ROYAUME DU MAROC
UNIVERSITE ABDELMALEK ESSAADI
Ecole Nationale des Sciences Appliquées

Tanger

Année universitaire 2019-2020

Cours : Théorie des Automates / Chapitre I

Mots et Langages

Les concepts fondamentaux de la théorie des langages formels

Version 6.0

Introduction

L'objectif de la théorie des automates est :

→ Proposer des modèles de mécanismes mathématiques qui formalisent les méthodes de calcul.

Cette théorie est le fondement de plusieurs branches importantes de l'informatique théorique. La théorie des automates est l'étude des machines abstraites qui permettent de formaliser les méthodes de calcul.

L'objet traité par un automate est un mot d'un langage.

Pour arriver à la généralité souhaitée, on convertit un « **problème** » en un **langage**, et la résolution du problème, en l'analyse d'un élément de ce langage

On représente chaque **instance** d'un « problème » par un **mot**. Savoir si l'instance du problème a une solution se ramène à tester si ce mot appartient au langage des mots représentant les instances du problème et qui ont une solution.

Un automate qui résout le problème prend en entrée un mot et décide s'il est accepté ou non.

Exemple :

Le problème de savoir si un entier *N* est premier (test de primalité) peut se traduire comme suit : on représente tous les entiers naturels par des chaînes binaires (écriture en base 2).

Dans ce langage, les mots représentant des nombres premiers forment un sous-ensemble. Le problème du test de primalité consiste alors à savoir si la chaîne binaire représentant un nombre *N* appartient à ce sous-ensemble ou non.

Un automate approprié prend en entrée une chaîne binaire et l'accepte précisément lorsqu'elle représente un nombre premier.

La formulation des problèmes et de leur résolution (ou même de leur calculabilité) en termes de langage formels est à la base des hiérarchies de complexité, et des hiérarchies des langages formels.

Un autre domaine concerne la *transformation* de mots. Dans ce domaine, on utilise plutôt le terme de « machine» ou de transducteur. La linguistique, mais aussi la compilation, font usage de tels transducteurs pour l'analyse et la transformation de textes ou de programmes.

D'abord, nous allons introduire :

- Quelques concepts fondamentaux de la théorie des **langages formels** et
- Quelques concepts fondamentaux de la **combinatoire** sur les mots. La **combinatoire** des **mots** étudie les propriétés des suites de **symboles**.

La théorie des **langages formels** englobe **la théorie des automates** et s'intéresse aux propriétés mathématiques des langages qui sont des **ensembles de mots**. Elle trouve notamment des applications en vérification et pour la compilation.

a. Alphabet

Un alphabet est un ensemble fini de symboles.
Un alphabet sera désigné par une lettre grecque majuscule.

Exemples :

$$\Sigma = \{a, b, c\},\ \Gamma = \{\heartsuit, \diamondsuit, \clubsuit, \spadesuit\},\ \Delta = \{0, 1\}, \Phi = \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$$

Les éléments d'un alphabet sont appelés ***lettres*** ou ***symboles***.

Les lettres n'ont pas de propriétés particulières. On demande seulement de savoir tester si deux lettres sont **égales** ou **différentes**.

Parmi les exemples d'alphabets :

- Il y a bien sûr l'alphabet latin, et tous les alphabets des langues naturelles.
- Il y a aussi l'alphabet binaire,
  - composé des symboles 0 et 1,
- L'alphabet hexadécimal,
- L'alphabet des acides aminés, etc.
- En informatique, on rencontre l'alphabet des lexèmes, c'est-à-dire des unités syntaxiques résultant de l'analyse lexicale d'un programme.
- En Biologie : le biologiste intéressé par l'étude de l'ADN utilisera un alphabet à quatre lettres : {A;C; G; T}
  pour les quatre constituants des gènes :
  Adénine, Cytosine, Guanine et Thymine

b. Mots ou chaînes

Un mot ou une chaîne sur un alphabet Σ est une suite finie (et ordonnée) d'éléments de Σ *:*

$$w = (w_1, \ldots, w_n)$$

On écrit plutôt :

$$w = w_1 \ldots w_n$$

L'entier *n* est la ***longueur*** du mot.

Exemple : abbac et ba sont deux mots sur l'alphabet Σ = {a, b, c}.

La longueur d'un mot w : est le nombre de symboles constituant ce mot; on la note |w|.

Ainsi, |abbac| = 5 et |ba| = 2:

Il existe un seul mot de longueur 0, appelé le *mot vide*, et noté souvent $\mathcal{E}$. | $\mathcal{E}$ |=0

L'ensemble des mots sur Σ est noté Σ^*.

Par exemple : $\{a, b, c\}^* = \{\mathcal{E}, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, \ldots\}$.

Le nombre d'occuence d'un symbole dans un mot :

Si σ est une lettre de l'alphabet Σ, pour tout mot $\omega \in \Sigma^*$ tel que $\omega = \omega_1 \ldots \omega_k$,

On dénote par : $|\omega|_\sigma = \#\{i \in \{1,\ldots,k\} / \omega_i = \sigma\}$

Le nombre de lettres σ apparaissant dans le mot ω.

Par exemple :

- $|abbac| = 5$
- $|abbac|_a = 2$
- $|abbac|_c = 1$.

Les préfixes et suffixes et Facteur d'un mot :

Soit $\omega = \omega_1 \ldots \omega_l$ un mot sur Σ. Et Soient $1 \leq i \leq j \leq l$.

Les préfixes :

- Les mots : ε, ω_1, $\omega_1 \omega_2, \ldots$; $\omega_1 \ldots \omega_{l-1}$; $\omega_1 \ldots \omega_l = \omega$ sont les préfixes de ω.
- Un préfixe de ω diffèrent de ε et de ω est dit préfixe propre.
- L'ensemble des préfixes de ω est noté Pre(ω)

Les suffixes :

- Les mots : ε, ω_l, $\omega_{l-1}\omega_l, \ldots, \omega_2 \ldots \omega_l$; $\omega_1 \ldots \omega_l = \omega$ sont les suffixes de ω.
- Un suffixe de ω est qualifié de suffixe propre s'il diffère de ε et de ω.
- L'ensemble des suffixes de w est noté Suf(ω)

Les facteurs :

- Le mot $\omega_i \ldots \omega_j$ est un **facteur** du mot ω. On le note parfois ω[i, j].
- Une fois encore, on parle de facteur propre lorsque ce dernier diffère de ω et de ε.
- L'ensemble des facteurs de ω est noté Fac(ω).

Exemple :

Soit le mot : ω = abaababa

- aba est à la fois préfixe et suffixe de ω
- abaa est un préfixe qui n'est pas suffixe de ω
- baba est un suffixe qui n'est pas préfixe de ω
- aaba est un facteur qui n'est ni préfixe ni suffixe de ω
- aaa n'est pas facteur de ω

Remarques :

- Remarquons que le mot vide est préfixe, suffixe et facteur de tout mot.
- Un mot de longueur n a :
  - exactement n+1 préfixes (n-1 préfixes propres) distincts
  - et également n+1 suffixes (n-1 suffixes propres) distincts,
  - Mais on ne connait pas à priori le nombre de facteurs distincts d'un mot de longueur donnée.
- Une façon naturelle pour exprimer des conditions sur les mots est de le faire en termes de préfixe, de suffixe, ou de facteur. Par exemple : commencer ou ne pas commencer de telle ou telle façon

Exemple :

Soit B = {0,1} l'alphabet des chiffres 0 et 1. Chaque mot de B^* est l'écriture binaire d'un entier positif et chaque entier strictement positif a une écriture binaire, unique si l'on spécifie qu'elle ne commence pas par 0. Les entiers divisibles par 8 sont caractérisés par le fait que leur écriture binaire se termine par 000.

La relation « être préfixe » est une relation d'ordre sur A^* on notera $g \leq f$ si g est un préfixe de f.

On appellera factorisation d'un mot f une suite $(g_1,g_2,\ldots,g_n)$ de mots telle que :
$f = g_1g_2\ldots g_n$ (par abus on écrira : « soit $f = g_1g_2\ldots g_n$ une factorisation de f ») ; cette factorisation est dite propre si aucun des g_i n'est égal au mot vide (et si i n'est pas nul).

Sous-mots :

Puisqu'un mot f est une suite de lettres, on appellera sous-mot g de f toute sous-suite de la suite f.

Tout facteur de f est un sous-mot de f, mais la réciproque est fausse, et il ne faut pas confondre les deux notions :

aaaa, bb et babaa sont des sous-mots de f = abaababa sans en être des facteurs.
Autrement dit, g est un sous-mot de f si f contient les lettres de g dans le bon ordre mais nécessairement de manière consécutive.

Le *produit de concaténation* *:*

Le produit de concaténation de deux mots x et y est le mot xy obtenu par juxtaposition des deux mots :

$x = a_1\ldots a_n$ et $y = b_1\ldots b_m$ $xy = a_1\ldots a_nb_1\ldots b_m$

En particulier*, on définit la puissance nième d'un mot ω comme étant la concaténation de n copies de ω,*

$$\omega^n = \underbrace{\omega\ldots\omega}_{n \text{ fois}} \qquad \text{On pose : } \omega^0 = \varepsilon.$$

Période :

Soit $w = w_1\ldots w_l$ un mot, avec $w_i \in \Sigma$ pour tout i.
L'entier $k \geq 1$ est une période de w si $w_i = w_{i+k}$, $\forall\, i = 1,\ldots,l-k$.
On dit aussi que w est k-périodique.
Un mot 1-périodique est constant.
Par exemple, le mot : abbabbabba est 3-périodique.

Image miroir :

Si $f = a_1a_2\ldots a_n$ est un mot de A^*, on appelle image miroir de f (ou transposé de f) le mot f^t :

$$f^t = a_na_{n-1}\ldots a_1$$

Ainsi, $(abaababa)^t = ababaaba$
Les préfixes de f^t sont les transposés des suffixes de f et vice versa.
Les facteurs de f^t sont les transposés des facteurs de f.

Définition : Palindrome

On définit, par récurrence sur la longueur de w, l'opération miroir de la manière suivante :

Si |w| = 0, alors w = Ɛ Et w^t = Ɛ;

Sinon |w| > 0 et w = σu, σ ϵ Σ, u ϵ Σ^* et $w^t = u^t \sigma$.

Si w est tel que $w^t = w$, Alors w est un palindrome.
Comme : « été », « radar » sont des palindromes connus.

On notera : Pal_A ou plus simplement Pal s'il n'y a pas ambiguïté sur l'alphabet, l'ensemble des palindromes de A^*.

$$Pal = \{f \epsilon A^* \mid f = f^t\}$$

c. Langage formel :

Définition ;

Un langage formel sur un alphabet *A* est un ensemble de mots sur A^* donc un sous-ensemble de A^*.

On distingue en particulier le langage vide.

Remarque :

Ne pas confondre le langage vide ne contenant aucun élément et le langage {Ɛ} contenant uniquement le mot vide.

Les opérations ensemblistes (union, intersection, complément) s'étendent bien entendu aux langages.

Exemple :

Considérons l'alphabet Σ = {a, b, c}.

L'ensemble {a, aa, bbc, ccca, ababab} est un langage fini.

L'ensemble L_{2a} des mots sur Σ^* comprenant un nombre pair de a est aussi un langage (infini),

- L_{2a} = {Ɛ, b, c, aa, bb, bc, cb, cc, aab, aac, aba, aca, …, abaacaaa,…}
- L'ensemble Pal(Σ^*) formé des palindromes de Σ est aussi un langage infini,
- Pal(Σ^*) = {Ɛ, a; b; c; aa; bb; cc; aaa; aba; aca; bab; bbb; bcb; cac; cbc; ccc; aaaa; abba; acca; baab; bbbb; bccb; caac; cbbc; cccc;…}:
- Soit l'alphabet Δ = {0; 1}. L'ensemble constitué des écritures binaires des entiers positifs pairs est un langage sur Δ

{10; 100; 110; 1000; 1010; 1100; 1110;…}

Remarque :

Un mot $\omega = \omega_l \ldots \omega_0 \in \{0,1\}^*$ représente l'entier n si $\sum_{i=0}^{l} \omega_i 2^i$

En général, on ne considère que des mots dont le premier symbole ω_l diffère de 0. Par convention, l'entier zéro est alors représenté par le mot vide.

- Soit A = {a, b} et Z_1 = {f ϵ A^* | $|f|_a = |f|_b$} : l'ensemble des mots de A^* qui contiennent autant de a que de b,
- Soit A = {a, b} et L_2 = {f ϵ A^* | |f| = 1 mod 2} : l'ensemble des mots de longueur impaire,
- Soit A = {a, b} et K_1 = {$a^n b^m$ | n, m ϵ **N**} : l'ensemble des mots constitués d'une suite de a suivie d'une suite de b.
- De même que le langage formé des écritures binaires des nombres premiers :

{10; 11; 101; 111; 1011; 1101; 10001;…}

Remarque :

$Z_1 \cap L_2 = \emptyset$
$Z_1 \cap K_1 = \{a^n b^n \mid n \in \mathbf{N}\}$
$A^* \setminus K_1$ = {f ϵ A^* | f contient un facteur de ba}

Les opérations sur les Langages :

Produit de concaténation :

Le produit de concaténation des mots s'étend aux langages de la manière suivante. Si *X* et *Y* sont deux langages sur A^*, leur produit est le langage

$$XY = \{xy \mid x \in X \text{ et } y \in Y\}$$

En particulier, on peut définir la puissance nième d'un langage L, n > 0, par :

$$L^n = \{\omega_1 \ldots \omega_n \mid \forall i \in \{1; : : : ; n\}, \omega_i \in L\}$$

Et on pose $L^0 = \{\mathcal{E}\}$.

Par exemple, si L = {a; ab; ba; ac},

Alors

L^2 = {aa, aab, aba, aac, aba, abab, abba, abac,
baa, baab,baba, baac, aca, acab, acba, acac}.

Remarque :

Soit n ≥ 0. L'ensemble des mots de longueur n sur Σ est Σ^n.

Notons aussi que si un mot uv appartient à LM avec u ϵ L et v ϵ M, cette factorisation n'est pas nécessairement unique.

Par exemple, avec L = {a; ab; ba},
L^2 contient le mot aba qui se factorise en a(ba) et (ab)a.

Demander l'unicité de la factorisation débouche sur la notion de code.

Ainsi, X inclus dans Σ^* est un code, si tout mot de X* se factorise de manière unique comme concaténation de mots de X.

Proposition :

La concaténation de langages est une opération associative, elle possède {Ɛ} pour neutre, ∅ pour absorbant et est distributive à droite et à gauche pour l'union, i.e. :

Si $L_1;L_2;L_3$ sont des langages

- $L_1(L_2L_3) = (L_1L_2)L_3$;
- $L_1\ \{Ɛ\} = \{Ɛ\}\ L_1 = L_1$,
- $L_1\ ∅ = ∅\ L_1 = ∅$
- $L_1(L_2 \cup L_3) = (L_1L_2) \cup (L_1L_3)$;
- $(L_1 \cup L_2)L_3 = (L_1L_3) \cup (L_2L_3)$:

Une autre opération est l'étoile de Kleene :

Pour une partie $L \subset \Sigma^*$ L'étoile de Kleene, de L, est donnée par :

$$L^* = \bigcup_{i \geq 0} L^i$$

Ainsi, les mots de L^* sont exactement les mots obtenus en concaténant un nombre arbitraire de mots de L.

Remarque *:*

On remarque que la notation Σ^* introduite précédemment est cohérente puisqu'il s'agit en fait de l'étoile de Kleene du langage fini Σ.

On dit parfois que Σ^* est le monoïde libre engendré par Σ.

On rencontre parfois l'opération L^+ définie par :

$$L^+ = \bigcup_{i \geq 1} L^i$$

Par exemple, si Σ est un alphabet, alors $\Sigma^+ = \Sigma^* \setminus \{Ɛ\}$.

D'une manière générale, si L est un langage ne contenant pas le mot vide,

Alors : $L^+ = L^* \setminus \{Ɛ\}$.

Définition:

On peut étendre les opérations d'obtention de préfixes, suffixes et facteurs aux langages. Soit L un langage.

On définit l'ensemble des préfixes des mots du langage L :

$$Pref(L) = \bigcup_{\omega \in L} Pref(\omega)$$

De la même manière, on pose

$$Suff(L) = \bigcup_{\omega \epsilon L} Suff(\omega)$$

$$Fac(L) = \bigcup_{\omega \epsilon L} Fac(\omega)$$

Un langage est dit préfixe s'il ne contient pas deux mots dont l'un est préfixe propre de l'autre.
Un langage est dit suffixe s'il ne contient pas deux mots dont l'un est suffixe propre de l'autre.

Enfin, un langage L est préfixiel si Pref(L) = L.

Il suffit donc de vérifier que, tout préfixe d'un mot de L, est encore un mot de L.

De la même manière, L est suffixiel si Suff(L) = L.

De la même manière, L est factoriel si Fac(L) = L.

Définition :

Soit f un morphisme de monoïdes entre Σ^* et Γ^*

On remarque que f est complètement caractérisé par les images de f sur les symboles de Σ.

Si L est un langage sur Σ, alors l'image de L par le morphisme f est

$$f(L) = \{f(u) \in \Gamma^* | u \in L\}$$

De la même manière, si M est un langage sur Γ, alors l'image inverse de M par le morphisme f est :

$$f^{-1}(M) = \{u \in \Sigma^* | f(u) \in M\}$$

Exemple :

Soient $\Sigma = \{a, b, c\}$ et $\Gamma = \{\mu, \nu\}$ et f le morphisme défini par :

$$f(a) = \mu \quad \text{et} \quad f(b) = \nu \text{ et } \quad f(c) = \nu$$

Si $L = \{ab, bc, cb, aaab, aaac\}$, alors

$$f(L) = \{\mu\nu, \nu\nu, \mu\mu\mu\nu\}$$

Si $M = \{\mu\nu, \nu\mu, \nu\mu\nu\}$ alors

$$f^{-1}(M) = \{ab, ac, ba, aaab, aaac\},$$

Dans notre exemple, pour tout $\sigma \epsilon \Sigma, |f(\sigma)| = 1$.

Néanmoins, on peut en toute généralité considérer un morphisme dont les images des lettres de l'alphabet d'origine seraient de longueurs différentes.

Remarque :

Il arrive, dans de nombreuses situations, qu'on distingue le cas où il existe $\sigma \epsilon \Sigma$ tel que f(σ) = ε, du cas où, pour tout $\sigma \epsilon \Sigma$, f(σ) ≠ ε.

Dans la section précédente, on a introduit le miroir d'un mot. Cette opération s'étend naturellement aux langages.

Définition :

Le miroir d'un langage L est :

$$L^R = \{u^R | u \epsilon \text{ L}\}$$

On peut avoir $L = L^R$ sans pourtant autant que les mots de L soient tous des palindromes.

Définition :

La clôture commutative d'un langage $L \subseteq \Sigma^*$ est définie par

$$Com(L) = \{\omega \epsilon \Sigma^* | \exists\, u \epsilon L : \sigma \epsilon \Sigma, |\omega|_\sigma = |u|_\sigma\}$$

Cela signifie que Com(L) contient les mots obtenus en permettant les lettres des mots de L

Par exemple,

Si $L = \{ab, bac, ccc\}$, alors $Com(L) = \{ab, ba, abc, acb, bac, bca, cab, cba, ccc\}$

En utilisant la fonction de Parikh déjà introduite il est clair que : $Com(L) = \Psi^{-1}\Psi(\text{L})$

Si L est un langage tel que : $Com(L) = L$ alors L est dit commutatif.

Voici une dernière sur les mots et les langages

Définition :

Le shuffle de deux mots u et v est le langage u ш v

$$\text{u ш v} = \{u_1 v_1 \dots u_n v_n | u = u_1 \dots u_n, v_1 \dots v_n, u_i v_i \in \Sigma^*, n \geq 1\}$$

Par exemple : si u = ab et v = cde

Alors u ш v = {abcde, acbde, acdbe, acdeb, cabde, cadbe, cadeb, cdabe, cdaeb, cdeab}

Le shuffle de deux langages se définit comme suit

$$\text{L ш M} = \bigcup_{\substack{u \in L \\ v \in M}} \text{u ш v}$$