ensemble des mots de longueur 2) telles que :

$$L \setminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \setminus (\Sigma^* N \Sigma^*)\,.$$

Remarques.

1. Un langage local est donc défini par l'ensemble P de ses premières lettres, l'ensemble S de ses dernières lettres et l'ensemble N des facteurs de longueur 2 que l'on ne peut trouver dans aucun des mots du langage.
2. Si de telles parties existent, nécessairement : $P = P(L)$, $S = S(L)$ et $N = N(L)$.
3. On préférera parfois donner l'ensemble F des facteurs de longueur 2 des mots du langage plutôt que l'ensemble N.

Exemples. On considère l'alphabet $\Sigma = \{a, b\}$.

1. Le langage L_1 dénoté par a^* est-il local ?

2. Le langage L_2 dénoté par $(ab)^*$ est-il local ?

3. Le langage L_3 dénoté par $a^* + (ab)^*$ est-il local ?

4. Le langage L_4 dénoté par a^*ba est-il local ?

3.2 Stabilité des langages locaux

Propriété 12 (Stabilité des langages locaux)

(1) Si L_1 et L_2 sont deux langages locaux, alors $L_1 \cap L_2$ est un langage local.

(2) Si L est un langage local, alors L^* est aussi un langage local.

Preuve.

□

Attention.

En revanche, la réunion ou la concaténation de deux langages locaux n'est pas forcément un langage local. Considérons les langages locaux $L_1 = ab$ et $L_2 = a^*$.

- $L_1.L_2$ n'est pas local car :

- $L_1 \cup L_2$ n'est pas local car :

Propriété 13 (Réunion/concaténation de deux langages locaux)

Soit L_1 et L_2 deux langages locaux définis sur des alphabets disjoints.
Alors $L_1.L_2$ et $L_1 \cup L_2$ sont des langages locaux.

Preuve.

□

3.3 Cas des expressions rationnelles linéaires

Remarque. Nous l'avons vu, une expression rationnelle générale ne dénote pas nécessairement un langage local. Il y a cependant un cas particulier pour lequel c'est le cas : les expressions rationnelles linéaires.

Définition.

Une expression rationnelle est dite **linéaire** si chaque lettre de Σ y apparaît au plus une fois.

Exemple. L'expression rationnelle $(ab)^*$ est linéaire, mais pas l'expression $a.b^* + b$.

Théorème 14 (Expressions rationnelles linéaires et langages locaux)

Toute expression rationnelle linéaire dénote un langage local.

Preuve.

□

3.4 Algorithmes de calcul des ensembles P, S et F

Nous utiliserons le type OCaml suivant pour représenter en machine une expression rationnelle :

Par exemple, l'expression rationnelle $(a+b)^*c$ est représentée en OCaml par :

Les différents résultats établis au sujet des langages locaux nous permettent d'écrire des algorithmes de calcul des ensembles P, S et F associés à une expression rationnelle linéaire, en procédant par induction structurelle.

Nous avons observé que pour calculer l'ensemble des préfixes et suffixes du langage local dénoté par une expression rationnelle linéaire, il est nécessaire de posséder :

- une fonction qui détermine si le mot vide appartient au langage :

- une fonction qui fait l'union de deux langages (représentés par des listes) :

Nous définissons maintenant les fonctions qui calculent les ensembles P et S :

Pour calculer l'ensemble F, il faut posséder une fonction qui calcule le produit cartésien de deux listes :

Nous en déduisons la fonction calculant F :

On obtient alors par exemple pour l'expression rationnelle $(a+b)^*c$: