Théorie des langages

1 Alphabets, Langages et Grammaires

1.1 Alphabets et mots. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1.2 Opération sur les mots . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1.3 Langages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1.4 Grammaires . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1.5 Types de grammaires . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

# Alphabets, Langages et Grammaires

1.1 Alphapets et mots

1. Un Alphabet :

Un alphabet est un ensemble finis non vide de symboles.

Exemples d’alphabets :

A = {0,1}

B = {a,b,c}

C = { •, ?, \*}

D = { if, then, else, id, nb, =, + }

1. Un mots :

Un mots sur l’alphabet X est une séquence fini et ordonner,(éventuellement vide) d’élèments de l’alphabets.

­🡪C’est une concaténation des symboles.

{ε,,a,bac,ca}

1. Mot vide :

Le mot vide noté’ε’, correspant à la suite de symbole vide.

1. Largeur d’un mot :

Langeur d’un mot ‘W’ est le nombre de symbole constituant ce mot, on l’a note |W|.

Exemple :

|bac|=3

|a|=1

| ε |=0

1. Ensemble de mots :

L’ensemble de mots sur un X est noté X\*.

Exemple :

Soit X={a,b,c}

X\*={ ε , aa , bcc , abc, …}

1. Nombre d’occurrence d’un symbole :

Soit W ∈ X\* :

°|W| la longeur du mot W.

° X est un alphabet.

° x ∈ X // x est un symbole.

🡪 |W|x : nombre d’occurrence de symbole x dans le mot W.

# 1 Alphabets, Langages et Grammaires

Exemple :

X={a,b}

|abb|b = 2

|aa|b = 0 // b existe dans X.

|abb|c = aucun réponse // c n’existe pas dans X.

1.2 Opération sur les mots

* 1. Concaténation :

1. Concaténation de lettres :

Soit un alphabet X et w1 , w2 ∈ X\* tel que w1 = x1 , x2 , … , xn.

Tel que xi ∈ [1 … n] ,xi ∈ X.

W2 = y1 , y2 , … ,yp tel que yi ∈ [1 … p ] , yi ∈ X.

🡪 On dit que w1 et w2 sont une concaténations de lettres.

1. Concaténation de mots :

Soit X et w1 , w2 ∈ X\* , alors si w = w1 . w2 = x1 … xn . y1 … yp .

🡪 On dit que w est une concaténation.

1. Propriétés de la concaténation :
2. Le produit est associative ( ∀ w1 , w2 , w3 ∈ X\* ).

( w1 . w2 ) . w3 = w1 ( w2 . w3 ) .

1. ε est l’élément neutre du produit : ∀ w ∈ X\*

ε . w = w . ε = w .

1. ∀ w1 , w2 ∈ X\* , la longeur | w1 , w2 | = |w1| + |w2| de même ,

| w1 . w2 |x = | w1 |x + | w2 |x tel que x ∈ X.

1. Le produit n’est pas commutative ∀ w1 , w2

w1 . w2 w2 . w1

* 1. Puissance :

w ° = ε

On note w ⁿ la concaténation de n copies de w , avec w° = ε .

Soit X = {a,b} , w = abb

w² = abb abb

abb² = abbb

(abb)² = abb abb

1. Egalite :

Deux mots sont égaux s’ils sont de même longeur et s’ils sont des lettres indentique de positon indentique.

1. Sufixe et préfixe :

Soit un alphabet X est w , u ∈ X\* :

⦁ u est préfixe de w, si et seulement si ∃ v ∈ X\* tel que w = u . v

# 1 Alphabets, Langages et Grammaires

⦁ u est sufixe de w , si et seulement si ∃ v ∈ X\* tel que w = v . u

Exemple :

Soit X = {a,b} ; w = babb

* Donner les préfixes et les sufixes de w :

\*Préfixe de w : { ε , b , ba , bab , babbb }

\*Sufixe de w : { ε , b , bb , abb , babb }

Propriétés :

1. On dit que u est un préfixe propre de w , si et seulement si u est un préfixe de w et u ≠ w .
2. On dit que u est un sufixe propre de w , si et seulement si u est un sufixe de w et u ≠ w .
3. Miroir d’un mot :

Soit un alphabet X et w ∈ X\* tel que w = x₁ , x₂, … , xₙ tel que xᵢ ∈ [1 … n ] xᵢ ∈ X.

Le miroir de w est noté R.

wᴿ = w = xₙ . xₙ\_₁ … x₁

En général :

wᴿ = w si w = ε.

1.3 Langages :

1) Définition :

Un langage sur un alphabet X est une partie de X\*.

Il s’agit donc d’un ensemble de mots, noté L .

L ⊂ X\* où L ∈ P(x\*).

Exemple :

Soit X = {a,b}

* Exemples de langage sur X\* :

L₁ = {ab , aa , aab}

L₂ = {aⁿ tel que n>0}

2) Opération sur les langages :

1. Union :

A,B ⊂ X\*

A ∪ B = { w ∈ X\* / w ∈ A et w ∈ B }

🡪 C’est une opération associative et commutative (A∪B ou B∪A c’est le même).

Elle est noté (+) dans la théorie de langages.

1. Intersection :

A,B ⊂ X\*

A ∩ B = A,B ⊂ X\*

A ∩ B = { w ∈ X\* / w ∈ A et w ∈ B }

🡪 C’est une opération associative et commutative.

1. Différence :

# 1 Alphabets, Langages et Grammaires

A,B ⊂ X\*

A\B = { w ∈ X\* / w ∈ A et w ∉ B' }.

1. Complémentation :

A,B ⊂ X\*

= X\*\A = { w ∈ X\* / w ∉ A }.

1. Produit :

Soit deux langages A,B ⊂ X\*

A . B = { u . v tel que u ∈ A et v ∈ B }.

Exemple :

Soit un alphabet X

X = {a,b} ; B = { b,ba}

A = { ε,a,ab}

A . B = { ab , aba , abb , εb , εba , abba }

B . A = { ba , bab , baa , baab , bε , baε }

🡪 L’opération de produit n’est pas commutatif.

1. Fermuture de Kleene :

Soit langage A ⊆ X\* , on note A\* = ∪ᵢ∞ =0 Aͥ C’est la fermuture par \* du langage A .

A° = { ε } 🡪 ε ∈ A\*

1. Fermuture positive :

Soit A ⊆ X\* , on note Aᶧ = uᵢͦͦ = Aͥ C’est la fermuture positive du langage A .

Théoréme :

Aᶧ = A\* . A = A . A\*

Aᶧ = Aᶧ + { ε }

1.4 Grammaire :

Représente l’ensemble des régles pour générer les mots du langage à partir de ces régle

est appelé « Dérivation » .

* Définition :

Grammaire le quadruple (V,N,S,R) tel que :

•V : l’ensemble fini des symboles terminaux ( symboles autorisée pour avoir un mots final ).

•N : l’ensemble fini ( disjoint de V ) de symboles non-terminaux .

•S : un symbole non-terminal particulier appelé « Axiome » .

•R : ensemble des régles de production de la forme :

g→d tel que g ∈ (V+N) ᶧ et d ∈ (V+N)

g→d : dérivation et signifie que g peut être remplacé par d.

Exemple1 :

Soit la grammaire :

G = ({a} , {S} , S , {S →aS , S→a} {S→aS|a}

On peut construire la chaine de dérivation suivante :

* Donner des mots qui peuvent être générer par G :

S→aS→aaS→aaaS→aaaa

# 1 Alphabets, Langages et Grammaires

S→aS→aa

* Définition :

L(G) : le langage de tous les mots générés par la grammaires G .

Exemple2 :

Soit G = ({a,b} , {S,T}, S , {S→aS|aT, T→bT|b})

1/ Les mots « abb » et «aab » peuvent- être générer par G ?

S→aT→abT→abb

S→aS→aaT→aab

2/Déduire une expression générale du langage générer par G :

L(G) = {aⁿ b ͫ tel que n,m >0 }

* Définition :

Un grammaire G = (V,N,S,R) construit l’arbre syntaxique de G tel que :

•La racine est l’axiome S.

•Les nœuds sont les symboles terminaux.

1.5 Type de grammaire :

La classification des grammaires, définie par Noam CHOMSKY, distingue les quatre classes suivantes :

* Type3 : grammaire réguliére :

A→WB|W avec { A,B ∈ N , W V\*} (S→aaS|aa)

* Type2 : grammaire hors-contexte :

A→B avec {A ∈ N , B ∈ (V+N)\*} (S→aSaS|aa)

* Type1 :grammaire contextuelle :

α →ᵦ avec {|α| ≤ |ᵦ| , α ∈ (V+N)ᶧ , ᵦ ∈ (V+N)\*} (ST→aSaS|aa)

* Type0 : pas de restriction sur les règles.

α →ᵦ avec { α ∈ (V+N)ᶧ , ᵦ ∈ (V+N)\*} (ST→aSaS|a)