2. Théorie et Calcul sur les mots

Bernard ESPINASSE

Professeur à l'Université d'Aix-Marseille

1998

• **Introduction**
• **Le monoïde libre**
• **Opérations sur les mots**
• **Congruence de Thue**
• **Calcul associatif**
• **Équivalence au sens de Thue ≈**
• **Le problème des mots**

Introduction

Théorie et calcul sur les mots :

• présentation générale du cadre dans lequel se situe la théorie des langages (définitions).

• le phénomène de l'écriture sous son aspect le plus immédiat : emploi de symboles que l'on dispose les uns derrière les autres pour former des mots

-> structure de monoïde libre : cas particulier des chaînes de caractères munies de la **concaténation**

-> théorie sur les mots

• **calcul associatif** construit sur la **congruence** et l'**équivalence de Thue**

• **problème des mots** et sa résolution

-> calculs sur les mots

Le monoïde libre: alphabet, mot et vocabulaire

• Un **alphabet** X = ensemble non vide fini de symboles (lettres) : X = {s_1, s_2, s_3, ...sn}

  • on défini sur l'alphabet X une relation d'ordre totale notée ≤ telle que : $s_i \leq s_j$ alors s_i sera placé avant s_j dans la suite formée par les éléments de X .

  • *exemples :*

    • B = {0, 1} = alphabet à 2 lettres dit alphabet binaire;
    • M = {., -, /} = alphabet Morse;
    • D = {0, 1, 2, 9} et L = {a, b, c, ..., z} = alphabets quelconques

• Un **mot** sur un alphabet X = suite finie et totalement ordonnées composées d'éléments de X

• Un **vocabulaire X*** défini sur un alphabet X = ensemble des mots constitués à partir des éléments de X.

  • X* contient l'élément vide ∅ dont la séquence n'est formée d'aucun élément de X.

  • *exemple :* Soit l'alphabet X = {a, b},
  on peut définir le vocabulaire X* = { ∅, U_1, U_2, ...U_n} avec U_i mot de X*.

Convention : majuscules = mots , minuscules = lettres de l'alphabet

Longueur, format et sous-mot d'un mot

Si [p] = suite d'entiers (1, 2, ..., p), un mot U = application de [p] dans X soit: U: [p] -> X.

• p appelé **longueur** (ou degré) du mot U, notée **|U|**

• Le **format** fixe la longueur des mots (ici le vocabulaire X* est de format variable)

exemple :

Soit l'alphabet X = {a, b}, soit U et V deux mots de X*

  • U = 'abaa' : U(1)=a, U(2)=b, U(3)= a, U(4) = a, |U| = 4 (U de longueur 4),
  • V = 'baa' : V(1)=b , V(2)=a, V(3)=a, |V| = 3 (V est un mot de longueur 3),

• Un **sous-mot** d'un mot U = mot constitué d'une sous suite de la suite des lettres constituant U

exemple : Soit l'alphabet X = {a, b, c} et soit U, V et W les trois mots de X* :

  • U = 'aabca' : U(1) = a, U(2) = a, U(3) = b, U(4) = c, U(5) = a,
  • V = 'abc' : V(1) = a, V(2) = b, V(3) = c
  • W = 'cba' : W(1) = c, W(2) = b, W(3) = a

  V est un sous mot de U, mais W n'en n'est pas un.

L'opération de concaténation

- Dans le vocabulaire X* de tous les mots définis sur un alphabet X on définit une opération, loi de composition interne, **concaténation** :

  Si **U** et **V** sont 2 mots de longueur p et q, la **concaténation** de U et V (dans cet ordre) et notée **UV** est le mot **W** de longueur p+q défini par :

  $W(i) = U(i)$ si $1 \leq i \leq p$

  $W(p+i) = V(i)$ si $1 \leq i \leq q$

- Le produit de concaténation est noté par **simple juxtaposition** (car loi associative et un mot = produit des lettres qui le composent)

  exemple :

  Soit l'alphabet X = {a, b, c}

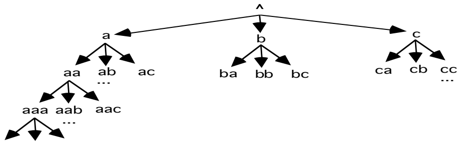
  Soit U et V ∈ X* avec : U='baca' et V = 'ac', on a : UV = 'bacaac' et VU = 'acbaca'

- Propriétés :
  - loi **associative** : $(U_1U_2)\ U_2 = U_1(U_2\ U_2)$
  - admet un **élément neutre**, mot vide et noté ∧ : ∀ U : ∧U = U∧ = U.
  - n'est **en général pas commutative** : $U_1U_2 \neq U_2U_1$

La concaténation confère à X* la structure algébrique de **monoïde libre** construit sur X.

Rappels d'algèbre sur la structure de monoïde

- **monoïde** = tout ensemble muni d'une **loi de composition interne associative**.
- **monoïde libre ou unitaire** = monoïde possédant un **élément neutre** (∧)
- **Représentation arborescente du monoïde** X*: *ex:* alphabet à trois lettres X = {a, b, c} :



- **Propriétés des monoïdes:** soit E monoïde muni d'une opération associative (.) : a.b ou ab :

Puissances : le composé d'un élément C de E avec lui-même: $C^n = C.C...C$ (n termes C).
Inverses : Si le monoïde est libre et si E est inversible à droite et à gauche, les inverses sont égaux donc unique. Si C_l et C_2 de E sont inversibles, C_lC_2 aussi et $(C_lC_2)^{-1} = C_2^{-1}\ C_1^{-1}$

Règles de calcul sur les exposants : pour tous n, p ∈ N* et pour tout C de E, on a $C^n\ C^p = C^{n+p}$ et $(C^n)^p = C^{np}$ et :

- si C_l et C_2 commutent $(C_lC_2)^n = C_l^nC_2^n$ (ex: A='ab' ; $A^{(-1)D}$ ='ba'; $A^{(-1)G}$ ='ba' d'où $A^{(-1)D} = A^{(-1)G}$)
- si C_l et C_2 sont inversibles : $C^{-n} = (C^n)^{-1} = (C^{-1})^n$ avec $n \in N^*$.

Sous-monoïde

- **sous monoïde** = sous-ensemble d'un monoïde

pour la loi de composition interne (un sous monoïde a aussi une structure de monoïde)

exemple 1: sous monoïde engendré par une partie d'un monoïde :

- soit alphabet X = {a, b} alors :

  X^+ = {a, b, a a, a b, b a, b b, a a a, a a b, a b a, a b b, b a a ... } = **monoïde**,

  X* = {∧,a,b, a a, a b, b a, b b, a a a, a a b, a b a, a b b, b a a ...} = **monoïde libre** *(∧ = élément neutre de la loi de composition interne)*

- soit S sous-ensemble de X^+ formé des chaînes ne contenant que le symbole a :

  S = {a, a a, a a a, a a a a ...)

  S = sous monoïde de X^+ engendré par {a}

  = plus petite partie stable de X^+ contenant {a}

  = partie de X^+ obtenue en composant de toutes les façons possibles les éléments de {a}.

→ le monoïde engendré par le sous-ensemble est l'ensemble de toutes les chaînes finies formées à l'aide de ces éléments.

Opérations sur les mots : Partie d'un mot

- Soit un mot U (évent. vide), U = **partie du mot** A s'il **∃ 2** mots B et C tels que : A = BUC. le triplet (B,U,C) = une **décomposition** de A.
- triplet (B, U, C) = **décomposition normale** de A si ∀ un autre triplet (B',U,C') on a : |B| ≤ |B'|

  exemple

  Décompositions de A = '01101011' ayant pour 2° élément le mot U = '101':

  - 1° triplet : (01, 101, 011) *décomposition normale*
  - 2° triplet : (0110, 101, 1)

  Dans (B,U,C) si B = ∧, le triplet (∧,U,C) = *décomposition normale* de A.

Théorème 1: si U est une partie de A, il **∃** une et une seule décomposition normale de A.

Théorème 2: soit (B_1,U,C_1) et (B_2,U,C_2) tels que $B_1 = B_2$ ou $C_1 = C_2$ alors : $(B_1,U,C_1) = (B_2,U,C_2)$.

Théorème 3: soit (B_1,U,C_1) et (B_2,U,C_2) :

si $(B_1,U,C_1) = (B_2,U,C_2)$ et $B_1 = B_2$ alors $C_1 = C_2$

si $(B_1,U,C_1) = (B_2,U,C_2)$ et $C_1 = C_2$ alors $B_1 = B_2$

Théorème 4 : Si XZ = YZ alors X = Y (Vrai d'après le théorème 3 dans lequel $C_1 = C_2$ = ^)

Décomposition normale

Théorème

Si un mot U est une partie d'un mot A, il existe **une et une seule** décomposition normale de A

Démonstration

Supposons 2 décompositions normales différentes de A: (B_1,U,C_1) et (B_2,U,C_2)

$A = B_1 \cup C_1 = B_2 \cup C_2$ (1)

on a : $|B_1| = |B_2|$ (2) car $(|B_1| \leq |B_2|$ et $|B_2| \leq |B_1|)$ avec $B_1 \neq B_2$ et $C_1 \neq C_2$

Montrons que c'est impossible:

(1) montre que : $|B_1| + |U| + |C_1| = |B_2| + |U| + |C_2|$

(2) (3) => $|C_1| = |C_2|$

Posons :

$B_1 = \beta^1{}_1 \beta^2{}_1 \beta^3{}_1 ... \beta^n{}_1$ et $B_2 = \beta^1{}_2 \beta^2{}_2 \beta^3{}_2 ... \beta^n{}_2$

$C_1 = \gamma^1{}_1 \gamma^2{}_1 \gamma^3{}_1 ... \gamma^r{}_1$ et $C_2 = \gamma^1{}_2 \gamma^2{}_2 \gamma^3{}_2 ... \gamma^r{}_2$

La relation (1) entraîne :

$\forall i \in [1,n], \beta^i{}_1 = \beta^i{}_2$

$\forall j \in [1,r], \gamma^j{}_1 = \gamma^j{}_2$

D'où $B_1 = B_2$ et $C_1 = C_2$

-> On a donc une seule décomposition normale

Facteur et factorisation d'un mot

Facteur d'un mot

Un mot V = **facteur d'un mot** U s'il existe 2 mots U_1 et U_2 tels que : $U = U_1VU_2$.

- si $U_1 = \wedge$ (resp. $U_2=\wedge$), V = **facteur** gauche (resp. facteur droit) de U = **préfixe** (resp. un **suffixe**) de U.
- si $V \neq \wedge$ et $V \neq U$, V = **facteur propre** de U.

Factorisation d'un mot

Factoriser un mot = l'écrire sous la forme d'un produit explicite de facteurs.

exemple :

soit l'alphabet X = {a, b} et 2 mots U = ab et V = ba.

le mot W = 'abababab a' peut être factorisé ainsi : W = UUUUa ou W = aVVVV

Occurrence d'un facteur

Une **occurrence d'un facteur** dans un mot = apparition de ce facteur à un endroit précis dans ce mot :

exemple : soit le mot U = **abababa, U** se factorise en :

- 4 occurrences du **facteur aba** (avec V = aba on a : U= VbVba = abV....= ...)

Ordres préfixiel, lexicographique

On définit dans X* plusieurs relations d'ordre :

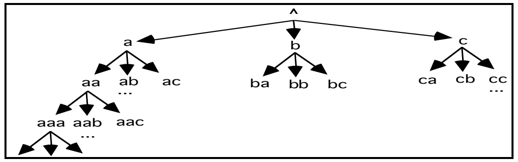
- **ordre préfixiel** = ordre <u>partiel</u> défini par U < V ssi **U est un facteur gauche de V**

(Sur la représentation arborescente, U se trouve sur la branche qui mène de la racine à V)

- **ordre lexicographique** (ordre du <u>dictionnaire</u>) = ordre <u>total</u> qui étend l'ordre préfixiel:
  - en ordonnant arbitrairement les lettres de l'alphabet X = {a, b,..., n} (par exemple a [b [c [... [n) et,
  - en posant **U [V** si et seulement si :

    soit U < V ou soit : $U = U_1y U_2$ avec $y \in X$ et $V = U_1z U_2$ avec $z \in X$ et y[z

  alors :
  - U_1 = plus long **facteur gauche** commun à U et V,
  - la première lettre qui diffère dans U et V est respectivement y et z

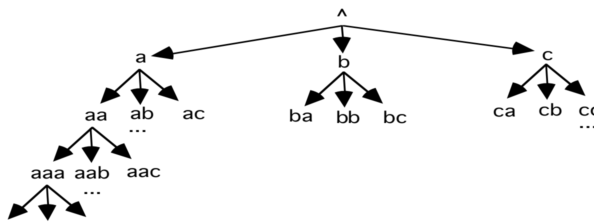


Ordre hiérarchique

- **ordre hiérarchique**
  - ordre <u>total</u>
  - les mots sont classés d'abord **par longueur**,
  - puis pour les mots de même longueur **par ordre lexicographique** (suppose que l'on a ordonné les lettres de l'alphabet).

Sur la représentation arborescente :

- un mot précède un autre s'il se trouve à un niveau plus élevé
- ou bien étant sur le même niveau s'il se trouve plus à gauche que celui-ci



Congruence de Thue ~

Relation de substitution, congruence de Thue

Soient 2 mots distincts P et Q (P ou Q pouvant être le mot vide) tel que P = une partie du mot A

Il existe au moins une décomposition (pas nécessairement normale) de A du type: **A = XPY**

On fait correspondre au mot A le mot B obtenu en substituant Q à P :

A = XPY ------- *transformation* ---------> **B** = XQY

(le remplacement de Q par P est aussi possible : transformation pas orientée)

Cette Règle de transformation = Règle de substitution = **congruence de Thue** est notée :**~** soit **P~Q**

Exemple

P = 'ara', Q ='ru', A ='parade' , B ='prude'

on a bien A = XPY = 'p'.'ara'.'de'

au mot A -> B = 'p'.'ru'.'de' = 'prude' en substituant à P une occurrence de Q :

on a alors P~Q soit ici : **'ara'~'ru'**

Remarque : appliquer une règle de Thue P~Q sur un mot A peut conduire à plusieurs transformations si A admet plusieurs décompositions ayant P (ou Q) pour second élément.

$A = B_1PC_1 \rightarrow D_1 = B_1QC_1$

$A = B_2PC_2 \rightarrow D_2 = B_2QC_2$

...

$A = B_nPC_n \rightarrow D_n = B_nQC_n$

Calcul associatif

Système de Thue

- un **système de Thue S** dans le vocabulaire X* = {de relations P~Q} définies sur X*:

S :

R_1: P_1~Q_1

R_2 : P_2~Q_2

...

R_n : P_n~Q_n

- l'application d'une ou plusieurs relations du système sur un mot A quelconque de X*, permet d'en déduire un autre mot B ∈ X*.

Calcul associatif

- **Calcul associatif** = alphabet **X** + système de Thue **S** : permet "calculs sur les mots"

Exemple S :

R_1 : 'ara' ~ 'ru'

R_2 : 'e' ~ 'is'

Mots contigus

- Soient un calcul associatif défini par alphabet X + un système de Thue S

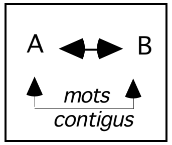
2 mots A et B sont "**contigus**" si B dérive de A par application d'**une seule** relation R_i de S

Exemple

- les mots 'paradis' et 'parade' de même que 'parade' et 'prude' sont contigus
- les mots 'paradis' et 'prude' ne sont pas contigus

Remarque

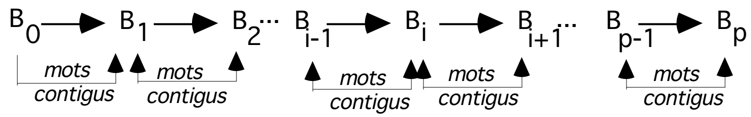
- si B dérive de A, alors A dérive de B (car relations de Thue pas orientées)



Mots en relation

- on définie une relation "**en_relation_avec**" telle que :

le mot B_0 est "**en_relation_avec**" avec le mot B_p s'il existe une suite de mots B_0, B_1,...B_p, tels que B_i soit contigu à B_{i-1} pour ∀ i ∈ [1, p].



- la relation "**en_relation_avec**" est définie ainsi :
  - tout mot "en_relation_avec" lui-même : "en_relation_avec" **reflexive**
    ex: 'parade' *en_relation_avec* 'parade'
  - la relation "en_relation_avec" est **symétrique :**
    ex: 'parade' *en_relation_avec* 'prude'
    'prude' *en_relation_avec* 'parade'
  - la relation "en_relation_avec" est **transitive :**
    ex: si 'prude' *en_relation_avec* 'parade'
    et si 'parade' *en_relation_avec* 'paradis'
    alors 'prude' *en_relation_avec* 'paradis'

=> la relation "**en_relation_avec**" est une relation **d'équivalence**

Équivalence au sens de Thue ≈

• **Théorème 1**: relation "en_relation_avec" = **relation d'équivalence** = "**équivalence au sens de Thue**" (notée **≈**)**:**

≈ réflexive : ∀ A ∈ E, A ≈ A (axiome)
≈ symétrique : ∃ A et B ∈ E tel que A ≈ B => B ≈ A (non orienté)
≈ transitive : ∀ A, B, C ∈ E : A≈B et B ≈ C =>A ≈ C

• **Théorème 2**: pour un calcul associatif donné : A ≈ B => **XAY ≈ XBY**

• **Théorème 3**: la relation d'équivalence ≈ est une **congruence** sur le monoïde libre.

En effet :

• la concaténation = LCI pour ≈ et est compatible à droite et à gauche

• si A ≈ A' et B ≈ B' par théorème (2) on a: AB ≈ A'B ≈ A'B' d'où AB ≈ A'B'

• **Théorème 4**: les classes définies sur un monoïde libre par une équivalence de Thue **forment un monoïde** pour la loi induite par la concaténation

{classes d'équivalence} sur ensemble X* = l'ensemble quotient noté X*/≈.

Il existe une application surjective de X* sur X*/≈ :

X* --f--> X*/≈ telle que si X ∈ X*, X -----> X_0 = f(x) avec X∈ C(X_0)

(X_0 est le représentant de la classe d'équivalence X = X_0).

Le problème des mots

Définition

Soit un **calcul associatif** : alphabet X+ système de Thue S (Pl~Ql, P2~Q2,...... Pn~Qn).

Thue a posé le problème fondamental suivant connu sous le nom de "**Problème des Mots**" :

soit un couple de mots (A, B),

A et B sont-ils ou non équivalents ?

Résolution

• il n'existe **pas une méthode universelle** propre à résoudre le problème des mots,

• en général le problème des mots est **indécidable (méthodes énumératives)**

• mais on peut chercher des **conditions suffisante** pour que ce problème devienne décidable:

-> pour cela trouver dans toute classe d'équivalence un **représentant canonique** = mot de **plus courte longueur** de la classe ou "**mot irréductible**"

Le problème des mots

Résolution

• **A ≈ B ? : en général problème indécidable, 2 méthodes de résolution possibles :**

1. méthode énumérative

• A ≈ B <=> A et B appartiennent à la même classe d'équivalence

=> génération de cette classe d'équivalence par exemple à partir de A :

A --> {mots contigus à A} --> {...} .... {...}

B fait-il parti de cette classe ?

2. méthode par réduction

• A ≈ B <=> A et B appartiennent à la même classe d'équivalence

d'où :

• détermination de la classe d'équiv. de A : représentant "canonique" = mot irréductible

• détermination de la classe d'équiv. de B : représentant "canonique" = mot irréductible

Si même représentant canonique <=> même classe d'équivalence <=> A ≈ B

Le problème des mots

Résolution par réduction

• soit le calcul associatif suivant :

• X ={a, b}

• S :

R_1: aaa ~ aa
R_2 : aba ~ aa
R_3 : bab ~ bb
R_4 : bbb ~ bb

• soit le mot ababa : cherchons le représentant de sa classe d'équivalence :

• ababa -R2 -> aaba - R2 -> aaa - R1 -> **aa** *mot irréductible*

• ababa - R3 -> **abba** *mot irréductible*

on a aa ~ abba et tous deux irréductibles !

Ainsi dans certain cas il y a **problème** :

une classe déquivalence peut avoir plusieurs représentants irréductibles !!!

=> **problème des mots pas résolu !!**

Résolution du problème des mots par réduction

Réduire un mot A

C'est former un nouveau mot U tel que U contigu à A et U plus court que A

Pour cela :

- constituer {mots} obtenus par **réduction successives** à partir du mot A initial
- s'arrêter quand on aboutit à un **mot irréductible** (peut être le mot vide)
- l'ensemble obtenu est une **classe d'équivalence** que l'on peut représenter par le mot irréductible

Résolution du problème des mots

le problème est **résolu** si chaque classe d'équivalence est représentée par un seul mot irréductible, en effet :

2 mots sont **équivalents** à la condition nécessaire et suffisante

d'avoir le **même équivalent irréductible.**

Résolution du problème des mots par réduction

- **Application:** Soit le calcul associatif suivant :
  - X={a,b}
  - S :
    - 'aa' ~ ∧
    - 'bb' ~ ∧

  Réduire le mot suivant : **A = abaaaabbababbbaab**

abaaaabbababbbaab

- le mot irréductible ne dépend pas de la façon dont on a conduit les opérations de réduction, il est unique : **A_R = ababa** et $A \approx A_R$
- chaque classe d'équivalence caractérisée par son mot irréductible ⇒ **le problème des mots résolu**

Montrons que A_R est unique

- supposons A conduise à 2 mots irréductibles A_R et A_R' différents. Ils sont irréductibles, donc nécessairement de la forme: --- ababab --- (1) et $|A_R| \neq |A'_R|$
- posons : $A'_R = A_R.\alpha$, (α = séquence du type (1)). Or $A_R \approx A'_R \Rightarrow A_R \approx A_R.\alpha$

impossible : il n'existe pas de relation réduisant des mots alternés du type α.

Résolution du problème des mots par réduction

Généralisation

Soit un calcul associatif défini par :

- X={x1, x2, ...xj..., xq}
- S :
  - $x^j{}_i \sim \wedge$ avec i prenant la valeur de 1 à q

*(Les relations du type **$x^j{}_i \sim \wedge$** sont dites "relations **nilpotentes**")*

un tel calcul associatif permet de représenter chaque classe d'équivalence par un seul mot irréductible ⇒ le problème des mots est donc résolu

Lemme 1 : ∀ mot A de la forme : $A = \alpha^{\beta 1}{}_{i1}\ \alpha^{\beta 2}{}_{i2} \ldots \alpha^{\beta j}{}_{ij} \ldots \alpha^{\beta k}{}_{ik} \ldots \alpha^{\beta n}{}_{in}$ avec ∀ i,j,k $\alpha_{ij} \neq \alpha_{ik}$ conduit toujours au **même mot irréductible** ∀ enchaînements des opérations de réduction

Lemme 2 : ∀ mot B de la forme : $B = \alpha^{\beta 1}{}_{i1}\ \mathrm{A}\ \alpha^{\beta p}{}_{ip}$ avec ∀ i,j $\alpha_{i1} \neq \alpha_{ip}$ conduit toujours au même mot irréductible ∀ enchaînements des opérations de réduction

Théorème: ∀ mot M conduit à un **seul mot irréductible**. En effet M est toujours le résultat de la concaténation de mots du type A et/ou B.

Exemples de résolution du problème des mots

Calcul associatif étudié par Tseitin :

soit :

X={a,b, c, d, e}

S :

- ac ~ ca
- ad ~ da
- bc ~ cb
- bd ~ db
- abac ~ abacc
- eca ~ ae
- edb ~ be

Les mots **abaacd** et **acbdad** sont-ils équivalents ??????

Non !!!

car dans chacune des règles permises de ce calcul, la partie droite et la partie gauche contiennent le même nombre d'occurrence de la lettre a (ou ne contiennent pas cette lettre).

Exemples de résolution du problème des mots

Exemple 2

soit calcul associatif C :

X={a, b, c}

S :

b ~ acc	(1)
ca ~ accc	(2)
aa ~ ^	(3)
bb ~ ^	(4)
cccc ~ ^	(5)

*Y a-t-il équivalence entre **cacb** et **bb** ?*

Remarque : C transforme tout mot R en un mot S plus court. Si aucune règle ne peut alors s'appliquer sur S, on a obtenu un mot irréductible

Idée : Construire un algorithme de réduction produisant tous les mots irréductibles.

On suppose que l'on applique les règles orientées dans l'ordre où elles sont décrites dans le système de Thue (en supprimant la règle (4)) :

a) les mots irréductibles ne contiennent pas la lettre b, d'après (1)

b) application de la règle (2) pour mettre tous les **a** avant tous les **c**

c) suppression des paires aa et quadruples cccc voisins, jusqu'à obtention de mots qui ne contiennent pas plus d'un a et pas plus de 3 c.

⇒ 8 mots irréductifs : **^; c ; cc ; ccc ; a ; ac ; acc ; accc**

d'où : 2 mots seront ≈ (équivalents) s'ils leur correspond le même mot réduit.

Exemples de résolution du problème des mots

• *Y a-t-il équivalence entre **cacb** et **bb** ?*

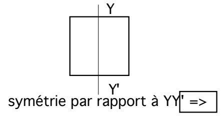
• cac**b** *-1->* cac**acc** *-2->* **ca**acccc *-2->* **accc**accccc *-2->* ac**ca**ccccccccc *-2->* ac**accc**c8 *-2->* **aa**ccc11 *-3->* c14 *-5->* c10 *-5->* c6 *-5->* **cc**

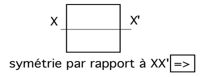
• **b**b *-1->* **acc**b *-1->* acc**acc** *-2->* ac**accc**cc *-2->* **aa**cccccccc *-3->* c8 *-5->* c4 *-5->* **^**

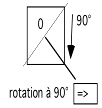
⇒ **cacb et bb ne sont pas ≈ (pas équivalents)**

• *Relation de ce problème des mots avec un problème géométrique simple.*

Considérons un carré et les 3 transformations du carré en lui-même :







On a les produits de transformation :

aa = ^
bb = ^
cccc = ^
b = acc
ca = accc

On retrouve le système de Thue du calcul associatif précédent.