UNIVERSITÉ PARIS 6

THÈSE DE DOCTORAT

Spécialité : Informatique

présentée par

Julien Cassaigne

pour obtenir le grade de Docteur de l'Université Paris 6

Sujet de la thèse :

Motifs évitables et régularités dans les mots

soutenue le 10 juin 1994 devant le jury composé de MM.

Jean-Paul Allouche	(rapporteur)
Jean Berstel	(directeur)
Maxime Crochemore	(rapporteur)
Pavel Goralčik	
Michel Mendès-France	
Jean-Éric Pin	(président)

Résumé

Nous étudions dans cette thèse les motifs évitables sur un alphabet fixé, ainsi que d'autres régularités évitables ou non. En particulier, nous présentons une classification complète des motifs binaires et un début de classification des motifs ternaires; nous prouvons par ailleurs que l'ensemble des motifs inévitables sur un alphabet donné et ayant un nombre de variables donné est toujours fini. Nous introduisons les notions de DOL-évitabilité et de HDOL-évitabilité, qui traduisent le lien entre l'évitabilité des motifs et les L-systèmes, et nous montrons que sous certaines conditions on peut décider si un HDOL-langage évite un motif. Nous abordons également deux problèmes plus combinatoires, celui du dénombrement des mots évitant un motif, dans le cas particulier des chevauchements, et celui de la construction de suites de complexité donnée.

Mots-clés : Théorie des langages, Combinatoire, Mot, Motif évitable, Décidabilité, L-système, Chevauchement, Complexité de suites, Suite régulière, Facteur spécial

Abstract

Avoidable patterns and regularities in words

In this thesis, we study the avoidability of patterns on a fixed alphabet, and other regularities in words, avoidable or unavoidable. Specifically, we give a complete classification of binary patterns and a partial classification of ternary patterns; moreover, we prove that there are always finitely many unavoidable patterns on a given alphabet with a given number of variables. We introduce DOL-avoidability and HDOL-avoidability, which reflect the link between pattern avoidability and L-systems, and we show that it is possible, with a few restrictions, to decide whether a HDOL-language avoids a pattern. We also consider two more combinatorial problems, namely counting words avoiding a pattern, in the special case of overlaps, and constructing sequences with a given subword complexity.

Keywords: Language theory, Combinatorics, Word, Avoidable pattern, Decidability, L-system, Overlap, Subword complexity, Regular sequence, Special factor

Remerciements

Je remercie tout d'abord Jean Berstel, qui, avec Michel Pocchiola, m'a initié aux langages formels et m'a encouragé à poursuivre dans cette voie, puis a encadré cette thèse avec enthousiasme, et a su me conseiller efficacement tout en me laissant travailler très librement.

J'ai trouvé au LITP une ambiance particulièrement favorable pour mener à bien ce travail, et je remercie Dominique Perrin qui m'y a accueilli, et Jean-Éric Pin, qui a par ailleurs bien voulu présider le jury.

Je remercie Jean-Paul Allouche et Maxime Crochemore d'avoir accepté d'être rapporteurs, et Pavel Goralčik et Michel Mendès-France de me faire l'honneur de participer au jury.

Je remercie Peter Roth de m'avoir accueilli à Francfort au début de mon travail de thèse. Ce séjour m'a beaucoup apporté, et j'aurais aimé continuer avec lui une collaboration qui s'annonçait fructueuse.

Je remercie les participants du groupe de travail *CIMI*, Jean Berstel, Christian Choffrut, Maxime Crochemore, Isabelle Fagnot, Pavel Goralčik et Patrice Séébold, pour les nombreuses discussions que nous avons eues ensemble, et ceux du séminaire *Automates et Théorie des Nombres* autour de Jean-Paul Allouche.

Je remercie enfin pour leur soutien mes parents et mes camarades de la rue d'Ulm et de Jussieu.

Introduction

La combinatoire des mots a déjà une longue histoire : au début de ce siècle (entre 1906 et 1914), le mathématicien norvégien Axel Thue publia une série d'articles sur les problèmes combinatoires soulevés par l'étude des suites de symboles. Deux d'entre eux [61, 62] en particulier traitaient des répétitions de facteurs dans les suites, et des moyens d'éviter de telles répétitions. Ainsi, il a montré qu'il est possible de construire une suite infinie sur un alphabet à trois lettres qui ne contienne pas de facteur répété deux fois consécutivement : on dit que c'est une suite sans carré.

Par la suite, les résultats de Thue furent plusieurs fois redécouverts, dans des buts divers. Alors que Thue avait fait cette étude « pour le développement des sciences logiques » et sans application en vue, Aršon [3] construisit une suite sans carré en 1937 pour résoudre un problème d'algèbre et Marston Morse et Gustav Hedlund [41] s'intéressèrent à ce problème dans les années 1940 en vue de l'étude des propriétés de surfaces de courbure négative. Citons également des applications aux algèbres universelles, à la théorie des groupes, à la théorie ergodique, mais aussi en physique et bien sûr en informatique où le problème de la recherche de motifs dans un texte est un sujet d'étude important.

De nombreux auteurs ont étudié les mots et les morphismes sans carré ou sans cube [1, 7, 8, 20, 22, 27, 29, 32, 33, 37, 49, 60, 64]. En 1979, Bean, Ehrenfeucht et McNulty [6] et indépendamment Zimin [65] proposèrent de généraliser l'étude de répétitions à celle de motifs arbitraires, formés de blocs de différents types disposés dans un ordre précis. Ils ont montré qu'on peut caractériser les motifs qui sont évités par un mot infini sur un alphabet fini mais non précisé; dans le cas où l'alphabet est fixé, aucune caractérisation n'est connue à ce jour. En 1989, Baker, McNulty et Taylor [5] ont poursuivi l'étude de l'évitabilité des motifs sur un alphabet fixé, en donnant entre autres un exemple de motif évitable sur un alphabet à quatre lettres mais pas sur un alphabet plus petit.

L'étude des motifs peut être vue comme un cas particulier de celle des ensembles de mots évitables : étant donné un ensemble de mots I, existe-t-il une suite infinie dont aucun facteur n'est dans cet ensemble, autrement dit qui *évite* I ? Le cas des ensembles finis est déjà riche en problèmes intéressants (voir par exemple [51]); toutefois, la décidabilité ne pose pas de problème dans ce cas, car l'ensemble des mots qui évitent I est un langage rationnel. Pour parler de décidabilité quand I est infini, il est nécessaire de se donner une description finie de I : un motif n'est autre que la description d'un ensemble infini de mots d'un certain type, et le motif est évitable si et seulement si l'ensemble qu'il décrit l'est.

Une des caractéristiques essentielles des motifs est le rôle important que jouent les morphismes, d'une part dans leur définition (l'ensemble de mots décrit par un motif p est l'ensemble des images morphiques de p), et surtout dans la construction de suites qui les évitent. Aussi, l'étude des motifs est très liée à celle des morphismes, notamment quand ceux-ci sont itérés, ce qui la rapproche de celle des L-systèmes (cf. [54]).

Dans cette thèse, nous nous proposons de poursuivre l'étude des motifs évitables sur un alphabet fixé. Le premier chapitre est consacré à une présentation des diverses notions utiles, des propriétés simples des motifs, des résultats désormais classiques de [5, 6, 65] ainsi que de quelques extensions de la notion de motif évitable.

Au chapitre 2, nous achevons la classification des motifs binaires entreprise par Peter Roth [53]. Ce chapitre permet de se rendre compte d'un certain nombre de difficultés posées par la détermination pratique de l'indice d'évitabilité d'un motif, et qui ont motivé les recherches exposées dans les chapitres suivants.

Ainsi, c'est le statut particulier des motifs binaires $ABAAB$ et $AABBA$ qui nous a amené à introduire la DOL-évitabilité et la HDOL-évitabilité, qui font l'objet du chapitre 3. Nous y étudions les motifs qui peuvent être évités par des langages engendrés par certains types de DOL-systèmes, ce qui nous permet d'ajouter un critère supplémentaire à la classification des motifs binaires.

Le chapitre 4 expose un résultat obtenu avec Peter Roth : pour un alphabet de variables donné, il n'y a qu'un nombre fini de motifs qui sont inévitables sur l'alphabet à deux lettres; cela généralise le résultat obtenu par Roth pour les motifs à deux variables.

Dans le chapitre 5, nous nous intéressons à un problème assez différent des précédents : cette fois, pour un motif bien précis (le chevauchement), nous cherchons à énumérer les mots binaires qui l'évitent en fonction de leur longueur. Nous prouvons que cette fonction d'énumération, dont on sait qu'elle a une croissance polynomiale, oscille en fait entre deux polynômes de degrés distincts. Pour cela, nous construisons un automate et des relations de récurrence explicites. Nous évaluons aussi précisément la croissance en moyenne de cette fonction.

Nous revenons ensuite aux L-systèmes dans le chapitre 6, où nous montrons comment décider si un HDOL-langage donné évite un motif, à condition que le HDOL-système vérifie certaines propriétés. Cet algorithme s'avère extrêmement utile pour classifier les motifs binaires ou autres, car il automatise des preuves souvent très longues. Il est de plus possible qu'il permette d'obtenir d'autres résultats de décidabilité concernant les motifs.

En mettant au point cet algorithme, nous avons été amené à étudier certaines propriétés des morphismes circulaires, qui ont fait l'objet de plusieurs travaux récents [40, 42] et qui semblent utiles pour de nombreux problèmes. Le chapitre 7 montre comment ils peuvent servir à résoudre un problème sur les suites de complexité ultimement affine posé par Jean-Paul Allouche.

En guise de conclusion, nous énumérons quelques questions ouvertes dont nous pensons qu'elles constituent des voies de recherche à explorer dans le domaine des motifs.

Enfin, nous donnons en annexe l'état actuel de la classification des motifs ternaires : un grand nombre de motifs 2-inévitables sont identifiés, ce qui est suffisant pour déterminer leur longueur maximale. Cependant il reste encore un certain nombre de motifs qui sont vraisemblablement 2-évitables mais pour lesquels rien n'est démontré. En ce qui concerne la 3-évitabilité, la situation est beaucoup plus favorable puisque sur les cinq motifs non traités par Silke Nilgens [47] il n'en reste plus qu'un, avec un mot infini conjecturé.

Chapitre 1

Mots et motifs

1.1 Définitions usuelles

Soit Σ un *alphabet* fini, c'est-à-dire un ensemble fini dont les éléments sont appelés *lettres* (les alphabets que nous considèrerons seront toujours finis). Le *monoïde libre* engendré par Σ, noté Σ^*, est l'ensemble des suites finies d'éléments de Σ, ou *mots*, y compris le *mot vide* ε, muni de l'opération de *concaténation* (ou simplement *produit*) qui consiste à mettre deux mots bout à bout. Ainsi, si $\Sigma = \{a, b\}$, le produit des deux mots $u = abaab$ et $v = bba$ est $uv = abaabbba$.

La *longueur* d'un mot u est le nombre de lettres qui composent ce mot, et on la note $|u|$. On a bien sûr $|\varepsilon| = 0$, $|x| = 1$ si $x \in \Sigma$, et $|uv| = |u| + |v|$ pour deux mots quelconques u et v. Pour tout mot u, on note $\mathrm{Alph}\,(u)$ l'ensemble des lettres de Σ qui apparaissent effectivement dans u.

Si u et v sont deux mots de Σ^*, on dit que u est un *facteur* de v s'il existe deux autres mots x et y tels que $v = xuy$. Si de plus x est le mot vide, le mot u est dit *préfixe* de v (*préfixe propre* si y n'est pas vide); si $y = \varepsilon$, le mot u est dit *suffixe* de v (*suffixe propre* si $x \neq \varepsilon$).

Un *langage* est un ensemble de mots, c'est-à-dire une partie (finie ou infinie) de Σ^*. On note $F(L)$ l'ensemble de tous les facteurs de tous les éléments d'un langage L; un langage tel que $F(L)$, qui a la propriété qu'un facteur d'un de ses éléments est encore un de ses éléments, est dit *langage factoriel*.

L'*image miroir* d'un mot $x = x_1x_2 \dots x_m$ est le mot, que l'on note $x^\sim$, obtenu en inversant l'ordre des lettres de x, à savoir $x^\sim = x_m x_{m-1} \dots x_1$. Les mots x tels que $x = x^\sim$ sont appelés *palindromes*.

Un *mot infini* $\mathbf{u}$ sur Σ est une suite $(u_n)_{n\in\mathbb{N}^*}$ à valeurs dans Σ, et on note Σ^ω l'ensemble des mots infinis sur Σ. Le monoïde Σ^* agit à gauche sur Σ^ω de la manière suivante: si $x = x_1x_2 \dots x_l$ est un mot de longueur l et $\mathbf{u}$ est un mot infini, alors $\mathbf{v} = x\mathbf{u}$ est défini par la suite $(v_n)_{n\in\mathbb{N}^*}$ avec $v_n = x_n$ pour $1 \leqslant n \leqslant l$ et $v_n = u_{n+l}$ pour $l < n$. On peut étendre aux mots infinis les notions de facteur, préfixe et suffixe : un mot fini u est facteur de $\mathbf{v}$ s'il existe x fini et $\mathbf{y}$ infini tels que $\mathbf{v} = xu\mathbf{y}$, c'est un préfixe si $x = \varepsilon$; un mot infini $\mathbf{u}$ est suffixe de $\mathbf{v}$ s'il existe x fini tel que $\mathbf{v} = x\mathbf{u}$. On note $F(\mathbf{u})$ l'ensemble de tous les facteurs du mot infini $\mathbf{u}$.

On peut également définir des *mots biinfinis*, qui formellement sont les orbites sous l'action du *décalage* σ d'éléments de $\Sigma^{\mathbb{Z}}$: si $x\colon n \mapsto x(n)$ est un élément de $\Sigma^{\mathbb{Z}}$, $\sigma(x)$ est défini par $\sigma(x)(n) = x(n+1)$ et représente le même mot biinfini que x. On note ${}^{\omega}\Sigma^{\omega}$ l'ensemble des mots biinfinis sur Σ.

On dit qu'une application $f\colon \Sigma^* \to \Sigma'^*$ est un *morphisme de monoïdes libres* si $f(uv) = f(u)f(v)$ pour tous mots u et v de Σ^*. Un morphisme est uniquement déterminé par l'image des lettres de Σ. Si aucune lettre n'a pour image le mot vide, on dit que le morphisme f est *non-effaçant*. Si les images de toutes les lettres sont de longueur au moins 2, on dit que f est *partout croissant*. Enfin, si toutes les images de lettres ont même longueur, on dit que f est *uniforme*.

Pour qu'un morphisme f soit injectif, il faut et il suffit d'une part que deux lettres distinctes aient des images distinctes (on dit alors que f est *injectif sur les lettres*), et d'autre part que l'ensemble $C = f(\Sigma)$ des images de lettres forme un code sur Σ', c'est-à-dire que tout mot de Σ'^* s'écrive d'au plus une manière comme produit d'éléments de C. On dira parfois que f est un code pour dire que f est injectif.

1.2 Motifs

Pour parler de motifs, nous avons besoin d'un autre alphabet, Δ. Les lettres de Δ sont appelées *variables* alors que celles de Σ sont des *constantes*. Pour les distinguer facilement, nous emploierons des minuscules pour les constantes et des majuscules pour les variables.

Un *motif sans constante*, ou simplement *motif*, est un élément du monoïde Δ^*. Un motif est donc un mot dont toutes les lettres sont des variables.

Étant donnés un mot $u \in \Sigma^*$ et un motif $p \in \Delta^*$, on dit que *u contient p* (ou contient une *occurrence* de p) s'il existe un morphisme $h\colon \Delta^* \to \Sigma^*$ non-effaçant (c'est-à-dire que l'image d'une variable n'est jamais le mot vide) tel que $h(p)$ est facteur de u. Dans le cas contraire, on dit que *u évite p*. Par exemple, le motif $AABBA$ est contenu dans les mots $abbaabb$ (avec $h(A) = b$ et $h(B) = a$) et $cbcabcababab caca$ (avec $h(A) = bca$ et $h(B) = ba$) mais est évité par $aaabaabbbaaabaaaa$. Cette définition s'applique aussi si u est un mot infini. On peut également l'étendre aux langages : on dit qu'un langage $L \subset \Sigma^*$ contient p si l'un des éléments de L contient p, et que L évite p si tous ses éléments évitent p.

1.3 Évitabilité

1.3.1 Définition

Le mot *évitabilité* regroupent plusieurs notions voisines : dans tous les cas, on dit qu'un motif est évitable s'il existe un langage infini qui l'évite, les différences résidant dans les langages autorisés.

Proposition 1.1. *Soit Σ un alphabet fini et p un motif. Les trois assertions suivantes sont équivalentes :*

(i) pour tout entier n, il existe un mot $u \in \Sigma^*$ de longueur supérieure ou égale à n tel que u évite p;

(ii) il existe un langage infini $L \in \mathscr{P}(\Sigma^*)$ qui évite p;

(iii) il existe un mot infini $\mathbf{u} \in \Sigma^\omega$ qui évite p.

Preuve : Il est clair que *(i)* et *(ii)* sont équivalentes : sur un alphabet fini, un langage est infini si et seulement s'il contient des mots arbitrairement longs.

Si *(iii)* est vraie, l'ensemble $F(\mathbf{u})$ des facteurs du mot infini $\mathbf{u}$ est un langage infini qui évite p, d'où *(ii)*. Réciproquement, si p est évité par un langage infini L, soit $\mathbf{u}$ un mot infini dans l'adhérence de L, c'est-à-dire que tout préfixe de $\mathbf{u}$ est préfixe d'une infinité d'éléments de L (un tel mot se construit aisément par récurrence); comme tous les préfixes de $\mathbf{u}$ évitent p, il en est de même de $\mathbf{u}$. ■

L'alphabet Σ étant fixé, on dit qu'un motif p est *évitable sur* Σ s'il vérifie les trois assertions équivalentes de la proposition 1.1. Dans le cas contraire, on dit naturellement que p est *inévitable sur* Σ.

Quand un motif p est évitable sur un alphabet Σ, il est évidemment évitable sur tous les alphabets de même cardinal. Si on ne veut pas préciser Σ, on dira dans ce cas que p est *k-évitable*, où k est le cardinal de Σ.

Si on ne veut pas non plus préciser k, on dira que p est *évitable*, tout court. On dira donc que p est *inévitable* s'il est inévitable sur tout alphabet fini.

1.3.2 Indice d'évitabilité

Il est clair que les différentes notions définies ci-dessus sont hiérarchisées de la manière suivante :

$$\text{2-évitable} \Longrightarrow \text{3-évitable} \Longrightarrow \text{4-évitable} \Longrightarrow \cdots \Longrightarrow \text{évitable}$$

et inversement

$$\text{inévitable} \Longrightarrow \cdots \Longrightarrow \text{4-inévitable} \Longrightarrow \text{3-inévitable} \Longrightarrow \text{2-inévitable}$$

ce qui permet de définir l'*indice d'évitabilité* d'un motif p, noté $\alpha(p)$: si p est évitable, c'est le plus petit entier k tel que p est k-évitable et $(k-1)$-inévitable; si p est inévitable, $\alpha(p) = +\infty$ par convention.

1.3.3 Exemples

Si l'alphabet Σ n'a qu'un seul élément, tous les motifs sont inévitables sur Σ, puisqu'un mot $w \in \Sigma^*$ contient le motif $p \in \Delta^*$ dès que $|w| \geqslant |p|$. On supposera désormais qu'il y a au moins deux lettres distinctes dans Σ.

Si Σ a au moins deux éléments, il existe des motifs évitables sur Σ, par exemple $ABABA$ qui est évité par le mot infini de Thue et Morse. L'indice d'évitabilité de ce motif est donc $\alpha(ABABA) = 2$.

Le motif AA est 2-inévitable, car tout mot de longueur au moins 4 sur un alphabet à deux lettres contient un carré. Nous verrons plus loin qu'il est 3-évitable : $\alpha(AA) = 3$.

Par contre, le motif ABA est k-inévitable pour tout k, car tout mot de longueur $2k + 1$ sur un alphabet à k lettres contient une lettre a répétée au moins 3 fois, donc contient un facteur de la forme axa où x est un mot non vide : ainsi ABA est un exemple de motif inévitable, $\alpha(ABA) = \infty$.

1.3.4 $\mathscr{L}$-évitabilité

Nous verrons plus loin qu'il peut être intéressant de considérer d'autres restrictions que la taille de l'alphabet. Plaçons-nous donc d'emblée dans le cadre le plus général possible : soit $\mathscr{L}$ une famille quelconque de langages; on dit qu'un motif p est *$\mathscr{L}$-évitable* s'il existe un langage infini $L \in \mathscr{L}$ qui évite p.

On retrouve les définitions précédentes pour des valeurs particulières de $\mathscr{L}$. Ainsi, un motif évitable sur Σ est $\mathscr{P}(\Sigma^*)$-évitable, un motif k-évitable est $\mathscr{P}(\Sigma_k^*)$-évitable avec $\Sigma_k = \{0, 1, 2, \ldots, k-1\}$, et un motif évitable est $\mathscr{L}$-évitable avec $\mathscr{L} = \bigcup_{k=1}^{\infty} \mathscr{P}(\Sigma_k^*)$.

On peut aussi essayer de prendre pour $\mathscr{L}$ des classes de langages usuelles. Par exemple, si $\mathscr{L} = \mathrm{RAT}(\Sigma)$ est l'ensemble des langages rationnels sur Σ, aucun motif n'est $\mathscr{L}$-évitable car d'après le lemme de l'étoile tout langage rationnel infini contient un sous-langage de la forme uv^*w (avec $v \neq \varepsilon$), donc contient comme facteur l'image de n'importe quel motif par le morphisme qui à chaque variable associe le mot v (voir la proposition 1.5). La situation est similaire avec la classe des langages algébriques. Comme nous le verrons plus loin, la théorie des L-systèmes fournit des exemples plus intéressants.

1.4 Premières propriétés

Commençons par énumérer quelques propriétés élémentaires des motifs.

1.4.1 Symétries

Quand on connaît les propriétés d'évitabilité d'un certain motif, on peut facilement, par des arguments de symétrie, déduire que plusieurs autres motifs ont les mêmes propriétés. Ainsi :

Proposition 1.2. *Si un motif p est $\mathscr{L}$-évitable, alors pour toute permutation σ de l'alphabet Δ le motif $\sigma(p)$ est également $\mathscr{L}$-évitable. En fait, p et $\sigma(p)$ sont évités par les mêmes mots.* ■

et

Proposition 1.3. *Si un motif p est $\mathscr{L}$-évitable et si la classe $\mathscr{L}$ est stable par image miroir (ce qui est le cas de toutes les classes qu'on utilisera), alors l'image miroir $p^{\sim}$ de p est également $\mathscr{L}$-évitable.* ■

1.4.2 Divisibilité

On dit qu'un motif $p \in \Delta^*$ *divise* un autre motif $q \in \Delta^*$ s'il existe un morphisme non-effaçant $h\colon \Delta^* \to \Delta^*$ tel que $h(p)$ est facteur de q; en d'autres termes, si q, vu comme un mot et non comme un motif, contient le motif p. Il est alors clair que

Proposition 1.4. *Si un motif p est $\mathscr{L}$-évitable et divise un motif q, alors ce dernier est également $\mathscr{L}$-évitable.* ■

Par exemple, sachant que AA est 3-évitable, cette proposition permet de conclure immédiatement que tous les motifs qui contiennent des carrés, par exemple $ABCBABCB$, sont également 3-évitables.

On notera que la relation de divisibilité est une relation d'ordre sur Δ^*, et que pour cet ordre l'indice d'évitabilité est une fonction décroissante de Δ^* dans $\mathbb{N} \cup \{\infty\}$

1.4.3 Langages répétitifs et motifs

Citons enfin une propriété très simple, mais qui est un argument décisif dans plusieurs preuves (nous nous en sommes d'ailleurs déjà servi en 1.3.4). On dit qu'un langage L est *répétitif* si $F(L)$ contient des puissances arbitraires, c'est-à-dire si pour tout entier n il existe un mot non vide u tel que u^n est facteur d'au moins un élément de L.

Proposition 1.5. *Si L est un langage répétitif et p un motif, alors L n'évite pas p.*

Preuve : Soit n la longueur du motif p : par hypothèse, il existe un mot non vide u tel que u^n est facteur d'un mot v de L. Soit alors h le morphisme de Δ^* dans Σ^* qui à toute variable $X \in \Delta$ associe le mot $u \in \Sigma^*$. Comme u n'est pas vide, h est non-effaçant. Le mot v contient $h(p) = u^n$ comme facteur, donc L n'évite pas p. ■

1.5 Motifs avec constantes, formules

Dans cette section nous présentons trois généralisations possibles des motifs, qui peuvent d'ailleurs se combiner à volonté.

1.5.1 Motifs avec constantes

Tout d'abord, nous pouvons ajouter des lettres constantes dans le motif : un *motif avec constantes* est un élément de $(\Delta \cup \Sigma)^*$. On dit qu'un mot w contient un motif avec constantes p s'il existe un morphisme $h : (\Delta \cup \Sigma)^* \to \Sigma^*$ non-effaçant et laissant les éléments de Σ^* inchangés tel que $h(p)$ est facteur de u.

Comme l'alphabet Σ est attaché au motif p, on ne peut plus parler de motif évitable ou k-évitable, mais seulement de motif évitable sur Σ, ce qui n'est d'ailleurs pas très intéressant : si p contient au moins une constante a, et si Σ contient au moins un autre élément b, alors p est évité par les mots b^m pour tout entier m donc est évitable sur Σ.

Les motifs avec constantes présentent toutefois un intérêt quand il s'agit de construire des mots évitant des motifs. En particulier, ils constituent un passage obligé dans notre algorithme pour décider si un DOL-langage évite un motif (cf. chapitre 6).

1.5.2 Ensembles de motifs

Il est parfois utile de dire que des motifs sont simultanément évitables. Pour cela, nous dirons qu'un mot w contient un ensemble de motifs $P \in \mathscr{P}(\Delta^*)$ s'il contient au moins l'un des éléments de P. On définit de même les ensembles de motifs évitables sur Σ, k-évitables, évitables, $\mathscr{L}$-évitables, etc.

On notera par exemple qu'un ensemble fini de motifs est évitable si et seulement si chacun de ses éléments est évitable. Cette condition est évidemment nécessaire, montrons qu'elle est suffisante : soit $P = \{p_1, \ldots, p_m\}$ un ensemble de motifs. On suppose que chacun des motifs p_i est évitable, c'est-à-dire qu'il existe un mot infini $\mathbf{u}_i = u_{i0}u_{i1}u_{i2}\ldots$ sur un alphabet Σ_i qui évite p_i. On considère alors le mot infini $\mathbf{v} = v_0 v_1 v_2 \ldots$ sur l'alphabet $\Sigma = \Sigma_1 \times \cdots \times \Sigma_m$ défini par $v_n = (u_{1n}, u_{2n}, \ldots, u_{mn})$. Il est clair que $\mathbf{v}$ évite chacun des motifs p_i, car si $\mathbf{v}$ avait un facteur de la forme $h(p_i)$, alors $\mathbf{u}_i$ aurait un facteur de la forme $(\pi \circ h)(p_i)$ où π est la projection de Σ sur Σ_i. Évidemment, l'alphabet Σ est bien plus grand que les alphabets Σ_i, donc ce raisonnement ne s'applique pas à l'évitabilité sur un alphabet fixé. Par exemple, l'ensemble de motifs $\{AAA, AABBA\}$ est 2-inévitable, bien que chacun de ses éléments soit 2-évitable, comme nous le verrons plus loin (2.4).

On remarquera que les ensembles de mots de [51] sont un cas particulier d'ensembles de motifs avec constantes.

1.5.3 Formules

Formules simples

Kirby Baker a proposé une autre extension des motifs : les formules. Baker, McNulty et Taylor considèrent dans [5] le motif

$$U_\Delta = ABWBCXCAYBAZAC$$

qui est 4-évitable mais 3-inévitable. Ils font remarquer que ce qui est important dans ce motif, ce sont les facteurs AB, BC, CA, BA et AC, et non l'ordre dans lequel ils interviennent ni ce qui les sépare. On peut exprimer cela en disant que le motif U_Δ est associé à la formule $f = AB.BC.CA.BA.AC$, où on a remplacé toutes les variables *isolées* de p (i.e. qui n'apparaissent qu'une fois) par des points.

Une *formule simple* est la donnée d'un ensemble fini de motifs sur un alphabet Δ, et on la note conventionnellement en juxtaposant les motifs qui la composent, dans un ordre quelconque, séparés par des points. En d'autres termes, l'ensemble des formules simples est le monoïde commutatif engendré par les motifs non vides, dont la loi ., que nous appellerons *superposition des formules*, vérifie $f.f = f$ pour toute formule simple f.

On dit alors qu'un mot w contient la formule simple $f = p_1.p_2.\ldots.p_m$ s'il existe un morphisme non-effaçant $h\colon \Delta^* \to \Sigma^*$ tel que les mots $h(p_1)$, $h(p_2)$, ..., $h(p_m)$ soient *tous* des facteurs de w. On peut alors parler de formule simple évitable, etc.

Il faut bien distinguer cette définition de celle, donnée dans le paragraphe précédent, d'un mot contenant un ensemble de motifs. La différence tient à un quantificateur : un mot w contient l'ensemble de motifs $\{p_1, \ldots, p_m\}$ s'il existe un morphisme non-effaçant h tel que *l'un des mots* $h(p_1)$, ..., $h(p_m)$ soit facteur de w.

L'algèbre des formules

Il est naturel de chercher à combiner ces deux notions qui semblent très proches : nous appellerons *formule* un ensemble fini de formules simples. Nous noterons désormais les ensembles de motifs de manière additive : une formule est donc une somme (finie) de formules simples. L'ensemble des formules $\mathscr{F}$, que nous appellerons *algèbre des formules* (même si ce n'est pas à proprement parler une algèbre, puisque l'addition n'est pas une loi de groupe), est muni de deux lois de composition interne commutatives et associatives $+$ et $.$ (on définit le produit de deux sommes de façon à avoir la distributivité). Nous noterons comme d'habitude 0 l'élément neutre de $+$, qui correspond à l'ensemble de motifs vide, et est évité par tous les mots, et 1 l'élément neutre de ., qui, comme le motif vide (auquel nous l'identifierons bientôt) est contenu dans tous les mots.

En cas d'ambiguïté, nous préciserons $\mathscr{F} = \mathscr{F}(\Delta, \Sigma)$ pour parler des formules à variables dans Δ avec constantes dans Σ, ou $\mathscr{F} = \mathscr{F}(\Delta) = \mathscr{F}(\Delta, \emptyset)$ pour les formules sans constante.

Équivalence de formules

L'un des intérêts majeurs des formules est qu'il existe des manières de les transformer sans changer leurs propriétés d'évitabilité. Pour cela, nous définissons une relation d'équivalence $\sim$ sur l'algèbre des formules $\mathscr{F}(\Delta, \Sigma)$ (avec $\Sigma = \emptyset$ pour les formules sans constante), appelée simplement *équivalence*, qui a la propriété de conserver la k-évitabilité.

Soit Δ' un sous-ensemble de Δ. On dit qu'une formule f divise une formule f' *en fixant* Δ' si toute formule simple de f', vue comme un mot sur l'alphabet $\Delta \cup \Sigma \cup \{.\}$, contient f, vue comme une formule à variables dans $\Delta \setminus \Delta'$ et à constantes dans $\Sigma \cup \Delta'$, avec la restriction supplémentaire que la « lettre » . ne peut pas apparaître dans l'image d'une variable.

En d'autres termes, si $f = \sum_{i=1}^{m} \prod_{p \in P_i} p$ et $f' = \sum_{j=1}^{m'} \prod_{q \in P'_j} q$, on dit que f divise f' en fixant Δ' si et seulement si pour tout $j \in [1, m']$, il existe $h\colon (\Delta \cup \Sigma)^* \to (\Delta \cup \Sigma)^*$, morphisme non-effaçant qui fixe $\Delta' \cup \Sigma$ tel que

$$\exists i \in [1, m], \forall p \in P_i, \exists q \in P'_j, h(p) \text{ est facteur de } q \ .$$

Il faut bien faire attention que, si dans une formule le signe $+$ peut être lu « ou » puisque « w contient $f_1 + f_2$ » signifie « w contient f_1 ou w contient f_2 », sa signification est inversée à droite de la relation de divisibilité : « f divise $f_1 + f_2$ » signifie « f divise f_1 et f divise f_2 ».

Si Σ' est un alphabet contenant Σ et $h\colon \Delta'^* \to \Sigma'^*$ un morphisme non-effaçant, on appelle *spécialisation de f par h* et on note $h(f)$ la formule de $\mathscr{F}(\Delta \setminus \Delta', \Sigma')$ obtenue en remplaçant dans f toute occurrence d'une variable

$X \in \Delta'$ par son image $h(X)$. Il est clair qu'un mot de Σ'^* contient f si et seulement s'il contient l'une de ses spécialisations : si $[f] = [f]_{\Sigma'}$ désigne l'ensemble des mots de Σ'^* qui contiennent une occurrence de f, on a

$$[f]_{\Sigma'} = \bigcup_{h\colon \Delta' \to \Sigma'^+} [h(f)]_{\Sigma'} \ .$$

Pour tout Δ', on définit une relation $\sim_{\Delta'}$ comme la plus petite relation d'équivalence vérifiant les propriétés suivantes, où f et f' sont deux formules :

(i) si σ est une permutation de Δ fixant Δ', alors $\sigma(f) \sim_{\Delta'} f$;

(ii) si $f = \varepsilon$ est la formule vide, $f \sim_{\Delta'} 1$.

(iii) si $f \sim_{\Delta'} f'$, alors $f + f'' \sim_{\Delta'} f' + f''$;

(iv) si $f \sim_{\Delta'} f'$, et si $(\mathrm{Alph}\,(f) \cup \mathrm{Alph}\,(f')) \cap \mathrm{Alph}\,(f'')$ est inclus dans $\Delta' \cup \Sigma$, alors $f.f'' \sim_{\Delta'} f'.f''$;

(v) si f divise f' en fixant Δ', alors $f + f' \sim_{\Delta'} f$;

(vi) si f divise f' en fixant Δ', et si $\mathrm{Alph}\,(f) \cap \mathrm{Alph}\,(f')$ est inclus dans $\Delta' \cup \Sigma$, alors $f.f' \sim_{\Delta'} f'$;

(vii) si $\Delta'' \subset \Delta'$ et $f \sim_{\Delta'} f'$, alors $f \sim_{\Delta''} f'$;

(viii) si p et p' sont des motifs et Z est une variable qui n'est ni dans Δ', ni dans $\mathrm{Alph}\,(p) \cup \mathrm{Alph}\,(p')$, alors $p.p' \sim_{\Delta'} pZp'$.

Si $f \sim_{\Delta'} f'$, on dira que f *est équivalente à* f' *en fixant* Δ'. Si $\Delta' = \emptyset$, on dit simplement que f *est équivalente à* f' et on note $f \sim f'$.

Les propriétés *(i)* à *(vii)* sont assez naturelles; toute la difficulté est de comprendre quelles variables doivent être fixées. La propriété *(viii)* est plus originale : elle traduit le fait que quand une variable n'apparaît qu'une seule fois dans un motif, elle sépare ce dernier en deux parties qui, dans une occurrence du motif, peuvent apparaître séparées par n'importe quoi, presque comme dans la formule composée de ces deux parties, sauf que pour la formule l'ordre des parties n'est pas imposé et elles peuvent même se superposer. Nous allons voir que cette différence n'est pas gênante. Montrons d'abord le lemme technique suivant :

Lemme 1.1. *Soient Δ' une partie de Δ et Σ' un alphabet contenant Σ. Soient f et f' deux formules équivalentes en fixant Δ'. Alors, pour tout morphisme non-effaçant $h\colon \Delta'^* \to \Sigma'^*$, et pour tout entier M, il existe un entier n tel que*

$$\left([h(f)]_{\Sigma'} \cap \Sigma'^{\leqslant M}\right)^n \subset [h(f')]_{\Sigma'}$$

où $\Sigma'^{\leqslant M}$ est l'ensemble des mots de Σ'^* de longueur inférieure ou égale à M.

Preuve : Il suffit de vérifier que cette assertion est compatible avec chacune des propriétés définissant $\sim_{\Delta'}$, et avec la transitivité.

Pour *(i)* et *(ii)*, il est clair que $[h(\sigma(f))] = [h(f)]$ et que $[h(\varepsilon)] = [h(1)] = \Sigma'^*$ donc l'assertion est vérifiée avec $n = 1$.

Pour *(iii)*, en supposant l'assertion vraie pour f et f', on voit qu'elle reste vraie avec la même valeur de n pour $f+f''$ et $f'+f''$. En effet, on a $[h(f + f'')] = [h(f)] \cup [h(f'')]$, donc un produit de n éléments de $[h(f + f'')]$ de longueur inférieure ou égale à M est soit dans $[h(f'')]$ (dès que l'un des éléments est dans cet ensemble), soit dans $\left([h(f)] \cap \Sigma'^{\leqslant M}\right)^n$, donc par hypothèse dans $[h(f')]$: dans les deux cas il est dans $[h(f' + f'')]$.

Pour *(iv)*, le raisonnement est similaire, la condition sur les alphabets garantissant que $[h(f.f'')] = [h(f)] \cap [h(f'')]$ et que $[h(f'.f'')] = [h(f')] \cap [h(f'')]$.

Pour *(v)*, on a $[h(f')] \subset [h(f)]$, donc $[h(f + f')] = [h(f)] \cup [h(f')] = [h(f)]$ et l'assertion est vraie avec $n = 1$.

Pour *(vi)*, on a $[h(f')] \subset [h(f)]$, donc (grâce à la condition sur les alphabets) $[h(f.f')] = [h(f)] \cap [h(f')] = [h(f')]$, et on conclut de même.

Pour *(vii)*, soit $h' \colon \Delta''^* \to \Sigma'^*$ un morphisme non-effaçant et M un entier. On a $[h'(f)] = \bigcup_{h''} [h''(h'(f))]$, où h'' décrit l'ensemble (infini) des morphismes non-effaçants de $(\Delta' \setminus \Delta'')^*$ dans Σ'^*, et de même pour f'; mais comme on s'intéresse à l'ensemble fini $[h'(f)] \cap \Sigma'^{\leqslant M}$, il suffit de considérer un ensemble fini $H = \{h''_1, \dots, h''_q\}$ de morphismes pour avoir

$$[h'(f)] \cap \Sigma'^{\leqslant M} = \bigcup_{h'' \in H} [h''(h'(f))] \cap \Sigma'^{\leqslant M} \quad .$$

À chacun de ces morphismes $h''_i \in H$, on associe un morphisme non-effaçant $h_i \colon \Delta'^* \to \Sigma'^*$ tel que $h_i(f) = h''_i(h'(f))$ et $h_i(f') = h''_i(h'(f'))$. On construit par récurrence des suites (M_i) et (n_i) pour $1 \leqslant i \leqslant q$ comme suit : $M_1 = M$; pour tout i, par hypothèse, il existe un entier n_i tel que l'assertion concernant $\sim_{\Delta'}$ soit vérifiée pour h_i et M_i. On pose alors $M_{i+1} = n_i M_i$, et on vérifie que

$$\left(\bigcup_{1 \leqslant j \leqslant i} [h_j(f)] \cap \Sigma'^{\leqslant M}\right)^{M_{i+1}/M} \subset \bigcup_{1 \leqslant j \leqslant i} \left([h_j(f)] \cap \Sigma'^{\leqslant M_j}\right)^{n_j} \quad .$$

Soit alors $n = \prod_{1 \leqslant i \leqslant q} n_i$. Tout mot de $\left([h'(f)] \cap \Sigma'^{\leqslant M}\right)^n$ est dans au moins un des ensembles $\left([h_i(f)] \cap \Sigma'^{\leqslant M_i}\right)^{n_i}$, donc dans $[h''_i(h'(f'))]$, qui est inclus dans $[h'(f')]$.

Pour *(viii)*, l'assertion est clairement vérifiée si on prend $f = pZp'$ et $f' = p.p'$ avec $n = 1$, puisque $[pZp'] \subset [p.p']$. Dans l'autre sens, c'est-à-dire pour $f = p.p'$ et $f' = pZp'$, soit $n = 1 + 2\#\left([p.p'] \cap \Sigma'^{\leqslant M}\right)$ ($\#E$ désigne le cardinal d'un ensemble E). Parmi n éléments de $[p.p'] \cap \Sigma'^{\leqslant M}$, au moins deux non consécutifs seront donc égaux. Tout élément de $\left([h(p.p')] \cap \Sigma'^{\leqslant M}\right)^n$ contient donc un facteur de la forme uvu, où $u \in [h(p.p')]$ et $v \in \Sigma'^{+}$, donc contient une occurrence de $h(pZp')$, ce qui permet de conclure.

Enfin, la transitivité est conservée : supposons l'assertion vraie pour les formules f et f', et pour les formules f' et f''. Étant donnés h et M, on obtient un entier n pour f et f'. Soit alors M' la longueur maximale des mots de $\left([h(f)] \cap \Sigma'^{\leqslant M}\right)^n$. Appliquée à h et M', l'assertion pour f' et f'' donne un entier n', et on a

$$\left([h(f)] \cap \Sigma'^{\leqslant M}\right)^{n''} \subset [h(f'')]$$

avec $n'' = nn'$. ■

Nous pouvons maintenant démontrer le résultat principal :

Théorème 1.1. *Soient f et f' deux formules équivalentes, et Σ' un alphabet. Si f' est évitable sur Σ', alors f est aussi évitable sur Σ'.*

Preuve : Supposons que f est inévitable sur Σ'. D'après le lemme 1.1, appliqué à $\Delta' = \emptyset$, pour tout entier M il existe un entier n tel que $\left([f] \cap \Sigma'^{\leqslant M}\right)^n \subset [f']$. Puisque f est inévitable sur Σ', on peut choisir l'entier M suffisamment grand pour que tout mot de longueur M contienne au moins une occurrence de f. Tout mot de longueur nM est donc dans $\left([f] \cap \Sigma'^{\leqslant M}\right)^n$, donc contient f', ce qui veut dire que f' est inévitable sur Σ'. ■

Ce théorème ne s'applique pas à la $\mathscr{L}$-évitabilité pour une classe de langages quelconque, car il n'est pas possible de borner la longueur des occurrences de f. Toutefois, on peut l'étendre au cas des classes de langages *uniformément récurrents*, c'est-à-dire aux classes $\mathscr{L}$ telles que pour tout langage $L \in \mathscr{L}$ et pour tout mot $u \in L$, il existe un entier M tel que tout élément de longueur M de $F(L)$ contient u comme facteur.

Théorème 1.2. *Soient f et f' deux formules équivalentes, et $\mathscr{L}$ une classe de langages uniformément récurrents. Si f' est $\mathscr{L}$-évitable, alors f est aussi $\mathscr{L}$-évitable.*

Preuve : Supposons que f est $\mathscr{L}$-inévitable. Soit $L \in \mathscr{L}$ un langage infini : il existe donc un mot $u \in L$ qui contient f. Comme L est uniformément récurrent, il existe donc un entier M tel que tout élément de longueur M de $F(L)$ contient u comme facteur, et donc contient f. D'après le lemme 1.1, il existe un entier n tel que $\left([f] \cap \Sigma'^{\leqslant M}\right)^n \subset [f']$. Tout élément de $F(L)$ de longueur nM contient donc f', donc L n'évite pas f'. ■

1.5.4 Répétitions

Les répétitions sont une autre forme de régularité, qui a été largement étudiée [21, 43, 48]. On dit qu'un mot w contient une répétition d'exposant α s'il existe deux facteurs non vides u et p de w tels que p est une période de u (c'est-à-dire que u est facteur d'une puissance de p) et que $\frac{|u|}{|p|} = \alpha$.

Le cas le plus simple et celui des exposants entiers, et il s'exprime aisément en termes de motifs : un mot contient une répétition d'exposant n si et seulement il contient le motif A^n.

Un autre cas qui se ramène aux motifs et celui des *chevauchements* : on dit qu'un mot contient un chevauchement s'il contient une répétition d'exposant strictement plus grand que 2. Or une répétition d'exposant strictement compris entre 2 et 3 est de la forme $u = xyxyx$, où x et y sont deux mots non vides, donc un mot contient un chevauchement si et seulement s'il contient l'un des motifs AAA et $ABABA$, c'est-à-dire s'il contient la formule $AAA + ABABA$.

Le cas des exposants fractionnaires sort du cadre des motifs. Cependant, certains résultats sur les répétitions peuvent être utiles à l'étude des motifs. Par exemple, tout mot contenant le motif $ABCBABC$ contient une répétition d'exposant strictement supérieur à $\frac{3}{2}$: en effet, un tel mot contient un facteur de la forme $xyzyxyz$ où x, y et z sont des mots non vides, c'est-à-dire une répétition de période $xyzy$ et d'exposant $\frac{|xyzyxyz|}{|xyzy|} = \frac{3}{2} + \frac{|xz|}{2|xyzy|} > \frac{3}{2}$. Comme Pansiot [48] a montré qu'il existe un mot infini sur un alphabet à 4 lettres sans répétition d'exposant strictement supérieur à $\frac{7}{5}$, on en déduit que $ABCBABC$ est 4-évitable (voir aussi [18]).

1.6 Variantes

Nous mentionnons ici quelques façons de modifier la définition de l'évitabilité (ou k-évitabilité, $\mathscr{L}$-évitabilité, etc.) d'un motif. Nous n'avons pas étudié plus en détail ces variantes, mais il est possible que les méthodes développées dans cette thèse conduisent à des résultats intéressants les concernant.

1.6.1 Évitabilité commutative

On dit qu'un mot $u \in \Sigma^*$ *contient commutativement* un motif $p \in \Delta^*$, que l'on notera $p = X_1X_2\dots X_m$ où les X_i sont des variables, s'il existe m mots u_1, u_2, ..., u_m non vides tels que $u_1u_2\dots u_m$ est facteur de u et que, pour tout couple (i, j) tel que $X_i = X_j$, les mots u_i et u_j ne diffèrent que par une permutation de leurs lettres. Par exemple, le mot $bbbaaababbbaaa$ contient commutativement $AABBA$, avec $u_1 = baa$, $u_2 = aba$, $u_3 = b$, $u_4 = b$, $u_5 = baa$. On définit ensuite les motifs *commutativement k-évitables*, etc. de manière habituelle, en remplaçant *contient* par *contient commutativement.*

Il est clair que si u contient p au sens habituel, alors il contient aussi p commutativement. Tout motif k-évitable commutativement est donc aussi k-évitable, mais la réciproque n'est pas vraie (par exemple AA n'est pas commutativement 3-évitable). Pour cette raison, James Currie [17] appelle ces motifs *fortement k-évitables*.

L'origine de cette définition est le problème des *carrés abéliens*, posé par Erdös [28, p. 240] : le motif AA est-il commutativement 4-évitable ? La réponse positive à ce problème a finalement été donnée par Keränen [34] au moyen d'un morphisme uniforme de longueur 85 (voir aussi [23, 27, 49]).

Currie pose également la question suivante : existe-t-il un algorithme pour décider si un motif donné est commutativement évitable (sur un alphabet arbitraire), de la même manière que l'algorithme de Zimin permet de décider si un motif est évitable. Or il se trouve que les seuls motifs non commutativement évitables que nous connaissions sont les motifs inévitables. Il serait donc intéressant d'essayer d'en construire d'autres; s'il s'avérait que ce n'est pas possible, l'algorithme de Zimin permettrait donc de décider également l'évitabilité commutative.

1.6.2 l-occurrences

Entringer, Jackson et Schatz [27] ont construit une suite binaire dont les seuls carrés sont de longueur inférieure ou égale à 4. De manière générale, on peut se demander, quand un motif est k-inévitable, si cela est dû simplement au fait qu'un certain nombre d'images du motif forment un ensemble inévitable de mots, ou s'il y a une raison plus globale. Il s'agit donc d'affaiblir la notion de k-évitabilité en ne considérant que les occurrences suffisamment longues du motif.

Quand l'alphabet des variables a deux éléments ou plus, il y a deux manières de procéder. La première consiste à ignorer toutes les occurrences $h(p)$ du motif p dont la longueur totale est inférieure à un entier l donné; ceci n'a à notre connaissance jamais été étudié. La seconde, encore plus tolérante, consiste à ne considérer que les morphismes h tels que l'image de toute variable soit de longueur au moins l. Dans ces conditions, Peter Roth [52] appelle le mot $h(p)$ une l-occurrence de p. On notera que des images de p arbitrairement longues peuvent être éliminées de cette manière, pour peu que l'image de l'une des variables soit courte.

On peut donc appeler *faiblement k-évitable* un motif p pour lequel il existe un entier l et un mot infini sur l'alphabet à k lettres sans l-occurrence de p. Peter Roth a montré que tout motif évitable était faiblement k-évitable pour tout $k \geqslant 2$, $k = 2$ en particulier, ce qui fournit une nouvelle caractérisation des motifs évitables. D'autres questions restent ouvertes : par exemple, pour un motif p donné, quelle est la plus petite valeur de l pour laquelle les l-occurrences de p sont 2-évitables ?

Dans les deux sections précédentes, nous avons décrit un certain nombre d'extensions possibles. Toutefois, il semble que les problèmes essentiels se posent déjà dans le cas le plus simple : la k-évitabilité des motifs ordinaires. Dans les chapitres suivants, nous nous y intéresserons en priorité, mais parfois il sera plus avantageux pour cela de travailler dans le cadre de l'une des généralisations.

1.7 Évitabilité sur un alphabet non fixé

Le problème de l'évitabilité sur un alphabet non fixé a été étudié en détail par Zimin [65] et Bean, Ehrenfeucht et McNulty [6]. En particulier, ils ont montré qu'on pouvait décider si un motif donné est évitable ou non. Nous présentons ici cet algorithme en termes de graphes d'adjacence, d'après [5].

1.7.1 Quelques propriétés

La proposition suivante illustre ce que peut apporter le fait que la taille de l'alphabet n'est pas fixée :

Proposition 1.6. *Un ensemble fini de motifs est évitable si et seulement si chacun de ses éléments pris individuellement est évitable.*

Preuve : Soit $P = \{p_1, \ldots, p_m\}$ un ensemble fini de motifs. Si l'un des p_i est inévitable, il est clair que P est également inévitable. Supposons maintenant que tous les p_i sont évitables : il existe donc des alphabets Σ_1, ..., Σ_m (ici Σ_i ne désigne pas l'alphabet à i éléments !) et des mots infinis $\mathbf{u}_1 \in \Sigma_1^\omega$, ..., $\mathbf{u}_m \in \Sigma_m^\omega$ tels que $\mathbf{u}_i$ évite p_i. On considère alors le mot infini $\mathbf{u} \in \Sigma^\omega$, avec $\Sigma = \Sigma_1 \times \cdots \times \Sigma_m$, tel que la projection sur ${\Sigma_i}^\omega$ de $\mathbf{u}$ soit égale à $\mathbf{u}_i$: ce mot infini évite simultanément tous les p_i, donc évite P. ∎

La proposition qui suit est une conséquence immédiate de l'algorithme exposé plus bas :

Proposition 1.7. *Tout motif inévitable contient une variable isolée (c'est-à-dire une variable qui n'apparaît qu'une seule fois).* ∎

Elle permet de montrer que

Proposition 1.8. *Un motif inévitable comprenant n variables différentes est de longueur au plus $2^n - 1$.*

Preuve : La preuve est par récurrence sur n. Le seul motif inévitable formé avec une seule variable est A, de longueur $1 = 2^1 - 1$. Supposons maintenant la proposition démontrée pour les alphabets de variables de taille inférieure à n.

Soit p un motif comprenant n variables différentes. D'après la proposition 1.7, l'une de ses variables X n'apparaît qu'une fois. On a alors $p = p_1 X p_2$, où p_1 et p_2 sont des motifs inévitables qui contiennent au plus $n-1$ variables différentes. Par hypothèse de récurrence, $|p_1| \leqslant 2^{n-1}-1$ et $|p_2| \leqslant 2^{n-1}-1$, donc $|p| = |p_1| + 1 + |p_2| \leqslant 2^n - 1$. ∎

Cette borne $2^n - 1$ est optimale, car on peut construire des motifs de cette longueur par le procédé suivant : soit $Z_1 = A_1$ et pour tout $n > 1$, $Z_n = Z_{n-1} A_n Z_{n-1}$ (les Z_n sont parfois appelés *sesquipuissances*, ou *motifs de Zimin*, et ont été introduits par Aršon [3]). Il est clair que l'alphabet des variables de Z_n a n éléments, et que la longueur de Z_n est $2^n - 1$.

Proposition 1.9. *Les motifs Z_n sont tous inévitables.*

Preuve : Cette propriété est évidente quand on la voit du point de vue des formules : en effet, $Z_n = Z_{n-1} A_n Z_{n-1}$ est équivalent à $Z_{n-1}.Z_{n-1}$, puisque la variable A_n est isolée. Mais cette dernière formule est elle-même égale à Z_{n-1}. En itérant cette opération, on voit que tous les Z_n sont équivalents, au sens des formules, à Z_1, et même au motif vide, qui est bien entendu inévitable. ∎

Enfin, en combinant les propositions 1.6 et 1.8, on obtient le résultat surprenant suivant :

Proposition 1.10. *Pour un alphabet de variables Δ donné, il existe un alphabet Σ et un mot infini sur Σ qui évite tous les motifs évitables de Δ^*.*

Preuve : L'ensemble des motifs évitables de Δ^* est infini : pour pouvoir appliquer la proposition 1.6, il faut se ramener à un ensemble fini. On se limite donc aux motifs évitables de longueur au plus 2^n, où n est le cardinal de Δ, et on peut alors construire un mot infini $\mathbf{u}$ qui les évite tous simultanément au moyen de la proposition 1.6. Si p est un motif évitable de longueur supérieure à 2^n, soit q un facteur de longueur 2^n de p : d'après la proposition 1.8, q est évitable, donc est évité par $\mathbf{u}$, donc p est aussi évité par $\mathbf{u}$: on a montré que $\mathbf{u}$ évite tous les motifs évitables. ∎

1.7.2 L'algorithme de Zimin-Bean-Ehrenfeucht-McNulty

Passons maintenant au résultat principal de Zimin [65] et Bean, Ehrenfeucht, McNulty [6] :

Théorème 1.3. *Un motif est évitable si et seulement s'il est irréductible.*

L'*irréductibilité* d'un motif, définie ci-dessous à la manière de [5], est une propriété décidable.

Soit $p \in \Delta^*$ un motif. On appelle *graphe d'adjacence* de p le graphe biparti $AG(p)$, dont les sommets sont étiquetés par deux copies de Δ, que nous noterons Δ^L et Δ^R quand nous aurons besoin de les distinguer, et dont les arêtes sont les (X^L, Y^R) pour tout XY facteur de p. Ainsi, le graphe d'adjacence de $ABACBA$ (figure 1.1) comporte 6 sommets et 4 arêtes.

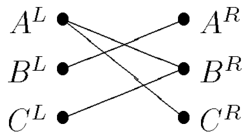

FIG. 1.1 - *Le graphe d'adjacence de* $ABACBA$

Une partie non vide F de Δ est dite *libre* (pour p) s'il n'existe pas de chemin dans $AG(p)$ allant d'un sommet de gauche X^L à un sommet de droite Y^R avec X et Y dans F. Pour déterminer les ensembles libres, il est donc utile de repérer les composantes connexes de $AG(p)$. Dans l'exemple ci-dessus, il y a deux composantes connexes et les ensembles libres sont $\{A\}$ et $\{B\}$.

On dit que le motif *p se réduit en p' en une étape* s'il existe un ensemble de variables F libre pour p tel que p' soit obtenu en effaçant dans p les variables de F. On dit que p se réduit en p' s'il existe une suite de réductions en une étape partant de p et arrivant à p'. Enfin, on dit que p est *réductible* s'il existe une réduction de p au motif vide, et *irréductible* sinon.

Dans l'exemple ci-dessus, $p = ABACBA$ se réduit en BCB si on efface l'ensemble libre $\{A\}$, et BCB lui-même se réduit en C puis en ε. Le motif p est donc réductible. Toutefois, si on avait commencé par effacer l'ensemble libre $\{B\}$, on aurait obtenu le motif irréductible $AACA$: pour s'assurer qu'un motif est irréductible, il faut donc explorer récursivement toutes les possibilités de réduction et vérifier qu'aucune ne conduit au motif vide, ce qui donne l'algorithme de la figure 1.2, que nous appellerons pour simplifier *algorithme de Zimin*.

Il convient de noter qu'il est parfois indispensable d'effacer des ensembles libres autres que des singletons, par exemple pour le motif

$$p = ABACA'BADABA'CA'BA'$$

dû à Bean, Ehrenfeucht et McNulty, pour lequel l'ensemble à effacer est $\{A, A'\}$.

1.8 Motifs verrouillés

Certains motifs ont un comportement particulièrement simple pour l'algorithme exposé ci-dessus : il s'agit des motifs dont le graphe d'adjacence est

Algorithme *réduction*
Entrée : un motif p
Sortie : un booléen, indique si p est réductible

Début
    Calculer le graphe d'adjacence de p
    Calculer l'ensemble $\mathscr{F}$ des ensembles libres pour p
    Pour tout $F \in \mathscr{F}$:
        Calculer p', motif obtenu en effaçant les variables de F dans p
        Appeler *réduction* avec pour entrée p'
        Si la réponse est *vrai*, rendre *vrai*
    Rendre *faux*
Fin.

FIG. 1.2 - *L'algorithme de Zimin*

connexe, et pour lesquels aucune réduction n'est donc possible. De tels motifs sont dits *verrouillés*. Un motif p est verrouillé si et seulement si, quelles que soient les variables X et Y qui apparaissent effectivement dans p, il existe un entier $m \geqslant 1$ et $2m$ variables X_1, ..., X_m, Y_1, ..., Y_m telles que $X_1 = X$, $Y_m = Y$, et que les $2m-1$ motifs X_iY_i (pour $1 \leqslant i \leqslant m$) et $X_{i+1}Y_i$ (pour $1 \leqslant i < m$) soient facteurs de p.

Baker, McNulty et Taylor [5] ont montré que les motifs verrouillés sont tous 4-évitables. Plus précisément :

Théorème 1.4. *Les motifs verrouillés sont tous évités par le mot infini $\Omega_{2,1}$, point fixe du morphisme suivant :*

$$\begin{array}{rrcl} h_{2,1}: & \{a,b,c,d\} & \longrightarrow & \{a,b,c,d\} \quad . \\ & a & \longmapsto & ab \\ & b & \longmapsto & cb \\ & c & \longmapsto & ad \\ & d & \longmapsto & cd \end{array}$$

■

Le mot $\Omega_{2,1}$ a été introduit par Dean [20]. C'est ainsi que Baker, McNulty et Taylor prouvent que le motif $U_\Delta = ABWBCXCAYBAZAC$ est 4-évitable, puisqu'il est verrouillé. Pour ce motif, il est plus naturel de parler de la formule équivalente $f_\Delta = AB.BC.CA.BA.AC$, qui n'a que trois variables distinctes. On dira encore que cette formule est verrouillée : on peut en effet construire le graphe d'adjacence d'une formule simple, avec une arête entre X^L et Y^R

quand XY est facteur de l'un des éléments de la formule. Pour la formule f_Δ, on obtient le graphe d'adjacence de la figure 1.3.

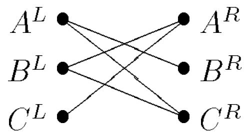

FIG. 1.3 - *Le graphe d'adjacence de f_Δ*

On a alors les propriétés suivantes :

Proposition 1.11. *Si le motif $p = p_0 Y_1 p_1 \ldots p_{m-1} Y_m p_m$, où les variables Y_i sont isolées, est verrouillé, alors la formule équivalente $f = p_0.p_1.\ldots.p_{m-1}.p_m$ est verrouillée.*

Preuve : Le graphe d'adjacence de p diffère de celui de f par les sommets Y_i^L et Y_i^R; de chacun de ces sommets part au plus une arête, donc cela ne peut pas faire diminuer le nombre de composantes connexes du graphe. Cela peut par contre en ajouter : ainsi $p = AABCA$ n'est pas verrouillé alors que $f = AA.A$ est verrouillée. ■

Proposition 1.12. *Un motif p est verrouillé si et seulement si la formule simple f dont les éléments sont les facteurs de longueur 2 de p (dite* 2-hachage *de p) est verrouillée.*

Preuve : Cette proposition découle immédiatement du fait que le graphe d'adjacence de p dépend uniquement de l'ensemble des facteurs de longueur 2 de p. ■

Plus généralement, on appelle *k-hachage* d'une formule simple f la formule simple dont les éléments sont les facteurs de longueur inférieure ou égale à k des composantes de f, dans laquelle on supprime les éléments qui sont facteurs d'autres éléments plus grands. Pour $k = 2$ et si f est un motif, cette définition coïncide avec celle de la proposition 1.12, sauf pour le motif réduit à une variable qui de toutes manières n'est pas verrouillé. Par exemple, le 3-hachage de $ABCB.BC.BA$ est $ABC.BCB.BA$. Le k-hachage d'une formule divise toujours cette formule, donc est *a priori* plus difficile à éviter. Il est donc raisonnable, si on cherche par exemple à construire des motifs évitables qui ne soient pas 4-évitables (l'existence de tels motifs est un problème ouvert, cf [5]), de considérer des k-hachages pour un entier k petit. Toutefois, $k = 2$ ne convient pas :

Proposition 1.13. *Tout 2-hachage est soit 4-évitable, soit inévitable.*

Preuve : Soit f un 2-hachage; si f a des composantes de longueur 1, il s'agit nécessairement de variables isolées, donc si on les supprime on obtient une formule équivalente. La formule f est donc entièrement caractérisée par son graphe d'adjacence. S'il existe une restriction de $AG(f)$ qui soit connexe (on ne considère que les restrictions à des ensembles de sommets du type ${\Delta'}^L \bigcup {\Delta'}^R$ où Δ' est une partie de l'alphabet des variables), alors f est divisible par une formule verrouillée, donc est 4-évitable. Si ce n'est pas le cas, il y a au moins un ensemble libre. On applique alors l'algorithme de Zimin à f, sachant que le graphe d'adjacence obtenu après effacement d'un ensemble libre F est tout simplement la restriction au complémentaire de F du graphe de départ : ce graphe a également la propriété qu'aucune de ses restrictions n'est connexe, donc on peut recommencer et éliminer des ensembles libres dans n'importe quel ordre, jusqu'à éliminer tous les sommets du graphe : f est donc inévitable. ■

Pour $k \geqslant 3$, il n'est plus possible d'utiliser le même genre d'arguments, car l'élimination d'un ensemble libre peut faire apparaître de nouvelles arêtes dans le graphe, et que toutes les suites de réductions ne sont pas nécessairement équivalentes.

Chapitre 2

Classification des motifs binaires

Dans ce chapitre, nous nous proposons de déterminer l'indice d'évitabilité de tous les motifs binaires, c'est-à-dire à variables dans $\Delta = \{A, B\}$. Cette étude a été commencée par Ursula Schmidt [56], qui a montré que seul un nombre fini d'entre eux avaient un indice strictement supérieur à 2 (plus précisément, elle a montré que tous les motifs binaires de longueur 13 ou plus sont d'indice 2). Peter Roth [52] l'a presque achevée, et nous avons montré que le seul motif encore manquant (à symétrie près), $AABBA$, était 2-évitable [15]. Ce résultat a aussi été obtenu indépendamment par T. Vaniček [31, 63].

2.1 Les motifs inévitables

Zimin a montré que les motifs inévitables à n variables étaient de longueur au plus $2^n - 1$. Il suffit donc d'examiner un à un les motifs binaires de longueur au plus 3, à savoir (à permutation de A et B près) : ε, A, AA, AB, AAA, AAB, ABA, ABB. On peut appliquer l'algorithme de Zimin à chacun d'eux, ou noter que AA est évitable et qu'il en est de même par conséquent de tous les motifs qui le contiennent. Les motifs restants : ε, A, B, AB, BA, ABA, BAB sont inévitables, leur indice d'évitabilité est donc infini.

2.2 Motifs 2-inévitables

2.2.1 L'algorithme de *backtracking*

Pour prouver qu'un motif p est 2-inévitable, il suffit d'énumérer tous les mots binaires qui évitent p au moyen de l'algorithme de la figure 2.1, dit de *backtracking*. La 2-inévitabilité est donc semidécidable.

Il consiste à faire un parcours en profondeur d'abord de l'arbre des mots binaires évitant p, qui est fini si et seulement si le motif p est 2-inévitable. Quand on arrive à l'extrémité d'une branche, c'est-à-dire à un mot qui ne peut

```
Algorithme backtracking
Entrée : un motif p
Sortie : si p est 2-inévitable, la liste des mots binaires qui évitent p
         sinon, l'algorithme ne termine pas
Variable : un mot w

Début
    w ← ε
    Répéter :
        Tant que w contient p :
            Tant que la dernière lettre de w est un b :
                Effacer la dernière lettre de w
            Si w = ε, terminer
            Remplacer la dernière lettre de w par un b
        Afficher w
        w ← wa
Fin.
```

FIG. 2.1 - *L'algorithme de* backtracking

être prolongé d'une lettre à droite sans faire apparaître le motif p, on revient en arrière (d'où le nom de l'algorithme) jusqu'à trouver une branche qui n'a pas encore été explorée.

Dans cet algorithme, on a besoin de tester un grand nombre de fois si un mot w contient le motif p. Pour cela, on peut utiliser l'algorithme naïf, en $O\left(|w|^{\#\Delta+2}\right)$, ou plutôt $O\left(|w|^{\#\Delta+1}\right)$ car, comme on sait déjà que les préfixes propres de w évitent p, il suffit de chercher des occurrences de p qui soient des suffixes de w. On peut aussi chercher des algorithmes plus rapides; de tels algorithmes sont connus pour les carrés, les chevauchements et les motifs binaires généraux [16, 35, 46], et nous n'avons pas approfondi cette question.

En pratique, soit le *backtracking* se termine en un temps raisonnable et on a prouvé que p est 2-inévitable, soit il semble diverger et il est probable que p soit 2-évitable : il permet donc, la plupart du temps, de se faire une idée intuitive du comportement de p.

2.2.2 Résultats

Les résultats, à symétrie près, sont résumés dans le tableau 2.2 [53], où w est un mot binaire évitant p de longueur maximale, et N est le nombre total de mots binaires (y compris le mot vide) qui évitent p.

p	w	$\|w\|$	N
A	ε	0	1
AB	a	1	3
AA	aba	3	7
ABA	$aabb$	4	13
AAB	$abab$	4	15
$AABA$	$abababaaa$	9	91
$ABBA$	$aabbbaaabb$	10	93
$AABB$	$abaaabaaaba$	11	147
$ABAB$	$abaabbaaabbbaabbab$	18	477
$AABAA$	$abaaaabbbbabababab$	18	1699
$AABAB$	$abababababbaaabbbaaaabbbbaaabbbaabbab$	38	26241

TAB. 2.2 - *Motifs binaires 2-inévitables*

2.2.3 Accélération du *backtracking*

L'énumération de tous les mots binaires évitant p peut parfois durer très longtemps, et il est utile de savoir se passer d'une partie de cette énumération. On peut alors vérifier plus rapidement qu'un motif est 2-inévitable, mais sans pouvoir exhiber w et N.

Une première idée est de chercher d'abord tous les mots évitant p d'une longueur fixée l_0 et de construire leur graphe de Rauzy (voir 7.3.1). Tout mot évitant p de longueur $l \geqslant l_0$ sera associé à un chemin de longueur $l - l_0$ dans ce graphe, et il suffit donc d'énumérer ces chemins. Si on se contente de cela, on ne perd pas de mots évitant p mais on diminue notablement le nombre de mots n'évitant pas p à tester. Mais on peut aller plus loin : si, pendant le déroulement de l'algorithme, on revient en arrière en-deçà de la position l_0, cela veut dire qu'on a trouvé un mot u de longueur l_0 qui n'est préfixe que d'un nombre fini de mots évitant p, et on est donc assuré que tout mot contenant u n'est pas prolongeable à droite en un mot infini évitant p; on peut alors supprimer u du graphe, ainsi que les mots dont u est le seul prolongement à droite.

Supposons par exemple qu'on veuille vérifier que $AABB$ est 2-inévitable. On prend $l_0 = 4$: le graphe a alors douze sommets (tous les mots de longueur 4 sauf $aaaa$, $aabb$, $bbaa$ et $bbbb$) et dix-huit arêtes. On commence ensuite l'énumération, qui débute par les préfixes successifs de $aaabaaaba$, puis revient à $aaabaab$, puis à aab: on peut alors supprimer $aaab$ du graphe, et du même coup $baaa$ puisque son seul prolongement possible était $aaab$; par symétrie on peut aussi supprimer $bbba$ et $abbb$. On poursuit alors l'algorithme avec un graphe réduit à huit sommets et douze arêtes, et au bout d'un certain temps on revient à ab: on supprime alors $aaba$, son symétrique $bbab$ et tous les mots du graphe qui perdent ainsi leur unique prolongement, et il ne reste plus que deux sommets ($abab$ et $baba$) et 2 arêtes. L'algorithme se termine alors très vite. En tout, on a énuméré 26 mots, de longueur au plus 9, au lieu de 147 mots de longueur au plus 11.

L'amélioration suivante part d'une constatation : le *backtracking* ajoute une orientation artificielle aux mots, alors que la notion de motif 2-inévitable est tout à fait symétrique. Par exemple, il est possible que l'accélération ci-dessus soit plus efficace avec un motif qu'avec son image miroir : pour le motif $AABA$, avec $l_0 = 3$, on énumère 15 mots alors que pour $ABAA$ on en énumère 34. Or, si un motif est 2-évitable, il existe un mot biinfini qui l'évite, et donc une infinité de mots infiniment biprolongeables. On peut donc éliminer également tout mot qui n'est pas infiniment prolongeable à gauche.

En pratique, on peut procéder ainsi : on construit un graphe qui à un instant donné contient tous les mots de longueur au plus l évitant le motif susceptibles d'être infiniment biprolongeables, et avec des arêtes de deux types : des arêtes indiquant un prolongement à droite de u à ua et de u à ub pour tout u (à condition bien sûr que ua et ub soient dans le graphe) et des arêtes indiquant un prolongement à gauche de u à au et de u à bu. On commence avec $l = 1$ et trois sommets ε, a et b, et à chaque étape on ajoute les sommets de longueur $l+1$ obtenus en prolongeant les sommets de longueur l (en éliminant ceux qui contiennent le motif), puis on supprime dans le graphe tous les sommets de longueur inférieure ou égale à l qui ne sont pas coaccessibles pour chacun des types d'arêtes ainsi que les sommets non vides qui ne sont pas accessibles, et on recommence jusqu'à ce qu'il n'y ait plus de sommets à effacer. Si le graphe est vide, on a montré que le motif était 2-inévitable; sinon, on incrémente l et on recommence.

Comme le graphe peut devenir très gros pendant le calcul, il peut être utile d'employer une méthode hybride avec la précédente : quand le graphe atteint une certaine taille, on arrête d'y ajouter des sommets et on continue avec du *backtracking* classique pour énumérer les chemins dans le graphe de Rauzy correspondant à la dernière couche du gros graphe. Si au bout d'un certain temps on revient suffisamment en arrière, on peut éliminer certains sommets du graphe et continuer avec un graphe plus petit.

Remarque : Nous avons exposé toutes ces techniques dans le cas de l'alphabet $\{a, b\}$, mais elles se généralisent naturellement à des alphabets de plus de deux lettres.

2.3 3-évitabilité

Il est bien connu, depuis Thue, que le motif AA est 3-évitable. Il est par exemple évité par le mot infini $\mu^\omega(a)$, point fixe du morphisme

$$\begin{array}{rrcl} \mu: & \{a,b,c\}^* & \longrightarrow & \{a,b,c\}^* \\ & a & \longmapsto & abc \\ & b & \longmapsto & ac \\ & c & \longmapsto & b \end{array} .$$

Tous les motifs qui contiennent un carré sont donc aussi 3-évitables. Il nous reste à chercher quels sont les motifs binaires qui ne contiennent pas de carré. Pour cela, on utilise exactement le même algorithme que pour montrer que AA est 2-inévitable : on énumère par *backtracking* tous les mots binaires qui évitent AA, à savoir ε, a, ab, aba, b, ba, bab. Cela signifie que seuls les sept motifs binaires correspondants, ε, A, AB, ABA, B, BA, BAB ne sont pas divisibles par AA et sont susceptibles d'être 3-inévitables. Or on constate que cette liste est exactement celle des motifs binaires inévitables déterminée en 2.1, qui sont bien sûr tous 3-inévitables : un motif binaire est donc soit inévitable, soit 3-évitable.

Il est intéressant de noter qu'on a appliqué aux motifs un résultat concernant initialement les mots. Cette démarche sera utilisée souvent, en particulier pour les problèmes de classification des motifs.

2.4 2-évitabilité : les résultats de P. Roth

L'algorithme de *backtracking* nous laisse penser que certains motifs sont 2-évitables, mais pour le prouver effectivement il est nécessaire de construire des mots infinis qui évitent ces motifs. C'est là la partie la plus délicate de la classification des motifs.

Deux motifs 2-évitables sont connus depuis longtemps : AAA et $ABABA$, qui sont évités par le mot infini de Thue-Morse, point fixe du morphisme

$$\begin{array}{rrcl} \theta: & A^* & \longrightarrow & A^* \\ & a & \longmapsto & ab \\ & b & \longmapsto & ba \end{array}$$

qui commence ainsi :

$$abbabaabbaababbabaababbaabbabaabbaababbaabbabaababbabaabbaab\ldots$$

(voir [39, 55]). Ursula Schmidt puis Peter Roth en ont déterminé d'autres, comme le résume le tableau 2.3.

p	Mot infini
AAA	Thue-Morse
$ABABA$	Thue-Morse
$ABBABAAB$	Fibonacci
$ABBAABBA$	Fibonacci
$ABABBA$	$\nu^\omega(a)$
$AABABB$	$\nu^\omega(a)$
$AABBAA$	$\nu^\omega(a)$
$ABBAAB$	$\nu^\omega(a)$
$ABAAB$	$\psi(\mu^\omega(a))$

Tab. 2.3 - *Motifs binaires 2-évitables*

Dans cette table, μ est le morphisme donné à la section précédente qui permet de construire une suite ternaire sans carré, ν et ψ sont dûs à P. Roth [53] :

$$\begin{array}{rrcl} \nu: & \{a,b\} & \longrightarrow & \{a,b\} \\ & a & \longmapsto & aab \\ & b & \longmapsto & bba \end{array} \qquad \begin{array}{rrcl} \psi: & \{a,b,c\} & \longrightarrow & \{a,b\} \\ & a & \longmapsto & aaa \\ & b & \longmapsto & bbb \\ & c & \longmapsto & ababab \end{array}$$

et la *suite de Fibonacci* est le mot infini

$$\varphi^\omega(a) = abaababaabaababaababaabaababaabaababaababaabaababaababa\ldots$$

point fixe du morphisme

$$\begin{array}{rrcl} \varphi: & \{a,b\} & \longrightarrow & \{a,b\} \\ & a & \longmapsto & ab \\ & b & \longmapsto & a \end{array} .$$

En utilisant les symétries et la relation de divisibilité des motifs, on peut voir que ceci permet de déterminer l'indice de tous les motifs binaires, à l'exception de $AABBA$, $BBAAB$, $ABBAA$ et $BAABB$, qui se déduisent d'ailleurs les uns des autres par symétries. Peter Roth a conjecturé que ces motifs étaient 2-évitables, et nous allons maintenant démontrer cette conjecture.

2.5 Le cas de $AABBA$

Dans cette section, nous allons montrer que le motif $AABBA$ est 2-évitable. La première étape de la preuve consiste à trouver un mot infini sur $\Sigma = \{a,b\}$ susceptible d'éviter le motif étudié. Il n'y a pas vraiment de méthode pour cela.

Nous sommes partis de l'idée suivante : construire ce mot comme l'image par un morphisme π du mot infini sans carré sur un alphabet à 3 lettres $\Sigma' = \{a, b, c\}$ mentionné ci-dessus, $\mu^\omega(a)$. Nous avons d'abord cherché à prendre pour $\pi(a)$, $\pi(b)$ et $\pi(c)$ des facteurs bien choisis d'un long mot évitant $AABBA$ construit par *backtracking*. C'est effectivement possible, par exemple le morphisme suivant convient :

$$
\begin{array}{rrcl}
\pi: & \Sigma'^* & \longrightarrow & \Sigma^* \\
& a & \longmapsto & bbbabbabababbbbaaabaababababbabbbbaaabaababababbbbabaabbba \\
& b & \longmapsto & bbbabbababaabbbabbbabaabbbabbbabbabababbbbabaabbbabbbabbaba \\
& & & \quad babaabababbbbabaabbba \\
& c & \longmapsto & bbbabbababaababbbbabbbbabaabaabbbabbbabbabaabbba
\end{array}
$$

mais il n'est pas très satisfaisant, bien qu'il ait l'avantage d'être *comma-free*.

Nous avons donc cherché à construire un morphisme π beaucoup plus petit au moyen d'une stratégie très différente : essayer systématiquement tous les morphismes pour lesquels la longueur des images de lettres ne dépasse pas une certaine constante l. Il est facile d'éliminer de nombreux candidats en calculant par exemple les 1000 premières lettres de $\pi(\mu^\omega(a))$ et en constatant que le mot obtenu contient $p = AABBA$. On peut espérer qu'un morphisme échappant à ce test donnera un mot infini qui évite p. Avec $l = 4$, on obtient ainsi le morphisme π figurant dans l'énoncé du théorème.

Théorème 2.1. *Soient $\Sigma = \{a, b\}$ et $\Sigma' = \{a, b, c\}$, et μ et π les deux morphismes suivants :*

$$
\begin{array}{rrcl}
\mu: & \Sigma'^* & \longrightarrow & \Sigma'^* \\
& a & \longmapsto & abc \\
& b & \longmapsto & ac \\
& c & \longmapsto & b
\end{array}
,\qquad
\begin{array}{rrcl}
\pi: & \Sigma'^* & \longrightarrow & \Sigma^* \\
& a & \longmapsto & aa \\
& b & \longmapsto & aba \\
& c & \longmapsto & abbb
\end{array}
.
$$

Alors le mot infini $\pi(\mu^\omega(a))$ évite le motif $AABBA$.

Corollaire 2.1. *Le motif $AABBA$ est 2-évitable.* ∎

Pour prouver ce théorème, nous allons utiliser les quelques lemmes qui suivent. Tout d'abord, une propriété classique [39] du morphisme μ, que nous avons déjà en partie utilisée :

Lemme 2.1. *Le mot infini $\mu^\omega(a)$ est sans carré et ne contient ni aba ni cbc comme facteur. (En termes de formules avec constantes : il évite $AA{+}aba{+}cbc$.)* ∎

Soit $Z = \{\pi(a), \pi(b), \pi(c)\}$. L'ensemble Z est un code bipréfixe.

Lemme 2.2. *Si u et au sont tous deux préfixes de mots de Z^*, alors u est une puissance de a.*

Preuve : Supposons que le mot u n'est pas une puissance de a. Il admet donc un préfixe de la forme $a^k b$, où k est un entier positif ou nul. Les mots $a^k b$ et $a^{k+1}b$ sont donc tous deux préfixes de mots de Z^*, ce qui est impossible, car l'un des deux nombres k et $k+1$ est pair, et aucun élément de Z^* ne commence par $(aa)^l b$ car aucun mot de Z ne commence par b. ■

Dans la suite de cette preuve, nous emploierons la convention suivante: quand nous écrivons $v = v_1 \cdot v_2 \cdot v_3 \cdot \cdots \cdot v_k$ (avec des points entre les v_i) cela signifie non seulement que $v = v_1 v_2 v_3 \ldots v_k$, mais aussi que tous les mots v_i sont des éléments de Z^*. Cela permet de discuter de manière concise la manière dont un mot se factorise sur le code Z.

Lemme 2.3. *Si uav appartient à Z^*, alors soit $uav = u \cdot av$, soit $uav{=}ua \cdot v$.*

Preuve : La lettre a n'apparaît dans les éléments du code Z qu'en première ou en dernière position. ■

Définition : Nous dirons qu'un facteur u de $\pi(\mu^\omega(a))$ est *synchronisant* s'il existe une décomposition $u = stp$ où p est un préfixe propre d'un élément de Z (i.e. ε, a, ab ou abb), s est un suffixe propre d'un élément de Z (i.e. ε, a, b, ba, bb ou bbb), et $t \in Z^*$, telle que chaque fois que $\pi(\mu^k(a)) = vuw$ pour des mots v et w et pour un entier k, c'est-à-dire pour chaque occurrence de u comme facteur de $\pi(\mu^\omega(a))$, on a $\pi(\mu^k(a)) = vs \cdot t \cdot pw$.

Lemme 2.4. *Tous les facteurs de $\pi(\mu^\omega(a))$ de longueur 5 ou plus sont synchronisants.*

Preuve :

<u>première étape :</u> Montrons que tout facteur u' de $\pi(\mu^\omega(a))$ contenant un mot synchronisant u est lui-même synchronisant. Soit $u' = qur$, et soit $u = stp$ une décomposition de u satisfaisant la définition des mots synchronisants. Si $\pi(\mu^k(a)) = vu'w = vqurw$, alors $\pi(\mu^k(a)) = vqs \cdot t \cdot prw$. On peut alors écrire $qs = s'q'$ et $pr = r'p'$, où $q', r' \in Z^*$ sont aussi grands que possible, et alors la décomposition $u' = s't'p'$ avec $t' = q'tr'$ montre que u' est synchronisant.

<u>deuxième étape :</u> Les décompositions suivantes montrent que ab, bba, bbb et $aaaa$ sont synchronisants: $vabw = v\varepsilon \cdot \varepsilon \cdot abw$, $vbbaw = vbb \cdot \varepsilon \cdot aw$, $vbbbw = vbbb \cdot \varepsilon \cdot \varepsilon w$, et $vaaaaw = va \cdot aa \cdot aw$ (pour ce dernier mot, nous utilisons le fait que $\mu^\omega(a)$ est sans carré, ce qui interdit $vaaaaw = v \cdot aa \cdot aa \cdot w$).

troisième étape : Comme tout mot de Σ^* de longueur 5 ou plus contient comme facteur au moins l'un des mots ab, bba, bbb ou $aaaa$, nous avons démontré le lemme. ■

Nous devons ensuite décrire tous les facteurs carrés de $\pi(\mu^\omega(a))$:

Lemme 2.5. *Si uu est un facteur de $\pi(\mu^\omega(a))$, alors $u \in \{\varepsilon, b, ba\} \cup a\Sigma^*$.*

Preuve : Supposons que $u \in b\Sigma^*$ et que $\pi(\mu^k(a)) = vuuw$. Nous devons montrer que u est soit b, soit ba.

cas 1 : Supposons d'abord que u est synchronisant : $u = stp$ et $\pi(\mu^k(a)) = vs \cdot t \cdot ps \cdot t \cdot pw$. Le mot s ne peut pas être vide, car sinon $u = tp$ commencerait par la lettre b, ce qui est impossible car tous les mots de Z commencent par la lettre a. On a donc $ps \in Z$ et $s \in b\Sigma^*$. Les seules possibilités sont :

cas 1.1 : $p = a$, $s = ba$. Nous avons alors $\pi(\mu^k(a)) = vba \cdot t \cdot aba \cdot t \cdot aw$. Le seul mot de Z qui se termine par ba est aba, donc la dernière lettre de v est nécessairement un a. En posant $v = v'a$, on obtient : $\pi(\mu^k(a)) = v' \cdot aba \cdot t \cdot aba \cdot t \cdot aw$. Mais cela voudrait dire que $\mu^k(a)$ contient le carré de $\pi^{-1}(abat)$, en contradiction avec le lemme 2.1. Les trois autres cas utilisent un argument similaire.

cas 1.2 : $p = a$, $s = bbb$. La dernière lettre de v doit être un a. $\pi(\mu^k(a)) = v' \cdot abbb \cdot t \cdot abbb \cdot t \cdot aw$ avec $v = v'a$.

cas 1.3 : $p = ab$, $s = bb$. Les deux dernières lettres de v doivent être ab. $\pi(\mu^k(a)) = v' \cdot abbb \cdot t \cdot abbb \cdot t \cdot abw$ avec $v = v'ab$.

cas 1.4 : $p = abb$, $s = b$. Les trois dernières lettres de v doivent être abb. $\pi(\mu^k(a)) = v' \cdot abbb \cdot t \cdot abbb \cdot t \cdot abbw$ avec $v = v'abb$.

Dans tous les cas nous avons trouvé une contradiction. Nous pouvons donc conclure que $\pi(\mu^\omega(a))$ ne contient aucun carré uu avec u synchronisant et $u \in b\Sigma^*$.

cas 2 : Supposons maintenant que le mot u n'est pas synchronisant. D'après le lemme 2.4, la longueur de u est donc au plus 4; de plus nous avons vu que les mots qui contiennent ab, bba ou bbb sont synchronisants. Les seules possibilités restantes pour u sont donc les suivantes, sachant qu'on a supposé que la première lettre de u était un b : b, ba, bb, baa et $baaa$.

cas 2.1 : $u = b$ ou $u = ba$. Les carrés de ces deux mots sont effectivement des facteurs de $\pi(\mu^\omega(a))$, et nous voulons prouver que ce sont les seuls parmi la liste ci-dessus.

cas 2.2 : $u = bb$. On voit aisément qu'aucun mot de Z^* ne peut contenir $bbbb$ comme facteur. Le mot uu ne peut donc pas être facteur de $\pi(\mu^\omega(a))$.

cas 2.3 : $u = baaa$. Alors $\pi(\mu^k(a)) = vb \cdot aa \cdot aba \cdot aa \cdot w$, ce qui est impossible car $\mu^\omega(a)$ ne contient pas aba.

cas 2.4 : $u = baa$. Alors $\pi(\mu^k(a)) = vba \cdot aba \cdot aw$. Comme dans le cas 1.1 ci-dessus, la dernière lettre de v doit être un a, et on obtient $\pi(\mu^k(a)) = v' \cdot aba \cdot aba \cdot aw$ avec $v = v'a$, ce qui est impossible car $\mu^\omega(a)$ est sans carré. ∎

Il serait possible d'affiner le lemme 2.5 pour caractériser les carrés commençant par un a qui apparaissent dans $\pi(\mu^\omega(a))$ (il y en a une infinité). Cependant cela serait assez long et nous n'en avons pas besoin pour montrer que $\pi(\mu^\omega(a))$ évite $AABBA$.

Lemme 2.6. *Si $\pi(\mu^k(a)) = v \cdot uuw$, alors $u \in \{\varepsilon, a\}$.*

Preuve : D'après le lemme 2.5, $u \in \{\varepsilon, b, ba\} \cup a\Sigma^*$. Les cas $u = b$ et $u = ba$ sont impossibles car u est préfixe d'un mot de Z^* qui ne peut pas commencer par la lettre b. Supposons maintenant que $u = at$, et donc $\pi(\mu^k(a)) = v \cdot atatw$. D'après le lemme 2.3 deux cas peuvent se présenter :

cas 1 : $\pi(\mu^k(a)) = v \cdot at \cdot atw$. On a alors nécessairement $\pi(\mu^k(a)) = v \cdot at \cdot at \cdot w$, car Z est un code préfixe. C'est impossible car $\mu^\omega(a)$ est sans carré.

cas 2 : $\pi(\mu^k(a)) = v \cdot ata \cdot tw$. D'après le lemme 2.2, on a $t = a^l$. Si $l = 0$, alors $u = a$. Si $l \geqslant 1$, alors $\pi(\mu^k(a)) = v \cdot aa \cdot aa \cdot a^{2l-2}w$, ce qui contredit à nouveau le fait que $\mu^\omega(a)$ est sans carré. ∎

Lemme 2.7. *Si $atata$ est facteur de $\pi(\mu^\omega(a))$, alors $t = \varepsilon$.*

Preuve : Supposons que $\pi(\mu^k(a)) = vatataw$. D'après le lemme 2.3, soit $\pi(\mu^k(a)) = v \cdot atataw$, soit $\pi(\mu^k(a)) = va \cdot tataw$. Dans les deux cas le lemme 2.6 permet de conclure que $t = \varepsilon$. ∎

Nous avons maintenant assez d'éléments pour démontrer le théorème :

Preuve du théorème 2.1: Supposons que $\pi(\mu^k(a)) = vxxyyxw$ avec $x \neq \varepsilon$, $y \neq \varepsilon$. D'après le lemme 2.5, x et y sont dans l'ensemble $\{b, ba\} \cup a\Sigma^*$.

cas 1 : x et y commencent tous deux par la lettre b. Alors $xxyyx$ est l'un des mots $bbbbb$, $bbbabab$, $babab bba$ ou $bababababa$. Aucun de ces mots ne peut être facteur d'un mot de Z^*, et *a fortiori* de $\pi(\mu^\omega(a))$.

cas 2 : $x = b$, $y = au$. En utilisant le fait que bba est synchronisant, nous obtenons une factorisation $\pi(\mu^k(a)) = vbb \cdot auaubw$. (Dans les cas qui suivent, nous utiliserons également le fait que certains facteurs sont synchronisants pour obtenir une factorisation de $\pi(\mu^k(a))$ sur Z^*, sans le mentionner à chaque fois.) D'après le lemme 2.6, $au = a$, mais alors $\pi(\mu^k(a)) = vbb \cdot aa \cdot bw$, ce qui est impossible car bw ne peut pas être dans Z^* puisqu'il commence par un b.

cas 3 : $x = ba$, $y = au$. Au moyen d'un argument similaire, on obtient la factorisation $\pi(\mu^k(a)) = vb \cdot aba \cdot aa \cdot baw$, qui est impossible car baw ne peut pas être dans Z^*.

cas 4 : $x = at$, $y = b$. Alors $\pi(\mu^k(a)) = vatatbb \cdot atw$, donc le mot $vatat$ se termine nécessairement par ab, ce qui peut être obtenu de deux manières différentes :

cas 4.1 : $t = b$. Alors $\pi(\mu^k(a)) = vab \cdot abbb \cdot abw$, impossible car aucun mot de Z^* ne se termine par ab.

cas 4.2 : $t = uab$. Alors $\pi(\mu^k(a)) = vau \cdot aba \cdot u \cdot abbb \cdot au \cdot abw$, impossible car u et au ne peuvent être tous deux dans Z^*.

cas 5 : $x = at$, $y = ba$. Alors $\pi(\mu^k(a)) = vatatb \cdot aba \cdot atw$, donc le mot $vatat$ se termine nécessairement par abb:

cas 5.1 : $t = bb$. Alors $\pi(\mu^k(a)) = vabb \cdot abbb \cdot aba \cdot abbw$, impossible car aucun mot de Z^* ne se termine par abb.

cas 5.2 : $t = uabb$. Alors $\pi(\mu^k(a)) = vauabbau \cdot abbb \cdot aba \cdot auabbw$, impossible car aucun mot de Z^* ne contient $abba$.

cas 6 : $x = at$, $y = au$: $\pi(\mu^k(a)) = vatatauauatw$. D'après le lemme 2.7, $t = u = \varepsilon$; mais $aaaaa$ n'est pas un facteur de $\pi(\mu^\omega(a))$. ∎

2.6 Autres preuves de 2-évitabilité

Dans ce paragraphe, nous donnons quelques autres résultats de 2-évitabilité, qui seront utiles au chapitre 3. Ces résultats, ainsi que celui de la section précédente, sont résumés dans le tableau 2.4, qui complète le tableau 2.3 de la section 2.4.

Théorème 2.2. *Le mot infini de Thue-Morse évite AABAAB.*

p	Mot infini
$AABBA$	$\pi(\mu^\omega(a))$
$AABAAB$	Thue-Morse
$AABBAB$	$\nu^\omega(a)$

TAB. 2.4 - *Motifs binaires 2-évitables (complément)*

Preuve : Supposons que le mot de Thue-Morse contient $AABAAB$. Pour chaque occurrence du motif, il existe un entier k, des mots u et v et un morphisme non-effaçant φ tels que $\theta^k(a) = u\varphi(AABAAB)v$. On choisit une occurrence où k est minimal. La première lettre de $\varphi(A)$ est, par exemple, a (le cas de b est symétrique). Pour simplifier, nous noterons $\varphi(A) = x$ et $\varphi(B) = y$, et x_i désignera la lettre d'indice i de x (ainsi $x_1 = a$).

Si l'on note $\overline{a} = b$ et $\overline{b} = a$, alors on sait que si c est la lettre d'indice $2j+1$ dans $\theta^k(a)$, alors la lettre qui suit est $\overline{c}$. Par conséquent, si deux lettres identiques se suivent dans $\theta^k(a)$, la première est obligatoirement à une position d'indice pair.

<u>cas 1 :</u> $|x|$ est impair et $|y|$ est pair.

Comme $\theta^k(a)$ contient xx, chaque facteur x_ix_{i+1} de longueur 2 de x apparaît à une position paire et à une position impaire; on a donc $x_{i+1} = \overline{x}_i$, donc $x = (ab)^l a$ avec $l \geqslant 0$. En fait x ne peut être que a ou aba puisque $\theta^\omega(a)$ évite $ABABA$. Comme la première et la dernière lettre de x sont des a, il y a deux a consécutifs dans xx : le premier est donc à une position paire, c'est-à-dire que $|ux|$ est pair. La dernière lettre de u, qui est à une position impaire, est donc un b puisqu'elle est suivie d'un a. De même, $|uxxy|$ est impair et la lettre qui suit $uxxy$ est un a, donc la dernière lettre de y est un b. On peut donc écrire $u = u'b$, $y = y'b$, et on a alors $\theta^k(a) = u'b(xxy')b(xxy')bv$, ce qui est impossible puisque $\theta^\omega(a)$ évite $ABABA$.

<u>cas 2 :</u> $|y|$ est impair.

Alors $|xxy|$ est impair : comme au cas précédent, on en déduit que xxy doit être a ou aba, ce qui est clairement impossible.

<u>cas 3 :</u> $|x|$ et $|y|$ sont pairs.

Si $|u|$ est pair, $|v|$ doit être pair aussi et on a :

$$\theta^{k-1}(a) = \theta^{-1}(u)\left[\theta^{-1}(x)\right]^2\theta^{-1}(y)\left[\theta^{-1}(x)\right]^2\theta^{-1}(y)\theta^{-1}(v)$$

ce qui contredit la minimalité de k. Si $|u|$ est impair, on montre que les dernières lettres de u, x et y sont toutes des b. On peut donc écrire

$x = x'b$, $y = y'b$, $u = u'b$ et on a alors :

$$\theta^{k-1}(a) = \theta^{-1}(u') \left[\theta^{-1}(bx')\right]^2 \theta^{-1}(by') \left[\theta^{-1}(bx')\right]^2 \theta^{-1}(by')\theta^{-1}(bv)$$

ce qui contredit à nouveau la minimalité de k (on peut également conclure comme au cas 1 en remarquant que $\theta^k(a) = u'b(xxy')b(xxy')bv$). ■

Théorème 2.3. *le mot infini $\nu^\omega(a)$ évite $AABBAB$, où ν est le morphisme de $\{a,b\}^*$ défini par $\nu(a) = aab$ and $\nu(b) = bba$.*

Preuve : Ce théorème peut être démontré par la même méthode que le théorème 3.2 de [53]. ■

2.7 Classification des motifs binaires

Grâce aux résultats de Roth et au théorème 2.1, nous connaissons maintenant l'indice d'évitabilité de tous les motifs binaires.

Théorème 2.4. *Les motifs binaires se répartissent dans les trois catégories suivantes :*

- *Motifs inévitables (d'indice infini) :*
  ε, A, AB, ABA, B, BA, BAB.
- *Motifs 2-inévitables 3-évitables (d'indice 3) :*
  AA, AAB, $AABA$, $AABAA$, $AABAB$, $AABB$, $ABAA$, $ABAB$, $ABABB$, ABB, $ABBA$, BAA, $BAAB$, $BABA$, $BABAA$, $BABB$, BB, BBA, $BBAA$, $BBAB$, $BBABA$, $BBABB$.
- *Motifs 2-évitables (d'indice 2) : tous les autres.*

Preuve : On a vu que les motifs AAA et $ABABA$ (Thue [62]), $ABABBA$, $AABABB$, $AABBAA$, $ABBAAB$ et $ABAAB$ (Roth [53]), et $AABBA$ (corollaire 2.1) sont 2-évitables. Par conséquent, tous les motifs divisibles par l'un de ceux-ci ou par l'une de leurs images miroir sont d'indice 2.

Il reste 29 motifs, dont les 7 motifs inévitables qui ont été décrits à la section 2.1 et 22 autres, qui sont tous 3-évitables puisqu'ils contiennent des carrés, et qui sont aussi 2-inévitables d'après la section 2.2, donc dont l'indice est 3. ■

On peut résumer cette classification sous la forme d'un arbre (figure 2.5), dont on coupe les branches dès qu'on rencontre un motif 2-évitable. Par symétrie, on peut se limiter aux motifs commençant par la variable A.

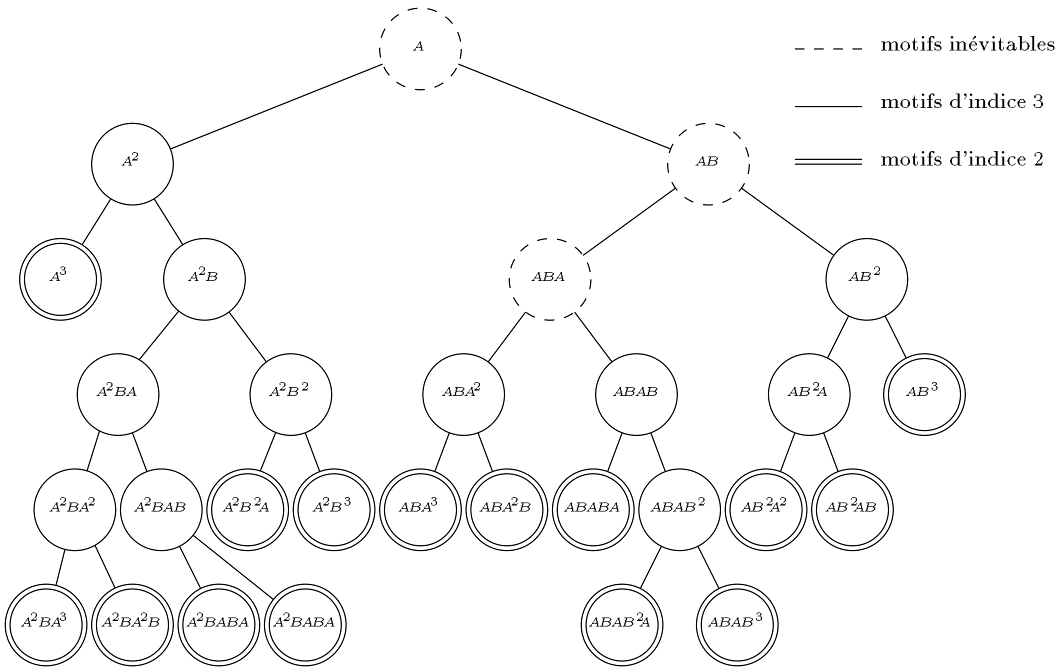

FIG. 2.5 - *Classification des motifs binaires par indice d'évitabilité*

Chapitre 3

DOL-évitabilité, HDOL-évitabilité

Dans le chapitre précédent, nous avons utilisé des mots infinis points fixes de morphismes, ou les images morphiques de ces mots, pour prouver que des motifs sont évitables sur un alphabet donné. En fait, pratiquement toutes les preuves existantes d'évitabilité reposent sur des mots de ces deux types (citons toutefois un résultat récent qui sort de ce cadre: Arturo Carpi [14] et Arto Lepistö [38] on montré que la suite de Kolakovski, qui ne peut vraisemblablement pas être engendrée par un HDOL-système, évite les cubes et que ses carrés sont de longueur bornée).

Il est donc naturel de se demander, d'une part si ces deux types de mots suffisent, et d'autre part si on peut se contenter du premier type (points fixes de morphismes). Ainsi, Peter Roth [53, Section 4] remarque que $ABAAB$ est le seul motif qu'il sait 2-évitable mais pour lequel il ne sait pas construire de morphisme itéré, et se demande si un tel morphisme existe.

Nous allons voir dans ce chapitre que la réponse à cette question est négative: il existe des motifs binaires 2-évitables qui ne sont pas évitables par morphisme itéré sur un alphabet à 2 lettres, et nous les appellerons motifs 2-évitables *2-DOL-inévitables*. Le motif $ABAAB$ est l'un d'entre eux.

3.1 DOL-langages et HDOL-langages

L'utilisation de mots infinis points fixes de morphismes pose quelques problèmes techniques. Il est plus aisé de considérer des ensembles de mots engendrés par des morphismes, ou plus précisément par des L-systèmes. En pratique, les morphismes utilisés pourront définir aussi bien des langages que des mots infinis.

3.1.1 DOL-langages

Formellement, un *DOL-système* est un triplet $G = (\Sigma, f, w)$, où Σ est un alphabet, f est un endomorphisme du monoïde libre Σ^* sur Σ, et w un élément de Σ^*, dit *axiome* de G. À un DOL-système G on associe le langage $L(G) = \{ f^n(w) \mid n \in \mathbb{N} \}$. Pour un alphabet Σ donné, nous noterons $\mathrm{DOL}(\Sigma)$ l'ensemble des langages ainsi obtenus, appelés *DOL-langages* sur Σ.

Note : Le nom des DOL-systèmes veut dire « système de Lindenmayer déterministe sans contexte » (le O est en fait un zéro), du nom de leur inventeur, Aristid Lindenmayer, qui s'en servait pour modéliser la croissance de certains organismes vivants. Comme on peut s'en douter, il en existe aussi des versions non déterministes ou dépendantes du contexte, dont nous ne parlerons pas ici. Pour plus de détails sur les L-systèmes, voir [54].

3.1.2 HDOL-langages

Un *HDOL-langage* (H comme homomorphisme) est l'image d'un DOL-langage par un morphisme. Les alphabets de départ et d'arrivée de ce morphisme peuvent être différents, et le seront d'ailleurs toujours dans les cas qui nous intéressent. Un *HDOL-système* est un quintuplet $G = (\Sigma, f, w, \Sigma', g)$, où $G^0 = (\Sigma, f, w)$ est un DOL-système (dit *sous-jacent* à G) sur l'alphabet Σ, qu'on appellera *alphabet de base* de G, Σ' est un autre alphabet, et g est un morphisme de Σ^* dans Σ'^*, dit *morphisme de codage*. Le HDOL-langage associé à ce HDOL-système est $L(G) = g(L(\Sigma, f, w)) = \{ g(f^n(w)) \mid n \in \mathbb{N} \}$. Nous noterons $\mathrm{HDOL}(\Sigma, \Sigma')$ l'ensemble des HDOL-langages pour des alphabets Σ et Σ' fixés, et $\mathrm{HDOL}(\Sigma') = \bigcup_{n=1}^{\infty} \mathrm{HDOL}(\Sigma_n, \Sigma')$ l'ensemble de tous les HDOL-langages sur Σ', quel que soit leur alphabet de base.

3.1.3 Définitions et propriétés usuelles

Soit $G = (\Sigma, f, w)$ un DOL-système.

Pour tout mot $w' \in \Sigma^*$, nous noterons $G_{w'}$ le L-système (Σ, f, w'), où l'axiome de G a été remplacé par w'. De même, si $G' = (\Sigma, f, w, \Sigma', g)$ est un HDOL-système, $G'_{w'} = (\Sigma, f, w', \Sigma', g)$.

On dit qu'une lettre $x \in \Sigma$ est *de rang* 0 pour G si le langage $L(G_a)$ est fini (voir par exemple [25]). On note en général Σ_0 l'ensemble des lettres de rang 0. Si G' est un HDOL-système, on dit qu'une lettre a est de rang 0 pour G' si elle est de rang 0 pour G'^0; il est donc possible que $L(G'_a)$ soit fini même si a n'est pas de rang 0, pourvu que $L(G'^0_a)$ soit infini.

On dit qu'une lettre a de Σ est *vivante* si, pour tout entier n, le mot $f^n(a)$ est non vide. Une lettre qui n'est pas vivante est nécessairement de rang 0.

Si aucune lettre n'est de rang 0, on dit que G est *expansif* : toute lettre donne naissance à un langage infini. Le DOL-système G est expansif si et seulement s'il existe un entier $n > 0$ tel que f^n est partout croissant. En particulier, si f est partout croissant G est expansif. Par abus de langage, on dira parfois que le morphisme f est expansif quand G est expansif.

3.2 Motifs DOL-évitables et HDOL-évitables

Ces classes de langages nous permettent de parler de motifs DOL(Σ)-évitables (ou *DOL-évitables sur* Σ), HDOL(Σ, Σ')-évitables, et HDOL(Σ')-évitables (ou *HDOL-évitables sur* Σ), au moyen de la définition générale des motifs $\mathscr{L}$-évitables. Si on veut juste préciser la taille de l'alphabet, on dira *k-DOL-évitable*, *k, k'-HDOL-évitable* et *k'-HDOL-évitable*; si on considère tous les alphabets simplement *DOL-évitable* et *HDOL-évitable*, mais nous verrons (proposition 3.3) que ce n'est pas très intéressant car tout motif évitable est DOL-évitable.

On peut également définir l'*indice de DOL-évitabilité* et l'*indice de HDOL-évitabilité* d'un motif p, notés respectivement $\alpha^{\mathrm{DOL}}(p)$ et $\alpha^{\mathrm{HDOL}}(p)$: $\alpha^{\mathrm{DOL}}(p)$ est le plus petit entier k tel que p est k-DOL-évitable et $(k-1)$-DOL-inévitable, ou $+\infty$ si p est DOL-inévitable, et de même pour $\alpha^{\mathrm{HDOL}}(p)$.

3.3 Quelques résultats faciles

Nous commençons par donner quelques propositions qui découlent immédiatement des définitions ci-dessus.

Proposition 3.1. *Les motifs k-DOL-évitables sont k-HDOL-évitables, et les motifs k-HDOL-évitables sont k-évitables.*

Preuve : Il est clair que $\mathrm{DOL}(\Sigma_k) \subset \mathrm{HDOL}(\Sigma_k, \Sigma_k) \subset \mathrm{HDOL}(\Sigma_k) \subset \mathscr{P}(\Sigma_k^*)$. ■

Proposition 3.2. *Les motifs k-DOL-évitables sont $(k+1)$-DOL-évitables, et les motifs k-HDOL-évitables sont $(k+1)$-HDOL-évitables. Plus précisément, les motifs k, k'-HDOL-évitables sont l, l'-HDOL-évitables pour tout (l, l') tel que $l \geqslant k$ et $l' \geqslant k'$.*

Preuve : De même, $\mathrm{DOL}(\Sigma_k) \subset \mathrm{DOL}(\Sigma_{k+1})$, $\mathrm{HDOL}(\Sigma_k) \subset \mathrm{HDOL}(\Sigma_{k+1})$ et $\mathrm{HDOL}(\Sigma_k, \Sigma_{k'}) \subset \mathrm{HDOL}(\Sigma_l, \Sigma_{l'})$. ■

La proposition suivante est implicite dans [5] car les preuves reposent sur des DOL-systèmes, ce qui montre que tout motif évitable est DOL-évitable :

Proposition 3.3. *Un motif est DOL-inévitable (resp. HDOL-inévitable) si et seulement s'il est inévitable.* ■

3.4 Réduction des DOL et HDOL

Dans ce chapitre, nous étudions dans quelle mesure il est possible d'imposer des hypothèses supplémentaires aux L-systèmes (DOL ou HDOL) employés pour éviter des motifs. De telles hypothèses sont utiles parce qu'elles simplifient certaines preuves concernant la DOL-évitabilité.

Ces preuves sont de deux types : celles où on étudie un L-système G donné et celles où on veut montrer un résultat général sur les motifs $\mathscr{L}$-évitables pour une classe $\mathscr{L}$ donnée. Dans le premier cas, la simplification consiste à se ramener, de manière effective, à un L-système G' vérifiant des hypothèses supplémentaires, mais il n'est pas nécessaire que G' conserve toutes les propriétés de G (par exemple l'alphabet de base peut être plus grand). Dans le second cas, elle consiste à considérer au lieu de $\mathscr{L}$ une classe plus petite $\mathscr{L}'$, telle que tout motif $\mathscr{L}$-évitable soit aussi $\mathscr{L}'$-évitable.

Nous supposerons toujours que les L-systèmes considérés engendrent des langages infinis (il est bien connu que c'est décidable) : dans le cas contraire, ils ne sont d'aucune utilité pour l'évitabilité. Nous raisonnerons en général dans le cas des HDOL-systèmes, les DOL-systèmes correspondant au cas particulier $k' = k$ et $g = Id$.

Les deux théorèmes suivants résument les résultats de ce chapitre (les termes *pushy*, *élémentaire* et *partout croissant* sont définis respectivement pages 54, 49 et 10) :

Théorème 3.1. *Soit $G = (\Sigma, f, w, \Sigma', g)$ un HDOL-système. Il est possible de construire, de manière effective, un ensemble fini de mots W et un ensemble fini $\mathscr{G}$ de HDOL-systèmes $G_i = (\Sigma_i, f_i, w_i, \Sigma'_i, g_i)$, avec $\Sigma'_i \subset \Sigma'$, vérifiant les propriétés suivantes :*

1. *$L(G_i)$ est infini;*
2. *G_i n'est pas pushy;*
3. *f_i est élémentaire;*
4. *f_i est partout croissant;*
5. *si $x \in \Sigma_i$ est de rang 0, $f_i(x) = x$;*
6. *si $x \in \Sigma_i$ n'est pas de rang 0, $\mathrm{Alph}\left({f_i}^2(x)\right) = \mathrm{Alph}\left(f_i(x)\right)$;*
7. *si $x \in \Sigma_i$ n'est pas de rang 0, la première et la dernière lettre de $f_i(x)$ ne sont pas de rang 0;*
8. *g_i est non-effaçant.*

de sorte qu'un motif est évité par $L(G)$ si et seulement s'il est évité par l'ensemble $W \cup \bigcup_{G_i \in \mathscr{G}} L(G_i)$.

Théorème 3.2. *Soient k et k' deux entiers tels que $1 \leqslant k' \leqslant k$. Alors tout motif k, k'-HDOL-évitable est $\mathscr{L}$-évitable, où $\mathscr{L}$ est la classe des langages engendrés par des HDOL-systèmes $G = (\Sigma, f, a, \Sigma', g)$, avec $\#\Sigma \leqslant k$ et $\#\Sigma' \leqslant k'$, tels que :*

1. *$a \in \Sigma$;*
2. *$f(a) \in a\Sigma^+$;*
3. *$L(G)$ est infini;*
4. *G n'est pas pushy;*
5. *f est élémentaire;*
6. *si $x \in \Sigma$ est de rang 0, $f(x) = x$;*
7. *si $x \in \Sigma$ n'est pas de rang 0, $\mathrm{Alph}\,(f(x)) = \Sigma$;*
8. *si $x \in \Sigma$ n'est pas de rang 0, la première et la dernière lettre de $f(x)$ ne sont pas de rang 0;*
9. *g est non-effaçant.*

On déduit facilement de ce deuxième théorème sa version DOL :

Corollaire 3.1. *Soit k un entier strictement positif. Tout motif qui est k-DOL-évitable est aussi $\mathscr{L}$-évitable, où $\mathscr{L}$ est la classe des langages engendrés par des DOL-systèmes $G = (\Sigma, f, a)$, avec $\#\Sigma \leqslant k$, vérifiant les conditions 1. à 8. ci-dessus.*

Preuve : On peut appliquer le théorème 3.2 avec $k' = k$, puisque tout motif k-DOL-évitable est k, k-HDOL-évitable. On considère ensuite les DOL-systèmes sous-jacents aux HDOL-systèmes obtenus : si le langage $L(\Sigma, f, a, \Sigma', g)$ évite p, alors $L(\Sigma, f, a)$ évite également p puisque g est non-effaçant. ■

On obtient de la même manière une propriété qui semble naturelle, mais qui n'est pas aisée à démontrer tant qu'on ne s'est pas ramené à g non-effaçant :

Proposition 3.4. *Un motif k, k'-HDOL-évitable est k-DOL-évitable.* ■

Remarque : Les théorèmes 3.1 et 3.2 sont également valables pour les formules sans constante. En effet, dans la plupart des cas, partant d'un L-système G nous construisons un autre L-système G' vérifiant l'hypothèse recherchée et tel que $F(\varphi(L(G'))) \subset F(L(G))$, où φ est un morphisme non-effaçant. (Rappelons que $F(L)$ désigne l'ensemble des facteurs du langage L, et que les motifs évités par $F(L)$ sont exactement les mêmes que ceux évités par L.) Il est alors clair que tout motif évité par $L(G)$ est aussi évité par $L(G')$, et ceci vaut aussi pour les formules. Le résultat ne peut s'étendre aux motifs et formules avec constantes que si φ est l'identité.

3.4.1 Minimalité de l'alphabet

Supposons qu'on veuille montrer qu'un motif p est k, k'-HDOL-évitable. Il suffit de trouver un couple d'entiers $(\widetilde{k}, \widetilde{k'})$ inférieur à (k, k') pour l'ordre partiel produit tel que p est $\widetilde{k}, \widetilde{k'}$-HDOL-évitable. En particulier, on pourra supposer que $(\widetilde{k}, \widetilde{k'})$ est minimal pour cette propriété.

On peut donc se limiter, dans le théorème 3.2, aux HDOL-systèmes pour lesquels $(\widetilde{k}, \widetilde{k'})$ est minimal. Comme ce n'est pas effectif, on ne peut pas en faire autant dans le théorème 3.1, mais en pratique chaque fois que l'un des résultats ci-dessous ne sera pas applicable car l'alphabet n'est pas minimal il sera facile de passer à un alphabet plus petit.

3.4.2 Élimination de l'axiome

La première réduction part de la constatation que le choix de l'axiome w dans un L-système (Σ, f, w) ou $(\Sigma, f, w, \Sigma', g)$ n'est pas essentiel pour les problèmes qui nous intéressent.

Proposition 3.5. *L'ensemble des facteurs de tout DOL-langage infini $L(G)$ contient un sous-DOL-langage infini $L(G_a)$ dont l'axiome est une lettre, l'alphabet et le morphisme étant inchangés. Il en est de même des HDOL-langages.*

Preuve : Raisonnons dans le cas des HDOL-langages, le passage aux DOL-langages se faisant en remplaçant g par l'identité.
Soit $G = (\Sigma, f, w, \Sigma', g)$ le HDOL-système de départ. Notons $w = a_1 a_2 \dots a_m$, où les a_i sont des lettres. Comme $L(G)$ est infini, la suite $(|g(f^n(w))|)$ n'est pas bornée. Mais $|g(f^n(w))| = |g(f^n(a_1))| + \cdots + |g(f^n(a_m))|$, donc l'une des suites $(|g(f^n(a_i))|)$ n'est pas non plus bornée. Notons j la valeur correspondante de i. Le HDOL-langage engendré par $G_{a_j} = (\Sigma, f, a_j, \Sigma', g)$ est évidemment un sous-langage de $F(L(G))$, et on a choisi j de sorte qu'il soit infini, donc il convient. ∎

Pour aller encore plus loin, c'est-à-dire ne même plus avoir besoin de mentionner quelle lettre sert d'axiome, il faut faire attention à deux difficultés, illustrées par le morphisme suivant, avec $\Sigma = \{a, b, c\}$:

$$\begin{array}{rccl} f\colon & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & aaba \\ & b & \longmapsto & b \\ & c & \longmapsto & cc \end{array}$$

qui donne trois DOL-systèmes complètement différents selon la lettre choisie comme axiome. La première difficulté a déjà été envisagée dans la preuve de la proposition 3.5: certaines lettres peuvent donner des DOL-langages finis. On peut donc réduire $G = (\Sigma, f, aba)$ à G_a mais pas à G_b. La seconde vient du fait que la lettre c n'apparaît pas du tout dans le DOL-langage engendré par G, et donc que le DOL-langage $L(G_c)$ peut avoir des propriétés très différentes (en l'occurrence, $L(G_c)$ n'évite aucun motif alors que $L(G)$ évite par exemple $AAAA$).

Pour remédier à cette seconde difficulté, nous appellerons *DOL-système complet* un DOL-système $G = (\Sigma, f, w)$ tel que toutes les lettres de Σ sont utilisées dans $L(\Sigma, f, w)$. Pour toute lettre $a \in \Sigma$ on est alors sûr que $L(G_a)$ est une partie de $F(L(G))$. De même, un *HDOL-système complet* est un HDOL-système $(\Sigma, f, w, \Sigma', g)$ dont le DOL-système sous-jacent (Σ, f, w) est complet, et tel que toutes les lettres de l'alphabet Σ' sont utilisées dans $L(\Sigma, f, w, \Sigma', g)$. Il est alors clair que

Proposition 3.6. *L'ensemble des facteurs de tout DOL-langage infini $L(G)$ contient un sous-DOL-langage infini complet $L(G')$ dont l'axiome est une lettre, sur un alphabet éventuellement plus petit. Il en est de même des HDOL-langages (les deux alphabets pouvant alors être plus petits).*

Preuve : Il suffit de partir du sous-HDOL-langage donné par la proposition 3.5 et de restreindre les morphismes aux lettres effectivement utilisées. ■

Si on a supposé que les alphabets sont minimaux, il est clair que pour toutes les lettres a non de rang 0, le système G_a est complet. De plus, si a et b sont deux lettres non de rang 0, on a $a \in F(L(G_b^0))$ (rappelons que G_b^0 est le DOL-système sous-jacent à G_b), donc $L(G_a^0) \in F(L(G_b^0))$, donc $L(G_a) \in F(L(G_b))$. On a de même $L(G_b) \in F(L(G_a))$, donc $F(L(G_a)) = F(L(G_b))$: les HDOL-langages engendrés par G_a et G_b évitent donc exactement les mêmes motifs : le choix de la lettre prise comme axiome est indifférent, du moment que ce n'est pas une lettre de rang 0.

3.4.3 Morphismes élémentaires

Un morphisme $h\colon \Sigma^* \to \Sigma'^*$ est dit *simplifiable* s'il existe un alphabet Σ'' de cardinal strictement plus petit que celui de Σ et deux morphismes $h_1\colon \Sigma^* \to \Sigma''^*$ et $h_2\colon \Sigma''^* \to \Sigma'^*$ tels que $h = h_2 \circ h_1$. Si h n'est pas simplifiable, on dit qu'il est *élémentaire.*

Proposition 3.7. *Soit $G = (\Sigma, f, w, \Sigma', g)$ un HDOL-système. Il existe (et on peut construire) un HDOL-système $\widetilde{G} = (\widetilde{\Sigma}, \widetilde{f}, \widetilde{w}, \Sigma', \widetilde{g})$, avec $\widetilde{f}$ élémentaire et $\#\widetilde{\Sigma} \leqslant \#\Sigma$, et un ensemble fini de mots W, tels que $L(G) = L(\widetilde{G}) \cup W$.*

Preuve : Supposons que f est simplifiable : il existe alors un alphabet $\widetilde{\Sigma}$ de cardinal strictement plus petit que celui de Σ et deux morphismes $f_1\colon \Sigma^* \to \widetilde{\Sigma}^*$

et $f_2 \colon \widetilde{\Sigma}^* \to \Sigma^*$ tels que $f = f_2 \circ f_1$. Posons $\widetilde{f} = f_1 \circ f_2$ et $\widetilde{g} = g \circ f_2$:

$$\begin{tikzcd}
\Sigma^* \arrow[r, "f"] \arrow[dr, "f_1"'] & \Sigma^* \arrow[r, "f"] \arrow[dr, "f_1"'] & \Sigma^* \arrow[r, "g"] & \Sigma'^* \\
& \widetilde{\Sigma}^* \arrow[u, "f_2"] \arrow[r, "\widetilde{f}"'] & \widetilde{\Sigma}^* \arrow[u, "f_2"] \arrow[ur, "\widetilde{g}"'] &
\end{tikzcd}$$

ce qui donne, pour tout entier $k > 0$:

$$g \circ f^k = g \circ (f_2 \circ f_1)^k = g \circ f_2 \circ (f_1 \circ f_2)^{k-1} \circ f_1 = \widetilde{g} \circ \widetilde{f}^{k-1} \circ f_1$$

donc, si $\widetilde{w} = f_1(w)$, $g(f^k(w)) = \widetilde{g}(\widetilde{f}^{k-1}(\widetilde{w}))$. On a donc $L(G) = L(\widetilde{G}) \cup \{g(w)\}$. Si $\widetilde{f}$ est élémentaire, on a terminé. Sinon, il suffit de recommencer l'opération; comme la taille de l'alphabet de base décroît strictement, on finira par arriver à un morphisme élémentaire. (Si $L(G)$ est infini, on s'arrêtera au plus tard avec un alphabet à une lettre; si $L(G)$ est fini, on peut parfois continuer jusqu'à l'alphabet vide, sur lequel l'unique morphisme $\varepsilon \mapsto \varepsilon$ est élémentaire...) L'ensemble W des mots à ajouter contiendra les premiers termes de la suite associée à G : $g(w)$, $g(f(w))$, etc. ■

Remarque : Il n'est pas toujours possible de se ramener à un morphisme élémentaire en une fois. Ainsi, si $\Sigma = \{a, b, c\}$ et f est le morphisme défini par $f(a) = abab$, $f(b) = ab$, $f(c) = bb$, il n'est pas possible de factoriser f par l'alphabet à une lettre (car les images ne sont pas puissances d'un même mot). Mais si on factorise par l'alphabet à deux lettres, par exemple avec $f_1(a) = aa$, $f_1(b) = a$, $f_1(c) = b$, $f_2(a) = ab$, $f_2(b) = bb$, de sorte que $f = f_2 \circ f_1$, alors $\widetilde{f} = f_1 \circ f_2$ est à son tour simplifiable : $\widetilde{f}(a) = aaa$, $\widetilde{f}(b) = aa$, et finalement on réduit f en deux étapes à un morphisme élémentaire sur l'alphabet $\{a\}$.

Si on a supposé que l'alphabet Σ est de taille minimale, alors on est sûr que f est déjà élémentaire.

3.4.4 Itération du morphisme

Dans un DOL-système, remplacer le morphisme par une de ses puissances revient à ne prendre qu'un élément sur n dans la suite de mots associée, ce qui en général ne perturbe pas les propriétés d'évitabilité. La proposition suivante est immédiate :

Proposition 3.8. *Soit $G = (\Sigma, f, w, \Sigma', g)$ un HDOL-système, et $n > 0$ un entier. Pour tout k compris entre 0 et $n-1$, on note G_k le HDOL-système $G_k = (\Sigma, f^n, f^k(w), \Sigma', g)$. On a alors*

$$L(G) = \bigcup_{0 \leqslant k < n} L(G_k)$$

et au moins l'un des langages $L(G_k)$ est infini. ■

La proposition suivante nous sera utile :

Proposition 3.9. *Soit E un ensemble fini, et φ une application de E dans E. Alors il existe un entier N tel que l'application itérée φ^N vérifie :*

$$\forall x \in E, (\varphi^N)^2(x) = \varphi^N(x) \ .$$

Preuve : Pour tout $x \in E$, la suite $\varphi^n(x)$ est ultimement périodique, de période $n_x \leqslant \#E$, avec une phase transitoire de longueur $t_x < \#E$; si N est un multiple de n_x supérieur à t_x, on a $\varphi^N(x) = \varphi^{2N}(x)$. On prend donc pour N un multiple commun de tous les n_x qui soit plus grand que tous les t_x, et φ^N a la propriété cherchée. On peut par exemple prendre $N = \#E!$. ∎

Nous allons d'abord appliquer cette propriété aux lettres de rang 0 :

Proposition 3.10. *Soit $G = (\Sigma, f, w)$ un DOL-système. Il existe un entier N tel que pour toute lettre b de rang 0 pour G on ait $f^N(b) = f^{2N}(b)$ (toutes les lettres de $f^N(b)$ sont donc invariantes par f^N).*

Preuve : On applique la proposition 3.9 à la restriction de f à l'ensemble fini $E = \bigcup_{b \in \Sigma_0} L(G_b)$, où Σ_0 est l'ensemble des lettres de rang 0. ∎

Faisons de même avec les alphabets des images de lettres :

Proposition 3.11. *Soit $G = (\Sigma, f, w)$ un DOL-système. Il existe un entier N tel que pour toute lettre a on ait $\mathrm{Alph}\left(f^N(a)\right) = \mathrm{Alph}\left(f^{2N}(a)\right)$.*

Preuve : Cette fois, on applique la proposition 3.9 à l'application $\mathrm{Alph}\,(f)$ de $\mathscr{P}(\Sigma)$ dans $\mathscr{P}(\Sigma)$ qui à tout ensemble de lettres S associe l'ensemble

$$\bigcup_{a \in S} \mathrm{Alph}\,(f(a))$$

de toutes les lettres apparaissant dans les images d'éléments de S. ∎

Et enfin avec les premières (et dernières) lettres non de rang 0 des images de lettres non de rang 0 :

Proposition 3.12. *Soit $G = (\Sigma, f, w)$ un DOL-système. Pour toute lettre a non de rang 0, on désigne par $\rho_1(a)$ la première lettre non de rang 0 qui apparaît dans $f(a)$, et par $\rho_2(a)$ la dernière. Il existe un entier N tel que pour toute lettre a non de rang 0, $\rho_1^N(a) = \rho_1^{2N}(a)$ et $\rho_2^N(a) = \rho_2^{2N}(a)$.*

Preuve : On applique directement la proposition 3.9 à ρ_1 et ρ_2. ■

Résumons les résultats obtenus jusqu'ici :

Proposition 3.13. *Soit G un HDOL-système. On peut construire, de manière effective, un ensemble fini de mots W et un ensemble fini $\mathscr{G}$ de HDOL-systèmes complets infinis dont les alphabets sont inclus dans ceux de G et tels que $L(G) = W \cup \bigcup_{G' \in \mathscr{G}} L(G')$. De plus, pour chaque élément $G' = (\Sigma, f, w, \Sigma', g)$ de $\mathscr{G}$, le morphisme f est élémentaire et il existe une partition de Σ en quatre ensembles Σ'_0, Σ''_0, Σ'_1 et Σ''_1 (tous peuvent être vides, sauf Σ'_1) tels que :*

- *$\Sigma_0 = \Sigma'_0 \cup \Sigma''_0$ est l'ensemble des lettres de rang 0 pour G';*
- *si $a \in \Sigma'_0$, $f(a) = a$;*
- *si $a \in \Sigma''_0$, $f(a) \in {\Sigma'_0}^*$;*
- *si $a \in \Sigma_1 = \Sigma'_1 \cup \Sigma''_1$, $\mathrm{Alph}\,(f^2(a)) = \mathrm{Alph}\,(f(a))$;*
- *si $a \in \Sigma'_1$, $L(G'_a)$ est infini;*
- *si $a \in \Sigma''_1$, $L({G'_a}^0)$ est infini mais $L(G'_a)$ est fini;*
- *enfin, si ρ_1 et ρ_2 sont les applications de Σ_1 dans Σ_1 définies dans la proposition 3.12, pour tout $a \in \Sigma_1$ on a $\rho_1^2(a) = \rho_1(a)$ et $\rho_2^2(a) = \rho_2(a)$.*

Preuve : On applique la proposition 3.7, puis, pour le HDOL-système obtenu, la proposition 3.8 en prenant pour N le produit des valeurs données par les propositions 3.9, 3.10, 3.11 et 3.12. Comme une puissance d'un morphisme élémentaire n'est pas toujours élémentaire, il peut être nécessaire d'appliquer à nouveau la proposition 3.7 sur chacun des HDOL-systèmes obtenus, mais cela finira par s'arrêter puisque les alphabets décroissent. On restreint alors la taille des alphabets pour avoir des L-systèmes complets, et on intègre à W les éventuels HDOL-langages finis.
Pour chaque HDOL-système de $\mathscr{G}$, on appelle Σ'_0 l'ensemble des lettres invariantes. Si a est de rang 0 mais n'est pas invariante, c'est-à-dire $a \in \Sigma''_0$, $f(a) \in {\Sigma'_0}^*$ puisque $f^2(a) = f(a)$; on répartit les lettres non de rang 0 entre Σ'_1 et Σ''_1 selon que $L(G'_a)$ est fini ou non; comme $L(G')$ est infini, Σ'_1 est non vide d'après la proposition 3.5. ■

3.4.5 Morphismes non-effaçants

Puisqu'on s'est ramené à f élémentaire, à plus forte raison f est non-effaçant (puisque tout morphisme élémentaire est injectif). Par contre, rien ne permet pour le moment de supposer g non-effaçant, ce qui s'avère très utile, par exemple pour démontrer la proposition 3.4. Nous pouvons maintenant nous y attaquer :

Proposition 3.14. *Dans la proposition 3.13, on peut ajouter la condition supplémentaire suivante : pour tout* $G' = (\Sigma, f, w, \Sigma', g) \in \mathscr{G}$, *g est non-effaçant et* $\Sigma_1'' = \emptyset$.

Preuve : Supposons que g est effaçant. Pour tout $a \in \Sigma_1'$, comme $L(G_a')$ n'est pas réduit au mot vide, il existe une lettre b dans $F(L({G_a'}^0))$ dont l'image par g n'est pas vide. Puisque $\mathrm{Alph}\,(f^2(a)) = \mathrm{Alph}\,(f(a))$, cette lettre b apparaît dans $f(a)$, donc on a $g(f(a)) \neq \varepsilon$.
L'ensemble Σ_2 des lettres a telles que $g(f(a)) = \varepsilon$ est donc inclus dans $\Sigma_0 \cup \Sigma_1''$. Si $a \in \Sigma_2$, pour toute lettre b de $\mathrm{Alph}\,(f(a))$ on a $\mathrm{Alph}\,(f(b)) \subset \mathrm{Alph}\,(f(a))$ donc $g(f(b))$ est aussi égal au mot vide : le monoïde libre Σ_2^* est donc stable par f.
Soit alors $\widetilde{\Sigma} = \Sigma \setminus \Sigma_2$ et $\widetilde{f} \colon \widetilde{\Sigma} \to \widetilde{\Sigma}$ qui à toute lettre x associe le mot obtenu en effaçant dans $f(x)$ les éléments de Σ_2. Si δ est l'application de Σ^* dans $\widetilde{\Sigma}^*$ consistant à effacer les éléments de Σ_2, on voit donc que le diagramme suivant est commutatif :

$$\begin{tikzcd}
\Sigma^* \arrow[r, "f"] \arrow[d, "\delta"] & \Sigma^* \arrow[r, "f"] \arrow[d, "\delta"] & \Sigma^* \arrow[r, "g"] & \Sigma'^* \\
\widetilde{\Sigma}^* \arrow[r, "\widetilde{f}"] & \widetilde{\Sigma}^* \arrow[r, "f"] & \Sigma^* \arrow[ur, "g"] &
\end{tikzcd}$$

puisque si un mot u contient des lettres de Σ_2, leur contribution à $f(u)$ est formée uniquement de lettres de Σ_2, et elles sont effacées par $g \circ f$. On prend donc $\widetilde{w} = \delta(w)$ et $\widetilde{g} = g \circ f$, et on a pour tout $k > 0$ $g(f^k(w)) = \widetilde{g}(f^{k-1}(\widetilde{w}))$. On obtient donc un nouveau HDOL-système $(\widetilde{\Sigma}, \widetilde{f}, \widetilde{w}, \Sigma', \widetilde{g})$ et un mot isolé supplémentaire $g(w)$. Si jamais $\widetilde{f}$ est simplifiable, il faut recommencer à appliquer la proposition 3.13, mais la taille de l'alphabet a diminué donc on finit par ne plus avoir de g effaçant. L'ensemble Σ_1'' est alors vide, car quand g est non-effaçant $L(G_a')$ est infini si et seulement si $L({G_a'}^0)$ l'est. ■

Si on a supposé Σ minimal, tout cela ne peut pas se produire : on est donc sûr que g est non-effaçant.

3.4.6 Le cas des L-systèmes *pushy*

Les HDOL-systèmes qui contiennent des puissances arbitraires, dits *répétitifs*, ou *fortement répétitifs* s'ils contiennent des puissances arbitraires d'un même mot, ne sont, d'après la proposition 1.5, d'aucune utilité pour l'évitabilité des motifs sans constante.

Ehrenfeucht et Rozenberg [26] ont démontré qu'on pouvait décider si un DOL-langage donné est (fortement) répétitif ou non (un DOL-langage répétitif est toujours fortement répétitif). Mignosi et Séébold [40] ont récemment donné une nouvelle démonstration de ce résultat, plus précise puisqu'on peut décider si un DOL-langage évite une puissance donnée. Parmi les DOL-systèmes qui engendrent des langages répétitifs, certains sont facilement identifiables : il s'agit des DOL-systèmes *pushy* : un DOL-système $G = (\Sigma, h, w)$ est dit *pushy* si $F(L(G))$ contient des mots arbitrairement longs composés uniquement de lettres de rang 0. Dans le cas contraire, G est dit *non pushy* et on note $q(G)$ la longueur maximale des mots de $F(L(G)) \cap \Sigma_0^*$ (rappelons que Σ_0 désigne l'ensemble des lettres de rang 0). On dira qu'un HDOL-système est *pushy* si et seulement si le DOL-système sous-jacent l'est.

Remarque : Nous n'avons pas trouvé de traduction française de *pushy* qui fasse l'unanimité. Jean Berstel propose *poussif*, mais l'idée est plutôt qu'un système *pushy* n'arrête pas de pousser des lettres de rang 0 : *poussant* ou *pousseur* serait donc préférable.

Ehrenfeucht et Rozenberg démontrent la proposition suivante, qui caractérise les DOL-systèmes *pushy* (l'énoncé original prend $x \in \Sigma$, mais cela ne suffit pas si G n'est pas complet) :

Proposition 3.15. *Un DOL-système $G = (\Sigma, h, w)$ est pushy si et seulement s'il vérifie la* condition de bord *: il existe une lettre $x \in \mathrm{Alph}\,(L(G))$, un entier $k > 0$ et deux mots $v \in \Sigma^*$ et $u \in \Sigma_0^+$ tels que $\mathrm{Alph}\,(u)$ contient une lettre vivante et soit $h^k(x) = vxu$, soit $h^k(x) = uxv$.* ∎

Cette caractérisation permet de voir aisément, d'une part qu'il est décidable si un DOL-système donné est *pushy*, et d'autre part qu'un DOL-système *pushy* engendre un langage fortement répétitif. Cela s'étend évidemment aux HDOL-systèmes, à condition que le morphisme de codage n'efface pas toutes les puissances :

Proposition 3.16. *Un HDOL-système pushy dont le morphisme de codage est non-effaçant engendre un langage fortement répétitif.* ∎

On peut donc éliminer tous les systèmes *pushy* de notre ensemble $\mathscr{G}$:

Proposition 3.17. *Dans la proposition 3.13, si $L(G)$ est non répétitif, on peut ajouter les conditions supplémentaires suivantes : G' est non pushy; pour tout $a \in \Sigma_1$, $f(\rho_1(a)) \in \rho_1(a)\Sigma^+$ et $f(\rho_2(a)) \in \Sigma^+\rho_2(a)$.*

Preuve : Si l'un des HDOL-systèmes $G' = (\Sigma, f, w, \Sigma', g)$ obtenus est *pushy*, $L(G)$ est nécessairement répétitif. Sinon, la condition de bord n'est pas satisfaite donc $f(\rho_1(a))$ ne peut pas commencer par des lettres de rang 0 (nécessairement vivantes puisque f est non-effaçant), donc commence directement par $\rho_1(a)$, et de même $f(\rho_2(a))$ se termine par $\rho_2(a)$. ∎

3.4.7 Simplification des images des lettres de rang 0

Pour le moment, les lettres de rang 0 sont réparties en deux catégories : celles qui sont invariantes par f (l'ensemble Σ'_0) et celles dont l'image est dans ${\Sigma'_0}^*$. Nous allons voir que ces dernières (l'ensemble Σ''_0) peuvent être éliminées. Par un procédé similaire, nous allons également éliminer les bords de rang 0 des images des lettres qui ne sont pas de rang 0, terminant ainsi le travail commencé à la proposition 3.17 dans le cas des lettres images par ρ_1 et ρ_2 :

Proposition 3.18. *Dans la proposition 3.13, si $L(G)$ est répétitif, on peut ajouter les conditions supplémentaires suivantes : pour tout $G' \in \mathscr{G}$, $\Sigma''_0 = \emptyset$ et pour tout $a \in \Sigma_1$, $f(a) \in \Sigma_1\Sigma^*\Sigma_1$.*

Preuve : Soit $\widetilde{\Sigma} = \Sigma \setminus \Sigma''_0 = \Sigma'_0 \cup \Sigma_1$. On définit deux morphismes $f_1 \colon \Sigma^* \to \widetilde{\Sigma}^*$ et $f_2 \colon \widetilde{\Sigma}^* \to \Sigma^*$ par $f_1(a) = a$ si $a \in \widetilde{\Sigma}$, $f_1(a) = f(a)$ si $a \in \Sigma''_0$, et $f_2(a) = f(a)$ si $a \in \widetilde{\Sigma}$. On a alors $f = f_2 \circ f_1$, donc puisque f est élémentaire $\widetilde{\Sigma} = \Sigma$ et $\Sigma''_0 = \emptyset$.

Soit maintenant deux nouveaux morphismes $f_1 \colon \Sigma^* \to \Sigma^*$ et $f_2 \colon \Sigma^* \to \Sigma^*$, définis comme suit : si $a \in \Sigma_0$, $f_1(a) = f_2(a) = f(a) = a$; si $a \in \Sigma_1$, on écrit $a = u_1(a)v(a)u_2(a)$, où $u_1(a)$ et $u_2(a)$ sont dans Σ_0^* et $v(a)$ commence par $\rho_1(a)$ et se termine par $\rho_2(a)$; on pose alors $f_1(a) = u_1(a)au_2(a)$ et $f_2(a) = v(a)$. On a encore $f = f_2 \circ f_1$, ce qui cette fois ne contredit plus le fait que f est élémentaire. On peut tout de même utiliser la construction de la proposition 3.7 : posons $\widetilde{f} = f_1 \circ f_2$, $\widetilde{g} = g \circ f_2$ et $\widetilde{w} = f_1(w)$ (il ne faut pas oublier d'ajouter $g(w)$ à l'ensemble W). Si $a \in \Sigma_1$, on voit que $\widetilde{f}(a) = f_1(v(a))$, qui commence par $f_1(\rho_1(a)) = \rho_1(a)u_2(\rho_1(a))$ puisque $u_1(\rho_1(a)) = \varepsilon$ et se termine par $u_1(\rho_2(a))\rho_2(a)$, donc $\widetilde{f}(a) \in \Sigma_1 \cup \Sigma_1\Sigma^*\Sigma_1$. Il reste à s'assurer que le nouveau HDOL-système $\widetilde{G}' = (\Sigma, \widetilde{f}, \widetilde{w}, \Sigma', \widetilde{g})$ vérifie bien toutes les conditions des propriétés 3.13, 3.14 et 3.17. Il se peut que $\widetilde{f}$ ne soit pas élémentaire, et il faut alors tout recommencer en faisant décroître la taille de l'alphabet comme nous en avons désormais l'habitude; il se peut aussi que pour une lettre $a \in \Sigma_1$ on ait $\mathrm{Alph}\left(\widetilde{f}^2(a)\right) \neq \mathrm{Alph}\left(\widetilde{f}(a)\right)$, si certaines lettres de rang 0 n'apparaissaient dans $f(a)$ qu'au bord; il est par contre certain que $\mathrm{Alph}\left(\widetilde{f}^3(a)\right) = \mathrm{Alph}\left(\widetilde{f}^2(a)\right)$, car $\mathrm{Alph}\left(\widetilde{f}^2(a)\right) \cap \Sigma_1 = \mathrm{Alph}\left(\widetilde{f}(a)\right) \cap \Sigma_1$. Dans ce cas, on applique la proposition 3.8 avec $N = 2$, ce qui peut à nouveau donner un morphisme simplifiable mais ne touche pas aux autres propriétés. On élimine du même coup la possibilité d'avoir $f(a) \in \Sigma_1$, donc on a bien $f(a) \in \Sigma_1\Sigma^*\Sigma_1$ pour $a \in \Sigma_1$ comme annoncé. ■

3.4.8 Morphismes expansifs

Notre dernière réduction va, contrairement aux précédentes, augmenter la taille de l'alphabet Σ. Il s'agit d'éliminer complètement les lettres de rang 0, de sorte que les L-systèmes seront tous expansifs, et même partout croissants puisqu'on a supposé que $\mathrm{Alph}\,(f(a)) = \mathrm{Alph}\,(f^2(a))$ pour toute lettre a. Notre construction est similaire à celle d'Ehrenfeucht et Rozenberg [26, Claim 4.1], mais fonctionne mieux grâce à la proposition 3.18

Proposition 3.19. *Dans la proposition 3.13, si $L(G)$ est répétitif et à condition de ne plus supposer que Σ est inclus dans l'alphabet de base de G, on peut ajouter la condition supplémentaire suivante : pour tout $G' = (\Sigma, f, w, \Sigma', g) \in \mathscr{G}$, f est partout croissant.*

Preuve : Soit $G' = (\Sigma, f, w, \Sigma', g) \in \mathscr{G}$. Supposons tout d'abord que la première lettre de w n'est pas de rang 0 (i.e. est dans Σ_1). On construit alors un nouveau système $\widetilde{G}' = (\widetilde{\Sigma}, \widetilde{f}, \widetilde{w}, \Sigma', \widetilde{g})$ avec $\widetilde{f}$ partout croissant, qui engendre le même langage que G'. L'alphabet de base $\widetilde{\Sigma}$ de $\widetilde{G}'$, qui peut être plus gros que celui de G', est formé des mots de $\Sigma_1\Sigma_0^*$ qui apparaissent dans $L(G'^0)$ soit suivis d'une lettre de Σ_1, soit en fin de mot. Pour bien distinguer les éléments de $\widetilde{\Sigma}$ des mots correspondants, on notera $[av]$ le symbole de $\widetilde{\Sigma}$ associé au mot $av \in \Sigma_1\Sigma_0^*$. Puisque G' n'est pas *pushy*, l'ensemble $\widetilde{\Sigma}$ est fini. Soit i le morphisme défini par

$$i\colon \begin{array}{ccc} \widetilde{\Sigma}^* & \longrightarrow & \Sigma^* \\ [av] & \longmapsto & av \end{array}.$$

Alors il existe des morphismes $\widetilde{f}\colon \widetilde{\Sigma}^* \to \widetilde{\Sigma}^*$ et $\widetilde{g}\colon \widetilde{\Sigma}^* \to \Sigma'^*$ tels que le diagramme suivant est commutatif :

$$\begin{tikzcd}
\widetilde{\Sigma}^* \arrow[r, "\widetilde{f}"] \arrow[d, "i"] & \widetilde{\Sigma}^* \arrow[d, "i"] \arrow[dr, "\widetilde{g}"] & \\
\Sigma^* \arrow[r, "f"] & \Sigma^* \arrow[r, "g"] & \Sigma'^*
\end{tikzcd}$$

(ils sont définis par $\widetilde{f}([av]) = i^{-1}(f(av))$ et $\widetilde{g} = g \circ i$). Si $\widetilde{w} = i^{-1}(w)$, il est clair que $\widetilde{g}(\widetilde{f}^n(\widetilde{w})) = g(f^n(w))$, et donc $L(\widetilde{G}') = L(G')$. Le nouveau morphisme $\widetilde{f}$ est partout croissant : $\left|\widetilde{f}([av])\right|$ est égale au nombre de lettres de Σ_1 dans $f(a)$, qui vaut au moins 2 puisque $f(a) \in \Sigma_1\Sigma^*\Sigma_1$.

Si la première lettre de w est de rang 0, notons u le plus long préfixe de w formé de lettres de rang 0 (u ne peut être égal à w car $L(G')$ est infini). Cette fois, $\widetilde{\Sigma}$ est formé de l'ensemble $\widetilde{\Sigma}_1$ des symboles $[av]$ comme ci-dessus, à condition toutefois qu'ils apparaissent ailleurs qu'en préfixe de $u^{-1}L(G')$, auquel on ajoute l'ensemble $\widetilde{\Sigma}_2$ des symboles $[uav]$ pour tous les mots $uav \in u\Sigma_1\Sigma_0^*$ qui apparaissent comme préfixes de mots de $L(G')$ suivis d'une lettre de Σ_1 ou de rien (il y a au plus trois symboles supplémentaires de ce type : un préfixe de w, disons ubv_1, un préfixe de $f(w)$, $u\rho_1(b)v_2$, et un préfixe de $f^k(w)$ pour $k \geqslant 2$, $u\rho_1(b)v_3$. On prolonge le morphisme i par $i([uav]) = uav$, et $\widetilde{f}$ de sorte que $\widetilde{f}(x) \in \widetilde{\Sigma}_1^*$ si $x \in \widetilde{\Sigma}_1$ et $\widetilde{f}(x) \in \widetilde{\Sigma}_2\widetilde{\Sigma}_1^*$ si $x \in \widetilde{\Sigma}_2$. Tout fonctionne alors comme ci-dessus.

Le nouveau système conserve la plupart des propriétés de G', mais $\widetilde{f}$ peut ne pas être élémentaire et ne pas satisfaire $\mathrm{Alph}\left(\widetilde{f}^2(a)\right) = \mathrm{Alph}\left(\widetilde{f}(a)\right)$. Mais si c'est le cas on peut appliquer les constructions des propriétés 3.7 et 3.8 en remarquant qu'elles conservent l'expansivité du morphisme (c'est évident pour l'itération; pour la simplification, si $f = f_2 \circ f_1$ est expansif avec f_2 non-effaçant, il existe un entier n tel que f^n est partout croissant; alors $f_1 \circ f^n \circ f_2 = (f_1 \circ f_2)^{n+1}$ est aussi partout croissant). ∎

3.4.9 Récapitulation

En regroupant les résultats des paragraphes précédents, on obtient le théorème 3.1. Il faut faire attention à un détail : à partir de la proposition 3.17, on a supposé que $L(G)$ n'était pas répétitif, et dans ce cas on a construit W et $\mathscr{G}$ tels que $L(G) = W \cup \bigcup_{G' \in \mathscr{G}} L(G')$; si $L(G)$ est répétitif, il n'évite aucun motif : on peut alors prendre arbitrairement $W = \emptyset$ et $\mathscr{G} = \{G'\}$ avec $G' = (\{a\}, f, a, \{a\}, Id)$ et $f(a) = aa$, de telle sorte que $L(G')$ n'évite aucun motif non plus.

Pour démontrer le théorème 3.2, on part d'un motif p qui est k, k'-HDOL-évitable et d'un HDOL-système G qui évite p avec des alphabets minimaux Σ et Σ' (et qui est donc forcément non répétitif). Si on applique à G les propositions ci-dessus, à l'exception de la proposition 3.19, on obtient des ensembles W et $\mathscr{G}$ tels que $L(G) = W \cup \bigcup_{G' \in \mathscr{G}} L(G')$. Puisque $L(G)$ est infini, $\mathscr{G}$ est non vide. On choisit un élément quelconque $G' = (\Sigma, f, w, \Sigma', g)$ de $\mathscr{G}$ (les alphabets sont les mêmes que ceux de G puisqu'ils sont minimaux) et une lettre b qui n'est pas de rang 0 pour G'. On prend alors $a = \rho_1(b)$, qui vérifie $f(a) \in a\Sigma^+$. Enfin, pour toute lettre $x \in \Sigma_1$, $\mathrm{Alph}(f(x)) = \Sigma$ car sinon $G'_{f(x)}$ serait un HDOL-système infini évitant p d'alphabet de base strictement plus petit que Σ. Le HDOL-système G'_a convient donc pour le théorème 3.2.

3.5 Preuves de DOL-inévitabilité

Revenons maintenant aux motifs binaires : nous allons essayer de trouver tous ceux qui sont 2-DOL-inévitables.

Pour montrer qu'un motif donné p est 2-DOL-évitable, il suffit de construire un DOL-langage qui évite le motif p. Il est moins facile de prouver qu'un motif p est 2-DOL-inévitable. Dans les deux cas que nous étudierons ici, $ABAABA$ et $AABBA$, nous montrons d'une part que p et AAA sont simultanément 2-inévitables, c'est-à-dire que la formule $p + AAA$ est 2-inévitable, et d'autre part que tout DOL-langage qui contient AAA contient nécessairement aussi p: en combinant ces deux propositions, on voit que p est 2-DOL-inévitable.

Lemme 3.1. *La formule $AABBA + AAA$ est 2-inévitable.*

Preuve : Au moyen de l'algorithme de *backtracking*, on dresse la liste de tous les mots binaires qui évitent $AABBA + AAA$. Il y en a 187 en tout, y compris le mot vide. Les mots de longueur maximale 14 sont au nombre de quatre : $aabaababbabbaa$, $abbabbabaabaab$ et les deux mots obtenus en permutant a et b. ∎

Lemme 3.2. *La formule $ABAABA + AAA$ est 2-inévitable.*

Preuve : Comme au lemme 3.1, on trouve par *backtracking* qu'il y a 6693 mots binaires qui évitent $ABAABA + AAA$, dont deux de longueur maximale 56 :

$$aabbababbaababaabbababbaabaabbabbaababaabbababbaababaabb$$

et son symétrique. ∎

Lemme 3.3. *Si f est un morphisme tel que $f(a) \in a\Sigma^+$ et $f(b) \in \Sigma^+$, avec $\Sigma = \{a, b\}$, alors soit $L(\Sigma, f, a)$ est sans cube, soit il contient des cubes autres que b^3.*

Preuve : Supposons que b^3 est le seul cube dans $F(L(\Sigma, f, a))$. On a alors nécessairement $f(b) = b$, car sinon $F(L(\Sigma, f, a))$ contiendrait $f(b)^3$. La dernière lettre de $f(a)$ doit donc être un a, car sinon f serait *pushy* et $F(L(\Sigma, f, a))$ contiendrait des cubes arbitrairement grands (par exemple, $f^6(a)$ se terminerait par $(bb)^3$). Soit k le nombre de b entre les deux premières occurrences de a dans $f(a)$, de sorte que ab^ka est un préfixe de $f(a)$, et l le nombre de b entre les deux dernières occurrences de a dans $f(a)$. Si $k \leqslant l$, alors $f^2(a)$ contient $f(ab^ka)$ qui contient $ab^{l-k}(b^ka)^3$. Si $k \geqslant l$, alors $f^2(a)$ contient $f(ab^la)$ qui contient $(ab^l)^3b^{k-l}a$. Dans les deux cas, on a trouvé un cube différent de aaa dans $F(L(\Sigma, f, a))$. ∎

Lemme 3.4. *Soit f un morphisme tel que $f(a) \in a\Sigma^+$ et $f(b) \in \Sigma^+$, avec $\Sigma = \{a, b\}$. Si $F(L(\Sigma, f, a))$ contient un cube autre que b^3, alors il n'évite ni $ABAABA$ ni $AABBA$.*

Preuve : Soit k un entier assez grand pour que $f^k(a) = uv^3w$ avec $v \in \Sigma^* \setminus \{\varepsilon, b\}$. Si $v = b^m$ avec $m \geqslant 2$, alors $f^k(a)$ contient b^6 et b^5, images des motifs étudiés par $(A \mapsto b, B \mapsto b)$. Sinon, v contient au moins un a; posons $v = sat$, si bien que $f^k(a) = usatsatsatw$. Alors

$$f^{2k}(a) = f^k(us)uv^3wf^k(ts)uv^3wf^k(ts)uv^3wf^k(tw)$$

qui contient l'image de $ABAABA$ par $(A \mapsto v, B \mapsto wf^k(ts)uv)$ et l'image de $AABBA$ par $(A \mapsto v, B \mapsto vwf^k(ts)uv^2)$. ∎

Théorème 3.3. *les motifs $ABAABA$ et $AABBA$ sont 2-DOL-inévitables.*

Preuve : Si l'un de ces motifs était 2-DOL-évitable, il existerait d'après le théorème 3.2 un morphisme f de Σ^*, avec $\Sigma = \{a, b\}$, $f(a) \in a\Sigma^+$ et $f(b) \in \Sigma^+$, tel que le DOL-langage $L(\Sigma, f, a)$ évite soit $ABAABA$ soit $AABBA$. D'après les lemmes 3.1 et 3.2, ce DOL-langage ne peut pas éviter AAA, donc il contient (comme facteurs) des cubes autres que b^3 d'après le lemme 3.3, et le lemme 3.4 donne une contradiction. ∎

Nous pouvons maintenant répondre à la question de P. Roth mentionnée au début du chapitre 3 :

Corollaire 3.2. *Le motif $ABAAB$ est 2-DOL-inévitable.*

Preuve : $ABAAB$ divise le motif $ABAABA$, qui est 2-DOL-inévitable. ∎

Le théorème 3.3 se généralise facilement à tous les motifs qui sont simultanément 2-inévitables avec AAA et qui divisent le motif $(ABBBC)^3$, ou plus généralement l'un des motifs C_n définis comme suit, que nous appellerons *multicubes* : $C_1 = A_1^3$ et $C_{n+1} = (A_{2n}C_nA_{2n+1})^3$ pour tout $n \geqslant 1$ (le motif C_n a $2n-1$ variables A_1, ..., A_{2n-1}, et sa longueur est $2.3^n - 3$).

L'étude des motifs ternaires donne lieu à d'autres preuves de DOL-inévitabilité, sur le même principe, mais parfois avec d'autres motifs auxiliaires que AAA. La seule difficulté est de démontrer que si un DOL-langage contient le motif auxiliaire, alors il le contient avec des images de variables de longueur arbitrairement grande (i.e. il contient des l-occurrences du motif pour tout l, cf. 1.6.2). Cela va de soi si le morphisme est expansif, mais pas s'il possède des lettres de rang 0 : pour l'alphabet binaire, le seul cas qui pose problème est celui des morphismes tels que $f(a) \in a\Sigma^+a$ et $f(b) = b$.

Ainsi, $ABBCABBC$ est 2-DOL-inévitable, car tout DOL-langage binaire infini contient des carrés arbitrairement longs, et tout carré binaire de longueur au moins 12 contient $ABBCABBC$. De même, $ABACABAC$ est 2-DOL-inévitable car tout carré binaire de longueur au moins 22 est divisible par $ABACABAC$.

On peut également utiliser plusieurs motifs auxiliaires (c'est-à-dire une formule) : ainsi, le motif $ABAACCB$ est simultanément 2-inévitable avec les chevauchements $AAA + ABABA$ (1401 mots évitent ces trois motifs, longueur maximale 33). Par la même méthode que pour le lemme 3.3, on montre qu'un DOL-langage binaire qui contient des chevauchements contient nécessairement des chevauchements autres que des puissances de b. Plus précisément, soit le DOL-système est expansif et le langage contient $h(ABABA)$ avec $|h(A)|$ arbitrairement long, soit $f(b) = b$ et le langage contient des cubes arbitrairement longs : dans les deux cas, il existe un entier k tel que $h^k(a) = usatrsatrsatw$.

Alors pour tout $n \geqslant 1$, le mot $h^{nk}(a)$ est divisible par le motif O_n, défini par $O_1 = A_1B_1A_1B_1A_1$ et $O_{n+1} = A_{2n}O_nA_{2n+1}B_{n+1}A_{2n}O_nA_{2n+1}B_{n+1}A_{2n}O_nA_{2n+1}$ pour tout $n \geqslant 1$ (le motif O_n a $3n-1$ variables A_1, ..., A_{2n-1}, B_1, ..., B_n, et sa longueur est $3^{n+1}-4$). Nous appellerons naturellement *multichevauchements* ces motifs O_n (ou plutôt les formules $C_n + O_n$, mais on notera que O_n divise C_{n+1} donc il suffit de montrer que les motifs étudiés divisent l'un des O_n pour en déduire qu'ils divisent un multicube). Comme O_3 est lui-même divisible par $ABAACCB$, on peut donc conclure que $ABAACCB$ est 2-DOL-inévitable. En utilisant une propriété des chevauchements dont nous reparlerons au chapitre 5, on obtient le critère suivant :

Théorème 3.4. *Soit p un motif non évité par la suite de Thue-Morse, tel que p divise l'un des multichevauchements O_n. Alors p est 2-DOL-inévitable.*

Preuve : La preuve de ce théorème repose sur le fait que tout mot biinfini sans chevauchement a exactement les mêmes facteurs que la suite de Thue-Morse, qui se démontre par exemple à partir du lemme 5.1. Si la formule $p + AAA + ABABA$ était 2-évitable, il existerait une infinité de mots sans chevauchement évitant p, et donc un mot biinfini, ce qui n'est pas possible puisque p n'est pas évité par la suite de Thue-Morse. Par conséquent cette formule n'est pas 2-évitable, donc s'il existait un DOL-langage évitant p, ce langage devrait contenir des chevauchements, et donc comme on l'a vu plus haut devrait contenir O_n, qui lui-même contient p, et on aboutit à une contradiction. ■

Au moyen du théorème 3.4, on retrouve facilement que $ABAABA$ est 2-DOL-inévitable.

Les preuves de 3-DOL-inévitabilité sont beaucoup plus difficiles, car il faut tenir compte d'un plus grand nombre de cas de non-expansivité; si $f\colon \Sigma^* \to \Sigma^*$ est non expansif, avec $\Sigma = \{a, b, c\}$, cinq cas sont à considérer après application du théorème 3.2, à permutation des lettres et miroir près :

<u>cas 1 :</u> $f(a) \in a\Sigma^+a$, $f(b) = b$, $f(c) = c$;

<u>cas 2 :</u> $f(a) \in a\Sigma^+a$, $f(b) \in b\Sigma^+b$, $f(c) = c$;

<u>cas 3 :</u> $f(a) \in a\Sigma^+a$, $f(b) \in b\Sigma^+a$, $f(c) = c$;

<u>cas 4 :</u> $f(a) \in a\Sigma^+a$, $f(b) \in a\Sigma^+a$, $f(c) = c$;

<u>cas 5 :</u> $f(a) \in a\Sigma^+b$, $f(b) \in a\Sigma^+b$, $f(c) = c$.

3.6 Nouvelle classification des motifs binaires

En tenant compte de la DOL-évitabilité, nous pouvons préciser notre classification des motifs binaires.

Théorème 3.5. *Les motifs binaires p se répartissent dans les quatre catégories suivantes, en fonction du couple* $(\alpha(p), \alpha^{DOL}(p))$ *:*

- *Motifs inévitables* (∞, ∞) *:*

  ε, A, AB, ABA, B, BA, BAB.

- *Motifs 2-inévitables 3-DOL-évitables* $(3, 3)$ *:*

  AA, AAB, $AABA$, $AABAA$, $AABAB$, $AABB$, $ABAA$, $ABAB$, $ABABB$, ABB, $ABBA$, BAA, $BAAB$, $BABA$, $BABAA$, $BABB$, BB, BBA, $BBAA$, $BBAB$, $BBABA$, $BBABB$.

- *Motifs 2-évitables 2-DOL-inévitables 3-DOL-évitables* $(2, 3)$ *:*

  $AABBA$, $ABAAB$, $ABAABA$, $ABBAA$, $ABBAB$, $BAABA$, $BAABB$, $BABBA$, $BABBAB$, $BBAAB$.

- *Motifs 2-DOL-évitables* $(2, 2)$ *: tous les autres.*

Preuve : La seule différence entre ce théorème et le théorème 2.4 et la distinction entre les motifs 2-évitables qui sont 2-DOL-évitables et ceux qui ne le sont pas.

Les motifs de la troisième catégorie sont 2-DOL-inévitables d'après le théorème 3.3 et le corollaire 3.2. On montre que ce sont les seuls par *backtracking* en utilisant les motifs connus pour être 2-DOL-évitables, en particulier $AABAAB$ et $AABBAB$ étudiés en 2.6. ■

Nous pouvons modifier en conséquence l'arbre des motifs binaires, pour y faire figurer tous les motifs 2-DOL-inévitables (figure 3.1).

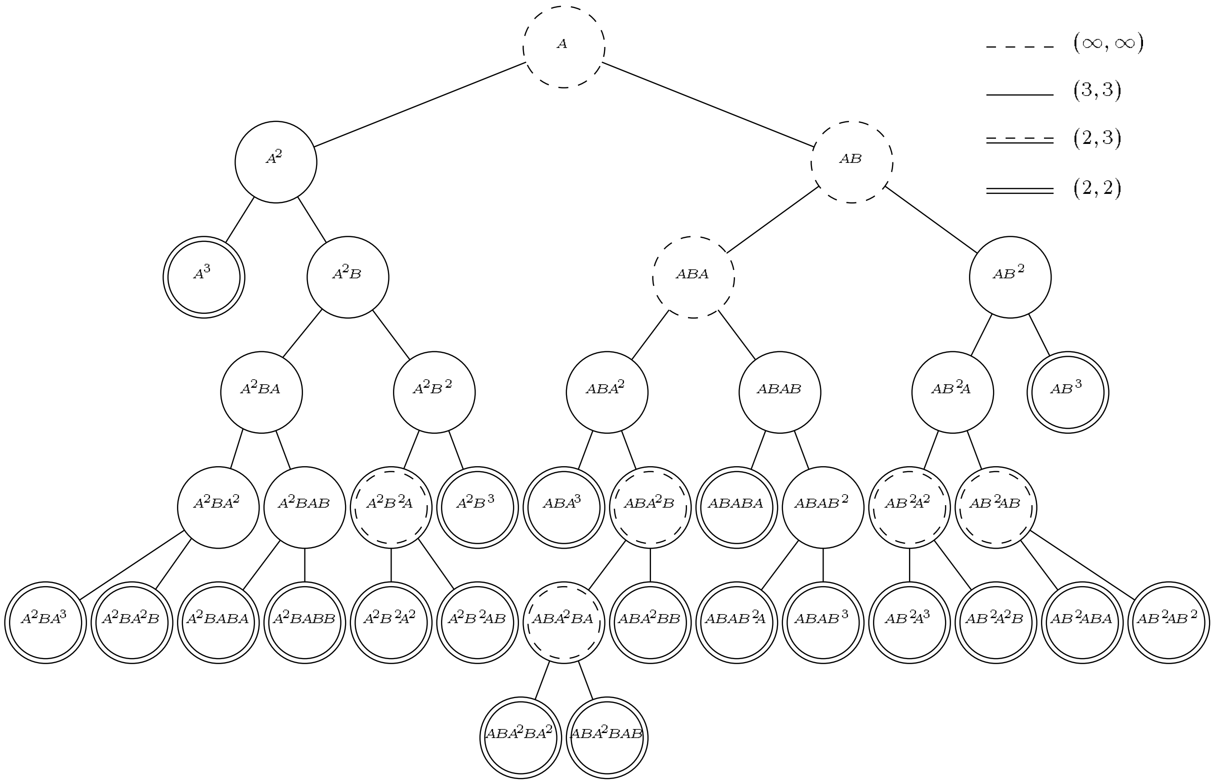

FIG. 3.1 - *Classification des motifs binaires selon* $(\alpha(p), \alpha^{DOL}(p))$

Chapitre 4

Finitude de l'ensemble des motifs n-aires k-inévitables ($k \geqslant 2$)

Ursula Schmidt [56] a démontré que tous les motifs binaires de longueur 13 ou plus sont 2-évitables (Peter Roth [53] a ensuite ramené cette longueur à 6). Cela veut dire qu'il y a un nombre fini de motifs binaires 2-inévitables, et donc de motifs binaires k-inévitables pour $k \geqslant 2$ puisque tout motif k-inévitable est aussi 2-inévitable. On peut se demander si cette propriété est vraie pour n'importe quel nombre de variables : l'alphabet Δ étant fixé, y a-t-il dans Δ^* un nombre fini de motifs 2-inévitables, et si oui peut-on borner leur longueur en fonction de la taille n de Δ ?

Dans ce chapitre, nous répondons par l'affirmative à cette question, au moyen de deux constructions différentes. La première utilise des DOL-langages et donne une borne en $O\left(48^{2^n}\right)$; la seconde, avec des HDOL-langages, permet d'améliorer cette borne à $O\left(5^n\right)$

Cette étude a été réalisée en collaboration avec Peter Roth.

4.1 Introduction : ℓ_{nk}

Pour commencer, nous allons résumer les résultats connus concernant la longueur des motifs k-inévitables à n variables. La notation suivante nous sera utile :

Notation: On note ℓ_{nk} le plus petit entier l (ou ∞ s'il n'y a pas d'entier ayant cette propriété) tel que tout motif n-aire de longueur l est k-évitable. Par convention, k peut également être ∞, auquel cas « k-évitable » signifie « évitable ».

Plus généralement, si, pour tout alphabet Σ, $\mathscr{L}(\Sigma)$ est une famille de langages sur Σ (en pratique $\mathscr{L}$ sera DOL ou HDOL, ou la famille de tous les langages quand $\mathscr{L}$ n'est pas indiqué), on note $\ell_{nk}^{\mathscr{L}}$ le plus petit entier l tel que tout motif n-aire de longueur l est $\mathscr{L}(\Sigma_k)$-évitable, avec les mêmes conventions pour ∞.

Il est clair que

$$\ell_{nk}^{\mathscr{L}} \leqslant \ell_{n'k'}^{\mathscr{L}'} \tag{4.1}$$

si $n \leqslant n'$, $k \geqslant k'$ et $\mathscr{L}' \subset \mathscr{L}$. Il est aussi bien connu (cf. proposition 1.8 et [65]) que $\ell_{n\infty} = \ell_{n\infty}^{\mathrm{HDOL}} = \ell_{n\infty}^{\mathrm{DOL}} = 2^n$. Le résultat d'Ursula Schmidt [56] s'écrit simplement $\ell_{22} \leqslant 13$, celui de Peter Roth [53] $\ell_{22} = 6$, et celui de Silke Nilgens [47] $\ell_{33} = 8$.

Les questions que l'on peut se poser sont : est-ce que tous ces nombres sont finis ? Que valent-ils exactement, ou à défaut peut-on les encadrer ? Quel est le plus petit k tel que $\ell_{nk} = \ell_{n\infty}$?

Cette dernière question est à relier au problème de trouver des motifs évitables k-inévitables, qu'on ne sait pas résoudre pour $k > 3$, et comme le motif trouvé par Baker, McNulty et Taylor [5] pour $k = 3$ a 7 variables et est bien plus court que 2^7, on n'a aucun moyen de dire que ℓ_{nk} n'est pas égal à 2^n pour $k > 2$.

Si on sait montrer que tous les motifs à n variables sont évitables sur un alphabet à N_n lettres, on peut en déduire que $\ell_{nk} = 2^n$ pour $k \geqslant N_n$. La construction de [5] donne $N_n \leqslant 4(n+2)\lceil \log(n+2) \rceil$, et Mel'ničuk [30] a montré que l'on peut prendre $N_n = 4\left\lceil \frac{n+1}{2} \right\rceil$.

Dans les sections suivantes, nous allons montrer que ℓ_{nk}, $\ell_{nk}^{\mathrm{HDOL}}$ et ℓ_{nk}^{DOL} sont finis pour tout n et tout $k \geqslant 2$. D'après l'inégalité (4.1) il suffit d'étudier ℓ_{n2}^{DOL}. Nous étudierons aussi $\ell_{n2}^{\mathrm{HDOL}}$, qui donne une meilleure borne.

4.2 Les grands motifs n-aires sont 2-DOL-évitables

Soit n un entier supérieur ou égal à 2. On considère les motifs sur l'alphabet $\Delta = \{X_1, \ldots, X_n\}$.

Proposition 4.1. *Supposons donnés $n-1$ entiers $B_1, \ldots, B_{n-1}$ avec $B_i \geqslant 2$. Soit*

$$M = 4B_{n-1} + 3B_{n-2} + 4B_{n-3} + \cdots + iB_{n+1-i} + \cdots + nB_1 + \frac{3n^2+7n+10}{2}$$

et soit P l'ensemble des motifs non vides $p \in \Delta^*$ tels que toute lettre qui apparaît dans p y apparaît au moins M fois et que, pour tout k compris entre 0 et $n-2$, tout facteur q de p de longueur B_k contient toutes les lettres qui figurent dans p, sauf au plus $n-1-k$.

Soit f le morphisme de Σ^* dans lui-même défini comme suit :

$$\begin{array}{rrcl} f: & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & a^{B_{n-1}+2}b^n z_0 b^{n+2} z_1 b^{n+4} \dots z_{n-3} b^{3n-4} z_{n-2} b^{3n-2} a b^{3n} \\ & b & \longmapsto & a^{B_{n-1}+2}b^{n+1} z_0 b^{n+3} z_1 b^{n+5} \dots z_{n-3} b^{3n-3} z_{n-2} b^{3n-1} a b^{3n+1} \end{array}$$

où $\Sigma = \{a, b\}$ et $z_k = \left(ab^{k+1}\right)^{B_{n-1-k}-1} a$.

Alors le DOL-langage $L(\Sigma, f, a)$ évite tous les motifs $p \in P$.

Remarque : Notons que si on prend $p \in P$ et qu'on supprime toutes les occurrences des variables dans une certaine partie de Δ, alors le motif obtenu est soit le mot vide, soit un élément de P.

Notons également que la valeur de M a été choisie de telle sorte que

$$M = \max(|f(a)|, |f(b)|) + B_{n-1} + 1 \ .$$

Preuve : Supposons qu'au contraire il existe un motif $p \in P$ qui ne soit pas évité par ce langage. Soit n' le nombre de variables différentes de p : on choisit p de sorte que n' soit minimal. Soit Δ' l'ensemble des variables qui apparaissent dans p. Soit k le plus petit entier tel que $f^k(a)$ contient p. On a alors $f^k(a) = w_1\varphi(p)w_2$, pour un certain morphisme non-effaçant $\varphi\colon \Delta'^* \to \Sigma^*$. Soit Δ'' l'ensemble des variables $Y \in \Delta'$ telles que $\varphi(Y)$ contient aa.

Comme $|\varphi(p)| \geqslant |p| \geqslant M$, le mot $\varphi(p)$ contient nécessairement $a^{B_{n-1}+2}$ (qui apparaît au début de chaque image de lettre par f), donc soit l'ensemble Δ'' est non vide, soit il y a B_{n-1} variables consécutives dans p dont l'image par φ est simplement a. Dans ce dernier cas, puisque par hypothèse toute variable $Y \in \Delta'$ apparaît dans tout facteur de longueur B_{n-1} de p, on a $\varphi(Y) = a$ pour tout Y. Alors $\varphi(p) = a^{|p|}$, ce qui est clairement impossible.

Définissons maintenant une fonction $h\colon \Delta' \to \{0, \dots, 3n+1\}^*$ de la manière suivante : pour $Y \in \Delta'$, on écrit $\varphi(Y)$ sous la forme $b^{j_0}ab^{j_1}a \dots ab^{j_m}$, avec $0 \leqslant j_i \leqslant 3n+1$, et on pose $h(Y) = j_1 j_2 \dots j_{m-1}$ (qui peut être le mot vide si $m \leqslant 1$).

Définition : Appelons *coupure* un couple de mots (u_1, u_2) tels que $u_1u_2 = f(a)$ (ou $f(b)$) et qu'il existe (p_1, p_2) et (v_1, v_2), avec $p_1p_2 = p$, $w_1\varphi(p_1) = f(v_1)u_1$ et $\varphi(p_2)w_2 = u_2 f(v_2)$. Cette coupure est appelée *coupure initiale* quand $h(Y) = \varepsilon$ pour tout Y figurant dans p_1.

Si u est un facteur figurant une seule fois dans $f(a)$, on dit que u est *coupé* s'il existe une coupure non initiale (v_1u_1, u_2v_2) où u_1 et u_2 sont des mots non vides tels que $u_1u_2 = u$.

La suite de la preuve de la proposition 4.1 repose sur le lemme suivant :

Lemme 4.1. *Il existe un entier r dans l'ensemble $\{n, n+2, \dots, 3n\}$ tel que ab^ra n'est pas coupé, et de même il existe un entier s dans l'ensemble $\{n+1, n+3, \dots, 3n+1\}$ tel que ab^sa n'est pas coupé.*

Preuve : Supposons au contraire que tous les ab^ra sont coupés. Soit j ($0 \leqslant j \leqslant n$) tel qu'il n'existe pas de variable $Y \in \Delta'$ pour laquelle $h(Y)$ se termine par j, et minimal pour cette propriété. Il existe un tel j, car les $n+1$ valeurs $0, \ldots, n$ ne peuvent pas toutes apparaître à la fin des n' mots $h(Y)$. Soit F l'ensemble des $Y \in \Delta'$ telles que le dernier symbole de $h(Y)$ n'est pas dans l'ensemble $\{0, 1, \ldots, n\}$ (cela inclut le cas $h(Y) = \varepsilon$). Cet ensemble F possède au plus $n' - j$ éléments, car il y a au moins j variables différentes $Y \in \Delta'$ pour lesquelles $h(Y)$ se termine par un élément de $\{0, 1, \ldots, j-1\}$. Le mot $ab^{n+2j}a$ est coupé, on peut donc écrire $w_1\varphi(p_1) = f(v_1)u_1ab^{i_1}$ et $\varphi(p_2)w_2 = b^{i_2}au_2f(v_1)$ avec $p = p_1p_2$, $i_1 + i_2 = n + 2j$ et $u_1ab^{n+2j}au_2 = f(a)$. Soit Y la dernière variable de p_1 telle que $h(Y) \neq \varepsilon$ (elle existe car la coupure n'est pas initiale), et $Y_1, \ldots, Y_m$ les variables qui suivent Y dans p_1. Pour $1 \leqslant i \leqslant m$, le mot $\varphi(Y_i)$ contient au plus un a, et u_1 se termine par $ab^{n+2j-2}(ab^j)^{B_{n-j}-1}$ (ou $u_1 = a^{B_{n-1}+1}$ si $j = 0$), par conséquent si $m < B_{n-j} - 1$ (ou $m \leqslant B_{n-1}$ si $j = 0$), les deux derniers a dans $\varphi(Y)$ sont séparés par b^j et $h(Y)$ se termine par le symbole j, une contradiction. D'un autre côté $m \geqslant B_{n-j}$ (ou $m \geqslant B_{n-1}$ si $j = 0$) est impossible d'après la définition de P car $Y_1 \ldots Y_m$ contient au plus $n' - j$ lettres différentes ($n' - 1$ si $j = 0$). Le seul cas restant est $j \neq 0$, $m = B_{n-j} - 1$; alors $h(Y)$ se termine par $n + 2j - 2$ ou j, donc $Y \in F$, et $YY_1 \ldots Y_m$ contredit à nouveau la définition de P. ■

Soit maintenant ψ l'application de Δ' dans Σ^* définie par $\psi = g \circ h$, où $g(r) = a$, $g(s) = b$ et $g(j) = \varepsilon$ pour toutes les autres valeurs de j, les entiers r et s étant donnés par le lemme 4.1. On prolonge ψ en un morphisme de Δ'^* dans Σ^*. Alors $\psi(p)$ est facteur de $f^{k-1}(a)$: en effet, chaque occurrence de $f(a)$ dans $\varphi(p)$ est marquée par le mot non coupé ab^ra, et correspond donc à un a dans $\psi(p)$, et de même chaque occurrence de $f(b)$ dans $\varphi(p)$ correspond à un b dans $\psi(p)$. Soit p' le motif obtenu en effaçant dans p les variables Y pour lesquelles $\psi(Y) = \varepsilon$. Comme p contient au moins une variable dont l'image par φ contient aa, et comme cette variable apparaît au moins M fois dans p, on a $|\varphi(p)| \geqslant 2M$, donc $\varphi(p)$ contient $f(a)$ ou $f(b)$, dans lesquels il y a une occurrence de ab^ra ou ab^sa, et il ne peut pas y avoir de coupure initiale dans ce mot à cause du aa correspondant à un élément de Δ'' qui apparaît quelque part avant ce mot. Cela veut dire que $\psi(p) \neq \varepsilon$, et donc que p' n'est pas vide. Si $p' = p$, k n'est pas minimal; si $p' \neq p$, la taille de l'alphabet des variables de p n'est pas minimale. Dans les deux cas on aboutit donc à une contradiction, ce qui achève la preuve de la proposition 4.1. ■

Nous pouvons maintenant démontrer que les nombres ℓ_{k2}^{DOL} sont tous finis, et en calculer un majorant :

Théorème 4.1. *Pour tout entier $n \geqslant 1$, il y a un nombre fini de motifs à n variables 2-inévitables. Plus précisément, tous les motifs à n variables de longueur supérieure ou égale à $\frac{1}{16}48^{2^{n-1}}$ sont 2-DOL-évitables.*

Preuve : Supposons que nous avons prouvé que pour tout $k < n$, ℓ_{k2}^{DOL} est fini. Posons $B_k = \ell_{k2}^{\mathrm{DOL}}$ pour $1 \leqslant k \leqslant n-1$, et appliquons la proposition 4.1 : un motif p de longueur MB_{n-1} à n variables est soit dans P, et alors il est évité par $L(\Sigma, f, a)$, soit contient un facteur p' de longueur B_k à au plus k variables. Dans les deux cas nous pouvons conclure que p est 2-DOL-évitable. Par conséquent, $\ell_{n2}^{\mathrm{DOL}} \leqslant MB_{n-1}$.

Soit (L_n) la suite définie par récurrence comme suit : $L_1 = 3$ et

$$L_n = L_{n-1}\left(2L_{n-1} + \sum_{k=2}^{n} kL_{n-k+1} + \frac{3n^2+7n+10}{2}\right) .$$

Il est clair que $\ell_{n2}^{\mathrm{DOL}} \leqslant L_n$ pour tout $n \geqslant 1$.

On voit que $L_n > 2L_{n-1}$, donc $L_{n-k+1} < 2^{-k+2}L_{n-1}$. Comme de plus

$$\frac{3n^2+7n+10}{2} \leqslant 9.2^{n-1} \leqslant 6L_{n-1} ,$$

on a

$$L_n < L_{n-1}^2\left(2 + \sum_{k=2}^{n} 2^{-k+2}k + 6\right) < 16L_{n-1}^2 .$$

Par conséquent $\ell_{n2} \leqslant \ell_{n2}^{\mathrm{DOL}} \leqslant L_n < \frac{1}{16}48^{2^{n-1}}$. ∎

Remarque : Il est bien entendu possible d'améliorer considérablement cette borne, vu la grossièreté des approximations effectuées. Elle resterait toutefois de la forme $O\left(\alpha^{2^n}\right)$.

4.3 Une meilleure borne avec des HDOL-systèmes

Dans [5], Baker, McNulty et Taylor construisent un mot infini W_n sur un alphabet à N_n lettres, où $N_n \leqslant 4(n+2)\lceil\log(n+2)\rceil$, qui évite tous les motifs évitables à au plus n variables. Ce mot est obtenu comme point fixe d'un morphisme, donc il est équivalent à un DOL-système. Il est construit par le procédé suivant : soit m et z des entiers tels que $z > 0$ et $m > (n+1)z$, et $N_n = mb$ avec $b = \left\lceil m^{1/z}\right\rceil$ (on peut choisir m et z de façon à avoir l'inégalité annoncée cidessus, à savoir $z = \lceil\log(r+2)\rceil$ et $m = z(r+2)$). Le mot infini $W_n = \Omega_{m,z}$ est le point fixe du morphisme $h_{m,z} : \Sigma^* \to \Sigma^*$, uniforme de longueur m sur l'alphabet $\Sigma = \{a_0, a_1, \ldots, a_{N_n-1}\}$, défini par $h(a_i) = a_{md_0^i+0}a_{md_1^i+1}a_{md_2^i+2}\cdots a_{md_{m-1}^i+m-1}$ où d_0^i, d_1^i,..., d_{z-1}^i sont les chiffres du développement de i en base b, en commençant par le chiffre le moins significatif, la suite d_j^i étant prolongée cycliquement par $d_{j+z}^i = d_j^i$.

Comme dans la section précédente, nous allons pour tout n construire un morphisme dépendant de $n-1$ entiers B_k, mais ce morphisme servira cette fois à coder le mot W_n sur l'alphabet binaire, et non à construire un DOL-langage.

Proposition 4.2. *Supposons donnés $n-1$ entiers $B_1, \ldots, B_{n-1}$ supérieurs ou égaux à 2. Soit*

$$M = 4B_{n-1} + 3B_{n-2} + 4B_{n-3} + 5B_{n-4} + \cdots + nB_1 + T_n$$

avec $T_n = \frac{(N_n-1)(n+1)(n+2)}{2} + n^2 + 2n + 4$, et soit P l'ensemble des motifs non vides p de longueur M ou plus tels que, pour tout k compris entre 1 et $n-1$, tout facteur q de p de longueur B_k contient au moins $k+1$ variables distinctes.

Considérons le morphisme :

$$\begin{array}{rccl} g: & \Sigma^* & \longrightarrow & \Sigma'^* \\ & a_k & \longmapsto & a^{B_{n-1}+2} b^{n+k} z_0 b^k z_1 b^k \ldots z_{n-3} b^k z_{n-2} b^k a b^{n+nN_n+k} \end{array}$$

où $\Sigma = \{a_0, \ldots, a_{N_n-1}\}$, $\Sigma' = \{a, b\}$ et $z_k = \left(ab^{k+1}\right)^{B_{n-1-k}-1} ab^{n+(k+1)N_n}$.

Alors le mot infini $g(W_n)$ évite tous les motifs $p \in P$.

Remarque : Si on prend $p \in P$ et que l'on supprime toutes les occurrences des variables d'un certain sous-ensemble de Δ, alors on obtient soit le mot vide soit un motif qui n'est pas nécessairement dans P mais qui est toujours évitable (par la proposition 1.7) car il ne contient pas de variable isolée : puisque $M \geqslant 4B_{n-1}$, toute variable apparaît au moins quatre fois.

La valeur de M a été choisie de telle sorte que $M = \max(|g(a_k)|) + B_{n-1} + 1$.

Preuve : La preuve est similaire à celle de la proposition 4.1, et n'est pas détaillée ici. ■

Comme à la section précédente, cette proposition permet de montrer que

Théorème 4.2. *La longueur ℓ_{n2}^{HDOL} à partir de laquelle tous les motifs à n variables sont 2-HDOL-évitables (et donc 2-évitables) est bornée par*

$$\ell_{n2} \leqslant \ell_{n2}^{HDOL} = O\left(\lambda^n\right)$$

pour toute base $\lambda > \frac{1}{1-2^{-\frac{1}{3}}} \simeq 4,847322102$ (par exemple $\lambda = 5$).

Preuve : Supposons que nous avons prouvé que pour $k < n$, tout motif de longueur S_k à k variables est évité par $h(W_j)$ pour un certain entier j ($1 \leqslant j \leqslant k$) et un certain morphisme h. Soient $B_1 = S_1, B_2 = S_2, \ldots, B_{n-1} = S_{n-1}$. Alors un motif p de longueur $S_n = M$ à n variables est soit dans P (auquel cas, d'après la proposition 4.2, il est évité par $g(W_n)$), soit contient un facteur p' de longueur B_k à au plus k variables, qui est évité par un certain $h(W_j)$.

Nous avons montré que tout motif à n variables de longueur S_n ou plus est 2-HDOL-évitable, où la suite (S_n) est définie par $S_1 = 3$ et

$$S_n = 2S_{n-1} + \sum_{k=2}^{n} kS_{n-k+1} + T_n \quad .$$

Soit $\lambda > \frac{1}{1-2^{-\frac{1}{3}}}$ (de telle sorte que $K = \left(1 - \frac{1}{\lambda}\right)^{-2} + \frac{2}{\lambda} - 1 < 1$) et

$$C = \max\left(\frac{3}{\lambda}, \frac{1}{1-K} \max_{n \geqslant 1}\left(\frac{T_n}{\lambda^n}\right)\right) \quad .$$

Alors $S_n \leqslant C\lambda^n$, car $3 = S_1 \leqslant C\lambda$ et si $S_k \leqslant C\lambda^k$ pour $1 \leqslant k < n$,

$$S_n \leqslant C\lambda^n \left(\frac{2}{\lambda} + \sum_{k=2}^{n} k\lambda^{1-k} + \frac{T_n}{C\lambda^n}\right)$$

avec $\frac{2}{\lambda} + \sum\limits_{k=2}^{n} k\lambda^{1-k} < K$ et $\frac{T_n}{C\lambda^n} \leqslant 1 - K$. On remarquera que l'expression de N_n, et donc de T_n, n'influe que sur la valeur de la constante C, pourvu que $N_n = o(n^{-2}\lambda^n)$. ■

4.4 Valeurs connues de ℓ_{nk}

Le tableau 4.1 résume les valeurs connues de ℓ_{nk}. La colonne $n = 1$ est connue depuis A. Thue, la colonne $n = 2$ depuis les travaux de P. Roth (cf. chapitre 2), et la colonne $n = 3$ résulte de notre classification partielle des motifs ternaires (cf. annexe A). La ligne $k = \infty$ résulte de la proposition 1.8. On notera que les valeurs indiquées dans ce tableau sont aussi des valeurs de $\ell_{nk}^{\mathrm{HDOL}}$ puisqu'elles sont toutes obtenues au moyen de DOL-systèmes ou de HDOL-systèmes.

	$n=1$	$n=2$	$n=3$	$n=4$
$k=1$	∞	∞	∞	∞
$k=2$	3	6	12	
$k=3$	2	4	8	
$k=4$	2	4	8	
$k=5$	2	4	8	
$k=\infty$	2	4	8	16

TAB. 4.1 - *Valeurs connues de* ℓ_{nk}

Le tableau 4.2 est l'équivalent du précédent pour ℓ_{nk}^{DOL}. La colonne $n = 1$ est identique, la première différence se situant dans la colonne $n = 2$: d'après le théorème 3.5, $\ell_{22}^{\mathrm{DOL}} = 7$; la colonne $n = 3$ résulte également de la classification partielle des motifs ternaires. Enfin, la ligne $k = \infty$ est identique en vertu de la proposition 3.3.

	$n=1$	$n=2$	$n=3$	$n=4$
$k=1$	∞	∞	∞	∞
$k=2$	3	7	15	
$k=3$	2	4	8	
$k=4$	2	4	8	
$k=5$	2	4	8	
$k=\infty$	2	4	8	16

TAB. 4.2 - *Valeurs connues de* ℓ_{nk}^{DOL}

Finalement, au vu des valeurs de ℓ_{nk} et ℓ_{nk}^{DOL} pour $n = 1, 2, 3$, nous proposons la conjecture suivante :

Conjecture 4.1. *Pour tout* $n \geqslant 1$, $\ell_{n2} = \ell_{n2}^{HDOL} = 3.2^{n-1}$, $\ell_{n2}^{DOL} = 2^{n+1} - 1$, *et pour* $k \geqslant 3$, $\ell_{nk} = \ell_{nk}^{HDOL} = \ell_{nk}^{DOL} = 2^n$.

Il est en tout cas certain que ces valeurs sont des minorants :

Proposition 4.3. *Pour tout* $n \geqslant 1$, $\ell_{n2}^{HDOL} \geqslant \ell_{n2} \geqslant 3.2^{n-1}$, $\ell_{n2}^{DOL} \geqslant 2^{n+1} - 1$, *et pour* $k \geqslant 3$, $\ell_{nk}^{DOL} \geqslant \ell_{nk}^{HDOL} \geqslant \ell_{nk} \geqslant 2^n$.

Preuve : Pour $k \geqslant 3$, cette proposition est une conséquence immédiate de la proposition 1.8 et de l'inégalité (4.1). Pour ℓ_{n2}, comme le motif AA est 2-inévitable, on montre de la même manière qu'à la proposition 1.9 que les motifs R_n, définis par $R_1 = A_1A_1$ et, pour $n > 1$, $R_n = R_{n-1}A_nR_{n-1}$, de longueur $3.2^{n-1} - 1$, sont 2-inévitables, et donc que ℓ_{n2} ne peut être inférieur à 3.2^{n-1}. Enfin, pour ℓ_{n2}^{DOL}, on considère les motifs $K_n = Z_nZ_n$, carrés des sesquipuissances, de longueur $2^{n+1} - 2$. Pour $n \geqslant 2$, le motif K_n est 2-évitable puisqu'il contient $ABAABA$; mais on vérifie que K_n est contenu d'une part dans $\theta^{2n-2}(aa)$, facteur de la suite de Thue-Morse, et d'autre part dans le multichevauchement O_n, donc est 2-DOL-inévitable par application du théorème 3.4. ■

Chapitre 5

Dénombrement des mots évitant un motif : le cas des chevauchements

5.1 Introduction

Quand un motif, ou plus généralement une formule f est évitable sur un alphabet Σ, il est intéressant de pouvoir évaluer la *fonction de complexité* de l'ensemble U des mots de Σ^* qui évitent f, c'est-à-dire la fonction qui à tout entier n associe le nombre u_n de mots de longueur n dans l'ensemble U.

Si f est un ensemble fini de mots constants, le problème revient à compter les mots dans un langage rationnel, ce qui peut se faire par exemple au moyen de fonctions génératrices. Mais dès qu'on introduit des variables, très peu de choses sont connues sur cette suite (u_n). En effet, si f est un motif, on ne sait même pas décider si (u_n) vaudra ultimement 0, c'est-à-dire si f est inévitable sur Σ. Cependant, Brandenburg [10] a prouvé que (u_n) a une croissance exponentielle pour le motif AA sur un alphabet ternaire, et Goralčik et Vaniček [31] ont obtenu le même résultat sur un alphabet binaire pour tous les motifs binaires 2-évitables (le cas de AAA avait déjà été traité par Brandenburg).

Dans ce chapitre, nous allons étudier la croissance de (u_n) dans le cas particulier des *mots binaires sans chevauchement*, c'est-à-dire pour la formule $f = AAA + ABABA$ sur l'alphabet $\Sigma = \{a, b\}$. (On dit qu'un mot contient un *chevauchement* si un de ses facteurs apparaît à deux positions qui se chevauchent, ce qui revient à dire que le mot contient un facteur de la forme $xuxux$ où $x \in \Sigma$ et $u \in \Sigma^*$.)

Après avoir rappelé ce qui est connu de (u_n) (section 5.2), nous décrirons (section 5.3) une bijection entre l'ensemble U des mots sans chevauchement et un langage rationnel. À partir de l'automate reconnaissant ce langage, nous construirons (section 5.4) des relations de récurrence explicites pour la suite

(u_n), qui montrent entre autres que u_n peut être calculé en temps logarithmique. Dans la section 5.5 enfin, nous étudierons les conséquences de ces relations sur le comportement asymptotique de la suite (u_n). En particulier nous montrerons que, bien qu'elle soit encadrée par des expressions polynomiales, la suite (u_n) n'est pas elle-même équivalente à un polynôme.

Nous utiliserons la notation suivante : si x est une lettre de Σ, alors $\overline{x}$ est l'autre élément de Σ. Ainsi, $\overline{a} = b$ et $\overline{b} = a$.

5.2 Résultats antérieurs

En 1912, Axel Thue [62] a montré qu'il existe une infinité de mots binaires sans chevauchement. Pour cela, il a construit un mot infini sans chevauchement, appelé maintenant *suite de Thue-Morse*, qui est le point fixe

$$\theta^\omega(a) = abbabaabbaababbabaababbaabbabaabba\dots$$

du morphisme

$$\begin{array}{rccc} \theta: & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & ab \\ & b & \longmapsto & ba \end{array} .$$

Ce mot infini $\theta^\omega(a)$ est sans chevauchement car le morphisme θ est lui-même sans chevauchement, ce qui signifie que si w est un mot sans chevauchement, alors $\theta(w)$ est aussi sans chevauchement.

Une fois qu'on sait que l'ensemble des mots binaires sans chevauchement est infini, il est naturel d'étudier sa fonction de complexité, c'est-à-dire la suite (u_n) où u_n est le nombre de mots binaires sans chevauchement de longueur n.

La figure 5.2 donne l'allure de la suite u_n, à partir de valeurs obtenues en énumérant tous les mots sans chevauchement de longueur inférieure ou égale à 200, indiquées dans le tableau 5.1.

Ce graphe ressemble beaucoup à ceux de certaines *suites 2-régulières* étudiées par Philippe Dumas dans [24], et on peut donc s'attendre à ce que la suite vérifie une récurrence du type *diviser pour régner*, ou plus généralement une *récurrence mahlérienne*.

Les questions suivantes se posent naturellement : est-il possible de donner un majorant de u_n ? un équivalent ? des relations de récurrence ? Voici un résumé de ce qui est connu jusqu'à présent.

Le nombre de facteurs de longueur n dans la suite de Thue-Morse $\theta^\omega(a)$ fournit un minorant de u_n dont la croissance est linéaire (voir [11, 19]) : pour tout $k \geqslant 0$,

$$\begin{cases} \text{si } 2.2^k \leqslant n \leqslant 3.2^k,\ \theta^\omega(a) \text{ a } 4n - 2.2^k \text{ facteurs de longueur } n+1; \\ \text{si } 3.2^k \leqslant n \leqslant 4.2^k,\ \theta^\omega(a) \text{ a } 2n + 4.2^k \text{ facteurs de longueur } n+1. \end{cases}$$

n	u_n	n	u_n	n	u_n	n	u_n	n	u_n	n	u_n	n	u_n
0	1	30	190	60	486	90	912	120	1198	150	1762	180	2316
1	2	31	196	61	476	91	908	121	1176	151	1756	181	2232
2	4	32	210	62	498	92	960	122	1234	152	1780	182	2324
3	6	33	216	63	500	93	976	123	1224	153	1780	183	2336
4	10	34	224	64	522	94	1008	124	1268	154	1848	184	2380
5	14	35	228	65	528	95	1000	125	1232	155	1824	185	2404
6	20	36	248	66	540	96	1016	126	1278	156	1868	186	2508
7	24	37	272	67	548	97	940	127	1240	157	1800	187	2476
8	30	38	284	68	568	98	942	128	1278	158	1840	188	2552
9	36	39	296	69	560	99	936	129	1292	159	1720	189	2468
10	44	40	300	70	592	100	958	130	1316	160	1716	190	2532
11	48	41	296	71	592	101	924	131	1312	161	1724	191	2420
12	60	42	320	72	620	102	946	132	1356	162	1784	192	2444
13	60	43	332	73	660	103	936	133	1340	163	1808	193	2312
14	62	44	356	74	688	104	922	134	1380	164	1892	194	2322
15	72	45	356	75	688	105	880	135	1392	165	1928	195	2300
16	82	46	376	76	722	106	926	136	1400	166	1996	196	2358
17	88	47	400	77	724	107	908	137	1392	167	2012	197	2304
18	96	48	416	78	740	108	960	138	1468	168	2028	198	2366
19	112	49	380	79	724	109	956	139	1456	169	2012	199	2336
20	120	50	382	80	724	110	986	140	1514	170	2112	200	2330
21	120	51	376	81	716	111	984	141	1444	171	2092		
22	136	52	382	82	748	112	1010	142	1492	172	2180		
23	148	53	356	83	788	113	1044	143	1480	173	2116		
24	164	54	374	84	824	114	1082	144	1524	174	2156		
25	152	55	392	85	816	115	1128	145	1588	175	2120		
26	154	56	410	86	856	116	1170	146	1632	176	2120		
27	148	57	432	87	868	117	1140	147	1692	177	2096		
28	162	58	458	88	880	118	1182	148	1744	178	2164		
29	176	59	464	89	868	119	1192	149	1696	179	2248		

TAB. 5.1 - *Valeurs de* u_n *pour* $0 \leqslant n \leqslant 200$

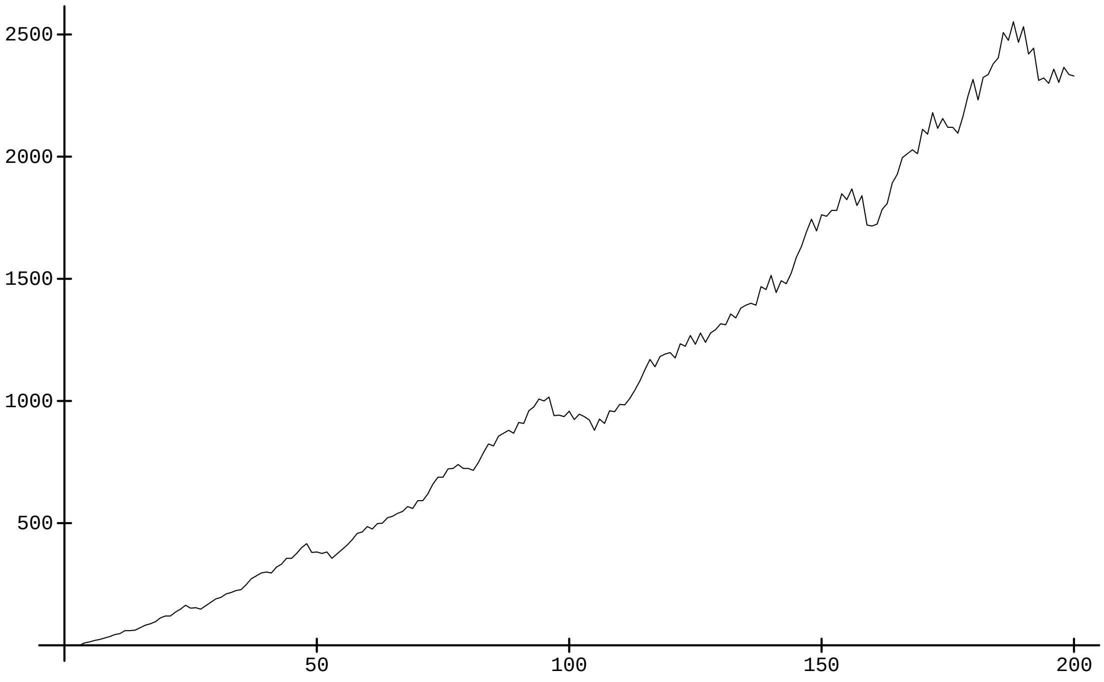

FIG. 5.2 - *La suite u_n pour $n \leqslant 200$*

Restivo et Salemi [50] ont donné un majorant de u_n à croissance polynomiale : $u_n \leqslant Cn^r$ avec $C > 0$ et $r = \log_2 15 \simeq 3,906$.

Kobayashi [36] a amélioré cet encadrement :

$$C_1 n^{1,155} \leqslant u_n \leqslant C_2 n^{1,587} \quad .$$

Le minorant est obtenu en comptant exactement le nombre de mots sans chevauchement qui sont prolongeables à droite en un mot infini (certains mots sans chevauchement, comme par exemple $abbabb$, ne sont pas du tout prolongeables à droite, et il est même probable que le nombre de mots sans chevauchement infiniment prolongeables à droite soit négligeable par rapport au nombre total de mots sans chevauchement de longueur n).

Carpi [13] a montré qu'on pouvait utiliser un automate fini pour calculer u_n, et a donné une méthode pour calculer des majorants de la forme Cn^r, où r est arbitrairement proche de la valeur optimale, en théorie du moins, car il faudrait calculer les normes d'un nombre gigantesque de produits de matrices pour obtenir un résultat intéressant. Carpi n'a publié aucun résultat numérique.

5.3 Une bijection entre le langage des mots binaires sans chevauchement et un langage rationnel

5.3.1 Une décomposition des mots binaires sans chevauchement

Comme nous le verrons plus tard, nous avons besoin de considérer non seulement des mots sans chevauchement, mais aussi des mots qui ne contiennent qu'un seul chevauchement qui les recouvre entièrement. Nous appelons ces mots *chevauchements simples.*

Définition : Un mot $w \in \Sigma^*$ est appelé *chevauchement simple* s'il peut être écrit sous la forme $w = xuxux$ avec $x \in \Sigma$ et $u \in \Sigma^*$ de telle sorte que $xuxu$ et $uxux$ soient sans chevauchement.

De plus, pour éviter le plus possible les cas particuliers, nous nous restreignons aux mots de longueur 4 ou plus. Soient U, V et S les trois ensembles définis comme suit :

U est l'ensemble des mots sans chevauchement de longueur au moins 4 (13 mots sans chevauchement sont exclus);

V est l'ensemble des chevauchements simples de longueur au moins 4 (seuls aaa et bbb sont exclus);

S est l'ensemble fini contenant les éléments de $U \cup V$ de longueur inférieure à 8. L'ensemble S a 76 éléments.

Kfoury [35], Kobayashi [36] et Carpi [13] démontrent des résultats similaires au lemme suivant :

Lemme 5.1. *Tout mot $w \in (U \cup V) \setminus S$ peut s'écrire $w = r_1\theta(u)r_2$ avec u sans chevauchement et $r_i \in \{\varepsilon, a, b, aa, bb\}$. Cette décomposition est unique.* ■

Au moyen de ce lemme, ils construisent des mots sans chevauchement en commençant avec un petit mot et en effectuant alternativement les deux opérations suivantes :

- appliquer le morphisme θ.
- ajouter quelques lettres à gauche ou à droite du mot courant.

Notre approche est similaire, mais au lieu d'ajouter des nouvelles lettres, nous supprimons ou modifions les lettres existantes aux extrémités des mots.

Définition : Soit $E = \{\delta, \iota, \kappa\}$, un ensemble de trois *transformations de base*, appelées respectivement *effacement*, *inversion* et *identité*, et G le monoïde $(E \times E)^*$, dit *monoïde des transformations* (les éléments de g sont appelés *transformations* et ceux de $E \times E$ *transformations simples*). L'élément unité de G est noté 1. Nous allons définir une action à droite de G sur Σ^*.

Pour toute transformation $g \in G$, on pose $\varepsilon.g = \varepsilon$.

Étant donnés $\gamma = (\gamma_1, \gamma_2) \in E \times E$ et $w \in \Sigma^+$, on écrit l'image de w par θ sous la forme $\theta(w) = x_1 u x_2$ avec $u \in \Sigma^*$, $x_1 \in \Sigma$ et $x_2 \in \Sigma$, et $w.\gamma$ est donné par le tableau 5.3 (on rappelle que $\overline{a} = b$ et $\overline{b} = a$).

	$\gamma_2 = \delta$	$\gamma_2 = \iota$	$\gamma_2 = \kappa$
$\gamma_1 = \delta$	u	$u\overline{x_2}$	ux_2
$\gamma_1 = \iota$	$\overline{x_1}u$	$\overline{x_1}u\overline{x_2}$	$\overline{x_1}ux_2$
$\gamma_1 = \kappa$	x_1u	$x_1u\overline{x_2}$	x_1ux_2

TAB. 5.3 - *Action de* (γ_1, γ_2) *sur* w *quand* $\theta(w) = x_1 u x_2$

Par exemple, si $w = aabb$ et $\gamma = (\delta, \iota)$, pour calculer $w.\gamma$ on prend $\theta(w) = ababbaba$, on efface la première lettre a, et on change la dernière en b, ce qui donne $w.\gamma = babbabb$.

Pour tout $w \in \Sigma^*$, on pose $w.1 = w$.

Finalement, pour $g \in G$, $\gamma \in E \times E$ et $w \in \Sigma^*$, on pose $w.(\gamma g) = (w.\gamma).g$.

Lemme 5.2. *Tout mot* $w \in (U \cup V) \backslash S$ *peut s'écrire de manière unique* $w = v.\gamma$ *avec* $\gamma \in E \times E$ *et* $v \in U \cup V$. *De plus,* $|v| < |w|$.

Preuve : Le lemme 5.1 fournit une décomposition unique $w = r_1 \theta(u) r_2$. On définit $v = x_1 u x_2$ et $\gamma = (\gamma_1, \gamma_2)$ comme indiqué dans les tableaux 5.4 et il est aisé de vérifier que $w = v.\gamma$ et $3 < |v| < |w|$ (en fait $\frac{|w|}{2} \leqslant |v| \leqslant \frac{|w|}{2} + 1$).

	$r_1 = \varepsilon$	$r_1 = a$	$r_1 = b$	$r_1 = aa$	$r_1 = bb$
x_1	ε	b	a	b	a
γ_1	κ	δ	δ	ι	ι

	$r_2 = \varepsilon$	$r_2 = a$	$r_2 = b$	$r_2 = aa$	$r_2 = bb$
x_2	ε	a	b	a	b
γ_2	κ	δ	δ	ι	ι

TAB. 5.4 - *Passage du lemme 5.1 au lemme 5.2*

Le mot u est sans chevauchement. Si $x_1 u x_2$ n'était pas dans $U \cup V$, $x_1 u$ commencerait par un chevauchement, ou $u x_2$ se terminerait par un chevauchement. Supposons que $x_1 u$ commence par un chevauchement : alors $\overline{x_1}\theta(u)$ commence également par un chevauchement; ce n'est possible que si $w = \overline{x_1}\theta(u) \in V$, et alors $x_2 = \varepsilon$, donc $x_1 u x_2 = x_1 u \in V$. L'autre cas est symétrique de celui-ci. On notera que le mot v peut être dans l'ensemble V même si w est dans U, par exemple $aabbaabb = babab.(\delta, \delta)$. C'est pourquoi il est nécessaire d'étudier $U \cup V$ et non seulement U.

L'unicité de (v, γ) est une conséquence de l'unicité de (u, r_1, r_2). ∎

Soit Z le sous-ensemble de $S \times G$ qui contient tous les éléments sauf ceux de la forme $(s, \gamma g)$ avec $|s.\gamma| < 8$. À chaque élément $z = (s, g)$ de Z, on associe le mot $[z] = s.g \in \Sigma^*$, dit *mot codé par z*. Inversement, on peut représenter certains mots par des éléments de Z :

Théorème 5.1. *Tout mot $w \in U \cup V$ admet une décomposition unique $w = s.g$ avec $(s, g) \in Z$.*

Preuve : Nous allons démontrer ce théorème par récurrence sur la longueur de w. Si $|w| < 8$, alors $w \in S$ et $(s, g) = (w, 1)$ est la seule possibilité d'après la définition de Z. Supposons maintenant que le théorème a été prouvé pour les mots de longueur strictement inférieure à n; alors si w est un mot de longueur n, selon le lemme 5.2 ce mot s'écrit de manière unique $w = v.\gamma$ avec $\gamma \in E \times E$, $v \in U \cup V$ et $|v| < n$. D'après l'hypothèse de récurrence, v a une décomposition unique $v = s.g$, donc w a aussi une décomposition unique $w = s.(g\gamma)$. ∎

Les éléments de Z peuvent être considérés comme des mots sur l'alphabet $S \cup (E \times E)$, avec seulement leur première lettre dans S. L'ensemble Z est alors clairement un langage rationnel sur cet alphabet : la restriction $(s, \gamma g) \notin Z$ si $|s.\gamma| < 8$ (nécessaire à l'unicité dans le théorème 5.1) revient à définir Z comme

$$Z = (S \times G) \setminus (S_4 \times \{(\delta, \delta), (\delta, \iota), (\delta, \kappa), (\iota, \delta), (\kappa, \delta)\} G)$$

où $S_4 = \{aaba, aabb, abaa, abab, abba, baab, baba, babb, bbaa, bbab\}$ est l'ensemble des éléments de S de longueur 4.

5.3.2 Les langages rationnels qui représentent U et V

Soient $L = \{\, z \in Z \mid [z] \in U \,\}$ et $M = \{\, z \in Z \mid [z] \in V \,\}$. Nous allons montrer que L et M sont des langages rationnels. Pour cela, nous devons répondre à la question : sous quelles conditions (sur z) le mot $[z]$ est-il élément de U ou de V?

Voyons tout d'abord ce qui peut être dit de $[z\gamma] = [z].\gamma$ quand la nature de $[z]$ est connue (en particulier selon que $[z] \in U$, $[z] \in V$ ou $[z] \notin U \cup V$), et quand $\gamma = (\gamma_1, \gamma_2) \in E \times E$ est une transformation simple.

- Si $[z]$ n'est pas élément de $U \cup V$, alors $[z\gamma]$ ne peut pas non plus être élément de $U \cup V$.
- Si $[z] \in U$, $\gamma_1 \in \{\delta, \kappa\}$ et $\gamma_2 \in \{\delta, \kappa\}$, alors $[z\gamma] \in U$.

- Si $[z] \in V$, alors $[z(\iota,\kappa)]$, $[z(\kappa,\iota)]$ et $[z(\kappa,\kappa)]$ ne sont pas éléments de $U \cup V$, $[z(\delta,\kappa)]$ et $[z(\kappa,\delta)]$ sont éléments de V, et $[z(\delta,\delta)]$ est élément de U.
- Si $[z]$ ne commence pas par $aaba$ ou $bbab$, alors $[z(\iota,\gamma_2)] \notin U \cup V$.
- Si $[z]$ ne se termine pas par $abaa$ ou $babb$, alors $[z(\gamma_1,\iota)] \notin U \cup V$.
- Si $[z]$ commence par $aaba$ et soit $\gamma_2 \in \{\delta,\kappa\}$ et $[z] \in U$, soit $\gamma_2 = \delta$ et $[z] \in V$, alors $[z(\iota,\gamma_2)] \in U$: ce mot commence par $bbabbaa$, et s'il contenait un chevauchement, $bbabba$ apparaîtrait à un autre endroit de ce mot, ce qui est impossible. Cela vaut aussi si $[z]$ commence par $bbab$.
- Si $[z]$ se termine par $abaa$ ou $babb$ et soit $\gamma_1 \in \{\delta,\kappa\}$ et $[z] \in U$, soit $\gamma_1 = \delta$ et $[z] \in V$, alors par un argument similaire $[z(\gamma_1,\iota)] \in U$.
- Si $[z]$ commence par $aaba$ ou $bbab$, se termine par $abaa$ ou $babb$, et est élément de $U \cup V$, alors $[z(\iota,\iota)] \in U$.

On voit qu'il est nécessaire de connaître un petit préfixe (au plus quatre lettres) et un petit suffixe de $[z]$ pour pouvoir conclure.

Définition : On définit le *type* d'un mot $w \in U \cup V$ comme étant le couple (i,j) donné par les tableaux 5.5 (comme w ne contient ni aaa ni bbb, le type est toujours bien défini).

$$\begin{cases} i=1 & \text{si } w \text{ commence par} & aba & \text{ou} & bab \\ i=2 & \text{si } w \text{ commence par} & abb & \text{ou} & baa \\ i=3 & \text{si } w \text{ commence par} & aaba & \text{ou} & bbab \\ i=4 & \text{si } w \text{ commence par} & aabb & \text{ou} & bbaa \end{cases}$$

$$\begin{cases} j=1 & \text{si } w \text{ se termine par} & aba & \text{ou} & bab \\ j=2 & \text{si } w \text{ se termine par} & bba & \text{ou} & aab \\ j=3 & \text{si } w \text{ se termine par} & abaa & \text{ou} & babb \\ j=4 & \text{si } w \text{ se termine par} & bbaa & \text{ou} & aabb \end{cases}$$

TAB. 5.5 - *Définition du type d'un mot*

Le nombre i est le *type préfixe* de w, et j est le *type suffixe* de w.

Connaissant le type (i,j) de $[z]$, on peut calculer le type (i',j') de $[z(\gamma_1,\gamma_2)]$ (quand ce mot est dans $U \cup V$) au moyen de la formule

$$(i',j') = (\varphi(i,\gamma_1), \varphi(j,\gamma_2))$$

où φ est la fonction définie par le tableau 5.6 où une case vide signifie que $[z(\gamma_1,\gamma_2)]$ ne peut pas être élément de $U \cup V$.

	1	2	3	4
δ	4	3	1	1
ι			3	
κ	2	2	1	1

TAB. 5.6 - *Action d'une transformation de base sur le type préfixe d'un mot*

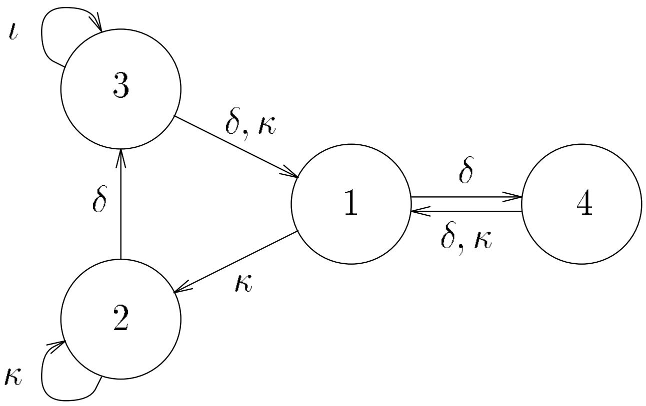

FIG. 5.7 - *Automate à 4 états calculant le type préfixe d'un mot*

Notons que φ n'est rien d'autre que la fonction de transition d'un automate à 4 états (figure 5.7) qui calcule le type préfixe ou le type suffixe de $[z]$. En prenant le produit direct de cet automate par lui-même, on obtient un automate à 16 états qui calcule le type de $[z]$: en effet, il n'y a pas d'interaction entre le début et la fin du mot pour ce qui concerne le type.

Si maintenant nous prenons deux copies de cet automate, une pour L et une pour M, et que nous plaçons des flèches entre ces deux copies en accord avec les règles données au début de cette section, ainsi qu'une flèche pour chaque élément $s \in S$ de l'état initial vers l'état correspondant au type de s (figure 5.8), on obtient un automate qui reconnaît L ou M selon les états finals choisis (figure 5.9). Cet automate a 32 états, plus un état initial, moins quelques états qui peuvent être supprimés car ils ne sont pas accessibles.

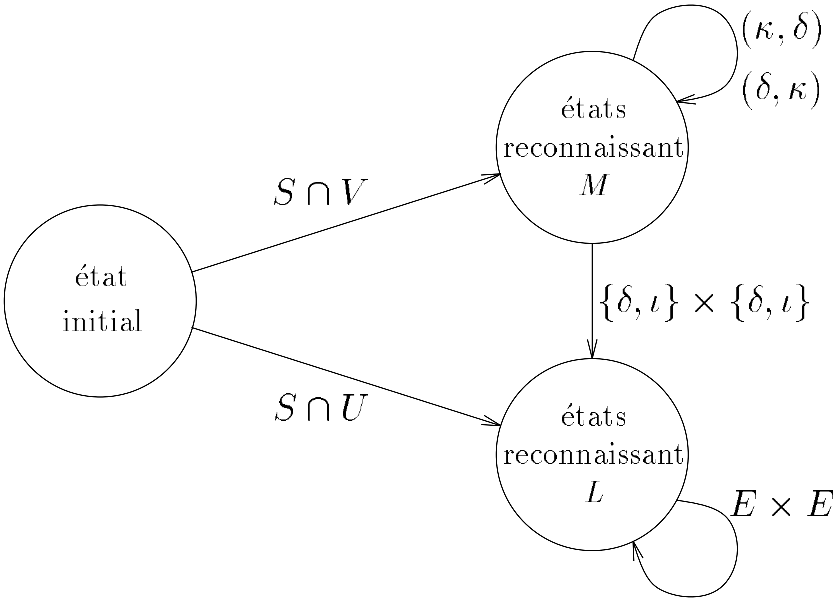

FIG. 5.8 - *Forme générale de l'automate reconnaissant L et M*

5.4 Des matrices pour calculer u_n

Soient (U_n) et (V_n) les deux suites de matrices carrées d'ordre 4 définies comme suit :

$U_n(i, j)$ est le nombre de mots de U de longueur n et de type (i, j),
$V_n(i, j)$ est le nombre de mots de V de longueur n et de type (i, j).

Soient δ, ι, κ les matrices associées aux transformations de base δ, ι, κ (ce sont les matrices de transition de l'automate de la figure 5.7) :

$$\delta = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \iota = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \kappa = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} .$$

Avec ces notations, $(\kappa {U_n}^t \iota)(i, j)$ par exemple est le nombre de mots de type (i, j) dans $\{ \ u.(\kappa, \iota) \ \mid u \in U \text{ et } |u| = n \ \}$.

En prenant en compte les conditions données dans la section 5.3, on peut établir le

Théorème 5.2. *Les matrices U_n et V_n définies ci-dessus satisfont les formules*

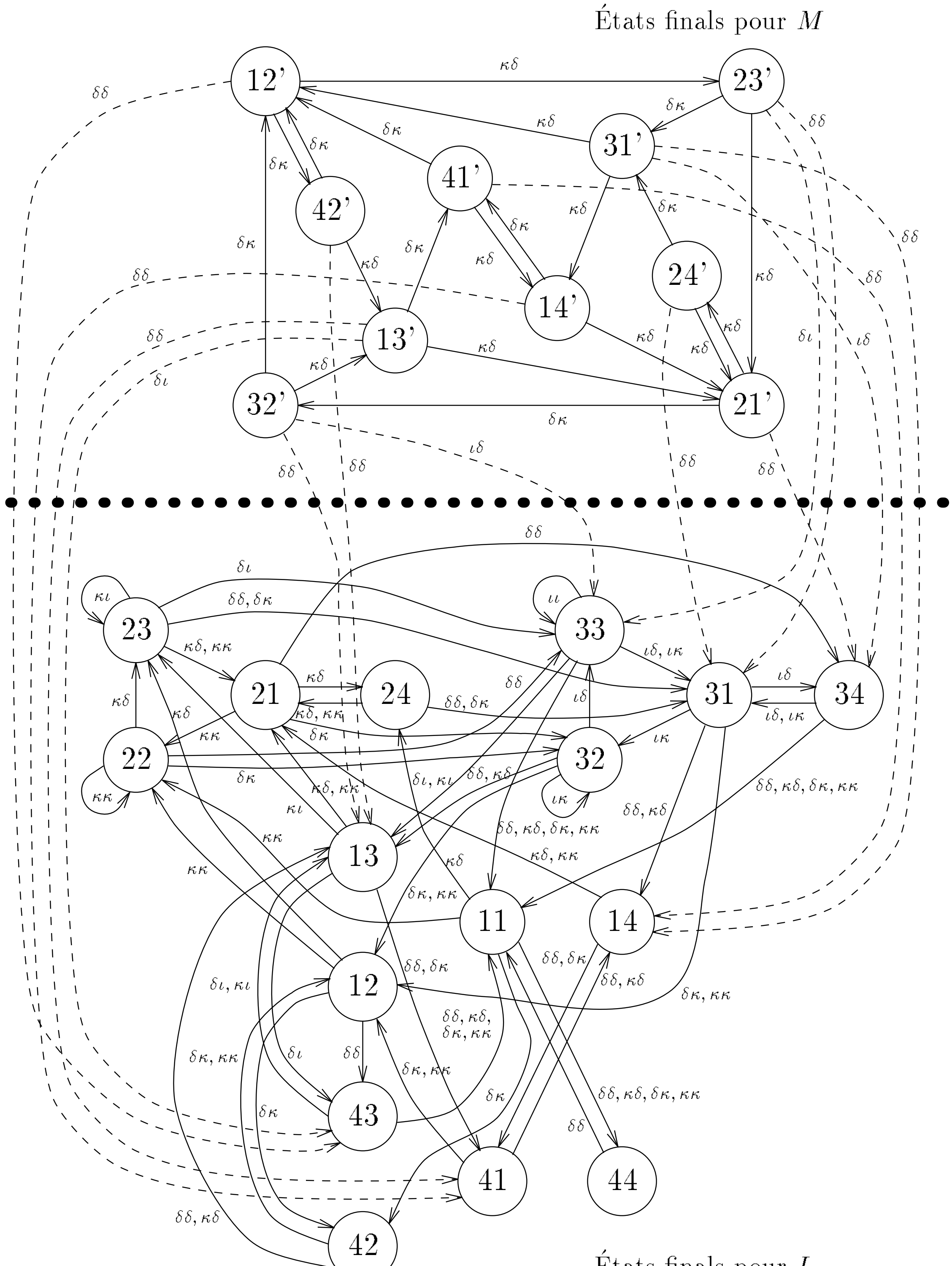

FIG. 5.9 - *Automate reconnaissant L et M (sans l'état initial)*

de récurrence suivantes :

$$\begin{cases} V_{2n} = 0 \\ V_{2n+1} = \kappa V_{n+1}{}^t\delta + \delta V_{n+1}{}^t\kappa \\ U_{2n} = \iota V_n{}^t\iota + \delta V_{n+1}{}^t\delta + (\kappa+\iota)U_n{}^t(\kappa+\iota) + \delta U_{n+1}{}^t\delta \\ U_{2n+1} = \iota V_{n+1}{}^t\delta + \delta V_{n+1}{}^t\iota + (\kappa+\iota)U_{n+1}{}^t\delta + \delta U_{n+1}{}^t(\kappa+\iota) \end{cases}$$

pour tout entier n strictement supérieur à 3. ■

En connaissant les quelques premières valeurs :

$$V_4 = 0, \quad V_5 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad V_6 = 0, \quad V_7 = \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$U_4 = \begin{pmatrix} 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad U_5 = \begin{pmatrix} 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}, \quad U_6 = \begin{pmatrix} 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}, \quad U_7 = \begin{pmatrix} 4 & 2 & 0 & 2 \\ 2 & 0 & 2 & 2 \\ 0 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{pmatrix}$$

il est ainsi possible de calculer U_n et V_n en temps logarithmique, et donc u_n (c'est ainsi que la figure 5.10 a été obtenue) :

Théorème 5.3. *Le nombre u_n de mots binaires sans chevauchement peut être calculé au moyen des formules de récurrence du théorème 5.2 et de la relation*

$$u_n = \sum_{i,j} U_n(i,j)$$

qui vaut pour $n > 3$ (et $u_0 = 1$, $u_1 = 2$, $u_2 = 4$, $u_3 = 6$).

La suite (u_n) est par conséquent une suite 2-régulière *au sens de [2].* ■

Remarque : La récurrence pour V_n peut être résolue :

V_5, V_7 comme ci-dessus, $V_9 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$, $V_{13} = \begin{pmatrix} 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$,

$$V_{2^k+1} = \tfrac{1}{3}.2^{k-3}\begin{pmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} + \tfrac{2}{3}(-1)^k\begin{pmatrix} 0 & 1 & 2 & -1 \\ 1 & 0 & -1 & -1 \\ 2 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{pmatrix},$$

$V_{3.2^{k-1}+1} = 2^{k-3}\begin{pmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ pour $k > 3$; dans tous les autres cas $V_n = 0$.

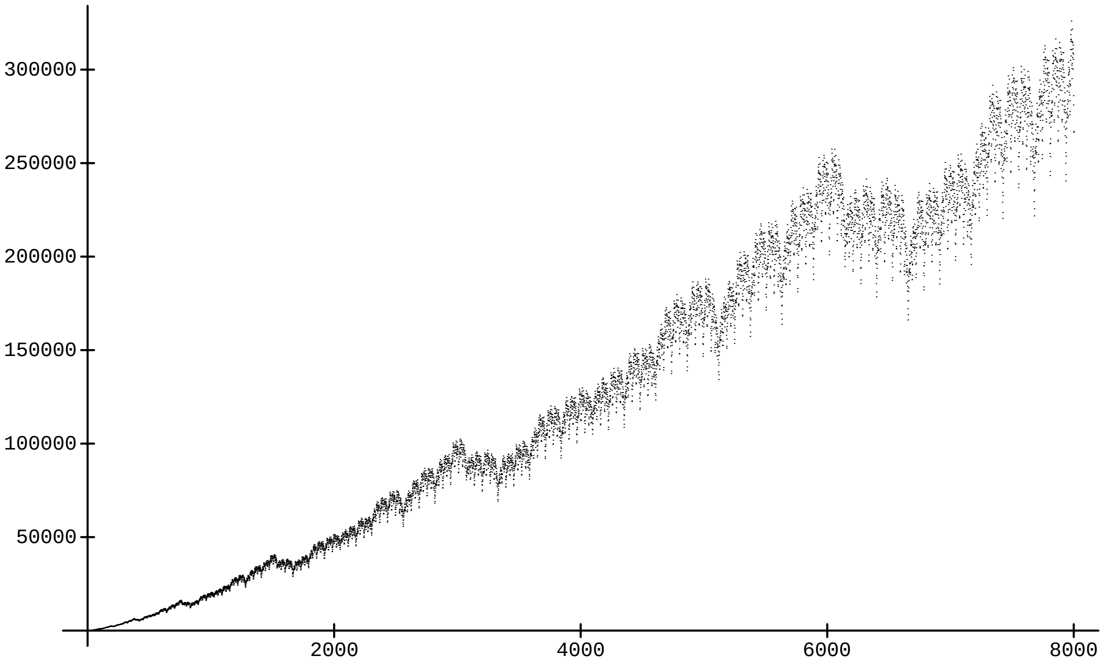

FIG. 5.10 - *La suite u_n pour $n \leqslant 8000$*

En faisant la somme des coefficients de V_n, on obtient le nombre de chevauchements simples, $v_n = \sum_{i,j} V_n(i,j)$ pour $n > 3$: les seules valeurs non nulles sont $v_{2^k+1} = 2^{k-1}$ et $v_{3.2^k+1} = 3.2^k$ pour $k \geqslant 1$. On voit donc que $v_n < n$, ce qui veut dire que v_n est négligeable par rapport à u_n.

5.5 Étude asymptotique de la suite (u_n)

5.5.1 Simplification des relations de récurrence

Il est plus aisé de travailler avec des suites de vecteurs qu'avec des suites de matrices : en tenant compte du fait que U_n et V_n sont des matrices symétriques, et en éliminant les composantes de V_n qui sont toujours nulles, on peut coder les deux matrices par un vecteur de dimension 15.

$$\begin{aligned} X_n &= \Big(U_n(1,1), U_n(1,2), U_n(1,3), U_n(1,4), U_n(2,2), U_n(2,3), U_n(2,4), \\ &\quad U_n(3,3), U_n(3,4), U_n(4,4), V_n(1,2), V_n(1,3), V_n(1,4), V_n(2,3), V_n(2,4)\Big) \ . \end{aligned}$$

Les formules de récurrence deviennent, pour $n > 6$:

$$\begin{cases} X_{2n} = AX_n + BX_{n+1} \\ X_{2n+1} = CX_{n+1} \end{cases}$$

où A, B et C sont des matrices carrées d'ordre 15 (figure 5.11).

On peut aussi écrire $Y_n = \begin{pmatrix} X_{n+1} \\ X_{n+2} \end{pmatrix}$, $F = \begin{pmatrix} C & 0 \\ A & B \end{pmatrix}$ et $G = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix}$. Alors $Y_{2n} = FY_n$ et $Y_{2n+1} = GY_n$ pour $n > 6$.

Remarque : Ces nouvelles formules de récurrence peuvent sembler beaucoup plus simples que celles qui ont été données au théorème 5.2; cependant la complexité du problème est cachée dans les énormes matrices F et G.

5.5.2 Les exposants α et β

Puisque la croissance de la suite (u_n) est polynomiale, il est naturel d'étudier les nombres

$$\alpha = \sup \{ \ r \ \mid \exists C > 0, \forall n, u_n \geqslant Cn^r \ \}$$

et

$$\beta = \inf \{ \ r \ \mid \exists C > 0, \forall n, u_n \leqslant Cn^r \ \} \quad .$$

Ces nombres peuvent aussi s'écrire $\alpha = \liminf \frac{\log u_n}{\log n}$ et $\beta = \limsup \frac{\log u_n}{\log n}$. La figure 5.12 montre le graphe de $\frac{\log u_n}{\log n}$ (à un terme correcteur près qui ne change pas les limites et sert seulement à accélérer la convergence) : on y voit très clairement que ces deux limites sont distinctes, et c'est ce que nous allons maintenant démontrer.

Théorème 5.4. *Les nombres α et β définis ci-dessus sont distincts : $\alpha < \beta$.*

Note : Nous dirons qu'une suite (f_n) *croît comme* une autre suite (g_n) s'il existe deux constantes strictement positives C_1 et C_2 telles que $C_1 g_n \leqslant f_n \leqslant C_2 g_n$ pour n assez grand (i.e. $f_n = \Theta(g_n)$).

Preuve : Considérons d'abord la sous-suite de (Y_n) de terme général $Y_{7.2^k} = F^k Y_7$. La matrice F a une seule valeur propre de module maximal $\rho(F)$ (le rayon spectral de F, qui vaut environ $2,421$), et il est facile de vérifier que Y_7 n'est pas dans la somme des espaces caractéristiques correspondants aux autres valeurs propres. Par conséquent au moins une coordonnée de $Y_{7.2^k}$ croît comme $\rho(F)^k$. Comme $u_{7.2^k+1} + v_{7.2^k+1} + u_{7.2^k+2} + v_{7.2^k+2}$ est une combinaison linéaire à coefficients strictement positifs de ces coordonnées, cette expression croît elle aussi comme $\rho(F)^k$, donc soit $u_{7.2^k+1}$ soit $u_{7.2^k+2}$ croît comme $\rho(F)^k$ (on a vu que $v_n = O\,(n)$, donc $v_{7.2^k+1} + v_{7.2^k+2}$ est négligeable). Nous avons ainsi trouvé une sous-suite de (u_n) qui croît comme n^r avec $r = \log_2 \rho(F) \simeq 1,275$, ce qui entraîne $\alpha < 1,276$.

Considérons maintenant la sous-suite de terme général $Y_{\frac{22.4^k-1}{3}} = (GF)^k Y_7$. Par un argument similaire, on montre que cette sous-suite croît comme $\rho(GF)^k$

$$A = \begin{pmatrix}
0&0&0&0&0&0&0&1&2&1&0&0&0&0&0\\
0&0&1&1&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&2&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}$$

$$B = \begin{pmatrix}
0&0&0&0&0&0&0&1&2&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&0&1&1\\
0&0&1&1&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}$$

$$C = \begin{pmatrix}
0&0&0&0&0&0&0&2&4&2&0&0&0&0&0\\
0&0&1&1&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&1&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&2&0&0&0&0&0&0&0&2&0\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0
\end{pmatrix}$$

FIG. 5.11 - *Les matrices A, B et C*

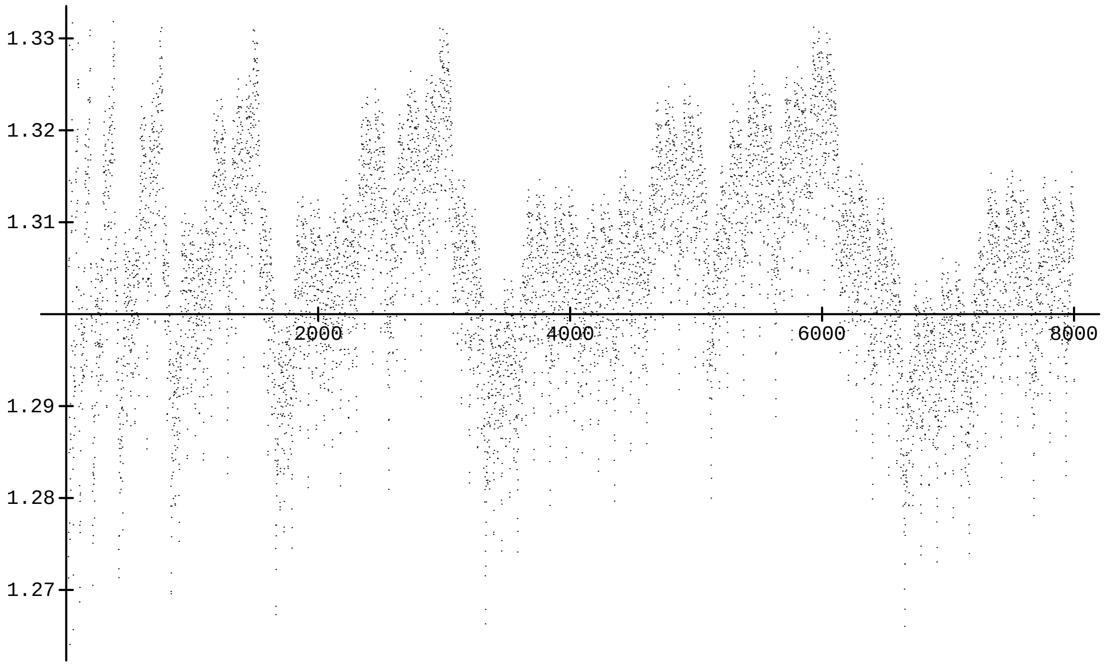

FIG. 5.12 - *Graphe de* $\frac{\log(u_n/2{,}4)}{\log n}$ *pour* $n \leqslant 8000$

avec $\rho(GF) \simeq 6,340$. Il y a donc une sous-suite de u_n qui croît comme n^r avec $r = \log_4 \rho(GF) \simeq 1,3322$, donc $\beta > 1,332$. ∎

Remarque : Ces résultats, avec ceux de Kobayashi, fournissent les encadrements suivants :

$$1,155 < \alpha < 1,276 < 1,332 < \beta < 1,587 \ .$$

5.5.3 Sommes partielles

Soit $s_n = \sum_{k=0}^{n-1} u_k$ le nombre de mots sans chevauchement de longueur inférieure à n. La suite (s_n) est parfois appelée *fonction de croissance* des mots sans chevauchement. Nous pouvons décrire précisément le comportement asymptotique de cette suite :

Théorème 5.5. *La suite s_n croît comme n^r, avec $r = \log_2 \zeta \simeq 2,310$, où*

$$\zeta = \rho(F+G) = \frac{3}{2} + \sqrt{3} + \sqrt{\frac{5}{4} + \sqrt{3}}$$

est la plus grande racine du polynôme $X^4 - 6X^3 + 5X^2 + 4$.

Preuve : Soit $S_n = \sum_{k=7}^{n-1} Y_k$, pour $n > 6$. Alors

$$S_{2n} = \sum_{k=7}^{13} Y_k + \sum_{k=7}^{n-1} Y_{2k} + \sum_{k=7}^{n-1} Y_{2k+1} = S_{14} + (F+G)S_n \ .$$

On vérifie aisément que la matrice $F + G - 1$ est inversible, posons donc $T = (F+G-1)^{-1}S_{14}$: alors $S_{2n} + T = (F+G)(S_n+T)$, et $S_{7.2^k} = (F+G)^k T - T$, donc au moins une coordonnée de $S_{7.2^k}$ croît comme ζ^k. Comme plus haut, on en déduit que $s_{7.2^k}$ croît aussi comme ζ^k.

Si $7.2^k \leqslant n \leqslant 7.2^{k+1}$, alors $s_{7.2^k} \leqslant s_n \leqslant s_{7.2^{k+1}}$, donc s_n croît comme $\zeta^{\log_2 n}$, i.e. $n^{\log_2 \zeta}$. ■

Remarque : Il est sans doute possible d'affiner ce résultat, en montrant par exemple, au moyen de techniques similaires à celles de [24], que $s_n = f(\log_2 n)n^r + o(n^r)$, où f est une fonction périodique de période 1.

5.6 Avec d'autres motifs

Il serait intéressant de pouvoir obtenir des résultats du même genre pour d'autres motifs ou formules. Il semble qu'il y ait une liaison assez forte entre, d'une part la croissance polynomiale, et d'autre part l'existence d'un lemme similaire au lemme 5.1, qui permette de coder les mots évitant le motif considéré au moyen d'un morphisme (ou du moins d'un nombre fini de morphismes). Un exemple semble prometteur : la formule $AB.BC.CA.BA.AC$ sur un alphabet à quatre lettres, pour laquelle Baker, McNulty et Taylor ont montré que la croissance était polynomiale [5, Theorem 1.5] et que les mots évitant cette formule avaient un graphe de transition particulier [5, Lemma 8.7]. Nous n'avons pas poussé plus loin l'analogie, mais c'est sans doute possible.

Chapitre 6

Un algorithme pour décider si un HDOL évite un motif

Pour montrer qu'un motif p est évitable sur un alphabet donné Σ, on doit fabriquer un langage infini L qui évite p. Le plus souvent, L est un DOL-langage ou un HDOL-langage. Dans ce chapitre, nous présentons un algorithme qui permet, sous certaines conditions, étant donné un HDOL-système $G = (\Sigma, h, w_0, \Sigma', g)$, de tester si le langage $L(G)$ évite p.

Nous définissons d'abord les *morphismes circulaires* et les *DOL-systèmes* et *HDOL-systèmes circulaires*, puis l'*image inverse* d'un motif par un morphisme circulaire. Nous montrons ensuite qu'en calculant les images inverses successives de p, nous pouvons décider si $L(G)$ évite p pour n'importe quel motif p fixé (l'algorithme est valable également pour les motifs avec constantes et les formules), à condition que le HDOL-système G soit circulaire et expansif.

Dans ce chapitre, le terme *motif* désigne les motifs avec constantes, sauf si nous précisons *motif sans constante*.

6.1 Morphismes circulaires

6.1.1 Morphismes circulaires sur Σ^*

Soit $h\colon \Sigma^* \to \Sigma'^*$ un morphisme injectif. On dit que h est *circulaire* si $C = h(\Sigma)$ est un code circulaire, c'est-à-dire

$$\begin{aligned}&\forall u \in \Sigma^*, \forall v \in \Sigma^*, \forall x \in \Sigma, \forall t \in \Sigma'^*, \forall s \in \Sigma'^+,\\ &\quad [(h(x) = ts \text{ et } h(u) = sh(v)t) \Longrightarrow (t = \varepsilon \text{ et } u = xv)]\,.\end{aligned}$$

Soit $w \in \Sigma'^*$. On dit que (w_1, w_2) est un *point de synchronisation* de w (pour h) si $w = w_1 w_2$ et

$$\forall v_1 \in \Sigma'^*, \forall v_2 \in \Sigma'^*, [v_1 w v_2 \in C^* \Longrightarrow (v_1 w_1 \in C^* \text{ et } w_2 v_2 \in C^*)]$$

et nous notons cela $w = w_1|w_2$. Dans cette notation, le symbole $|$ n'a rien à voir avec la relation de divisibilité. Il doit être compris comme une *barre de mesure* qui indique un endroit dans w qui tombe nécessairement à la limite entre deux mots du code C chaque fois que w est facteur d'un élément de C^*. On pourra aussi mettre plusieurs barres de mesure dans le même mot : $w = w_1|w_2|w_3$ signifie que (w_1, w_2w_3) et (w_1w_2, w_3) sont deux points de synchronisation de w, ce qui implique, si w est facteur d'au moins un élément de C^*, que w_2 est dans C^*. Remarquons enfin que si w n'est facteur d'aucun mot de C^*, tout couple (w_1, w_2) tel que $w = w_1w_2$ est trivialement un point de synchronisation de w et qu'on peut donc mettre des barres de mesure entre toutes les lettres de w.

Nous utiliserons la caractérisation suivante des morphismes circulaires [9, Theorem 7.2.6] :

Proposition 6.1. *Un morphisme injectif $h\colon \Sigma^* \to \Sigma'^*$ est circulaire si et seulement si tout mot assez long de Σ'^* a au moins un point de synchronisation.* ■

Si tout mot $w \in \Sigma'^*$ de longueur supérieure ou égale à L a au moins un point de synchronisation pour h, on dit que le morphisme h est un morphisme circulaire de *délai de synchronisation L*.

Remarque : Le nom *délai de synchronisation* est souvent donné à un autre nombre : le plus petit entier s tel que tout mot de C^s a un point de synchronisation. Il est clair que s est fini si et seulement si L l'est; donc, quelle que soit la définition adoptée, un morphisme injectif est *à délai de synchronisation borné* si et seulement s'il est circulaire.

Soit S_h l'ensemble des mots non vides qui ont au plus un point de synchronisation (*petits mots*); si h est circulaire de délai L, cet ensemble est clairement fini et la longueur de ses éléments est au plus $2L$. À tout mot $w \in \Sigma'^+ \setminus S_h$ qui est facteur d'au moins un mot de C^* on peut alors associer trois mots $c_w \in \Sigma^+$, $d_w \in \Sigma'^*$, $e_w \in \Sigma'^*$ tels que $|d_w| \leqslant L$, $|e_w| \leqslant L$ et $w = d_w|h(c_w)|e_w$.

Exemples:

Si h est injectif et C est un code *comma-free*, alors h est un morphisme circulaire et son délai de synchronisation est $L \leqslant 2 \max_{u \in C} |u| - 2$.

Le morphisme de Thue-Morse :

$$\begin{array}{rccc} \theta\colon & \{a,b\}^* & \longrightarrow & \{a,b\}^* \\ & a & \longmapsto & ab \\ & b & \longmapsto & ba \end{array}$$

n'est pas circulaire, car $(ab)^n$ n'a pas de point de synchronisation quel que soit n.

Le morphisme

$$\begin{array}{rccc} h\colon & \{a,b\}^* & \longrightarrow & \{a,b\}^* \\ & a & \longmapsto & aab \\ & b & \longmapsto & ba \end{array}$$

est circulaire de délai de synchronisation 5 : les seuls mots sans point de synchronisation sont ε, a, b, aa, ab, ba, aba, baa, $abaa$ (à cause des points de synchronisation $a|aa$, $|aab|$, $b|b$, $|ba|b$). S_h contient en outre bb, aaa et abb.

6.1.2 Morphismes circulaires sur U

Le morphisme de Thue-Morse n'est pas circulaire, mais cela n'est dû qu'à certains mots particuliers, à savoir les puissances de a ou de b. On aimerait pouvoir dire que θ est circulaire sauf pour ces mots-là : nous allons maintenant tenter de donner un sens précis à cette phrase.

Soit $h\colon \Sigma^* \to \Sigma'^*$ un morphisme, que l'on ne supposera pas nécessairement injectif, et $U \in \mathscr{P}(\Sigma^*)$ un langage factoriel (la définition n'a d'intérêt que si U est infini).

Soit $w \in \Sigma'^*$. On dit que (w_1, w_2) est un point de synchronisation de w (pour h) *sur U* si $w = w_1 w_2$ et

$$\begin{aligned} &\forall v_1 \in \Sigma'^*, \forall v_2 \in \Sigma'^*, \forall u \in U, \exists u_1 \in U, \exists u_2 \in U, \\ &\quad [v_1 w v_2 = h(u) \Longrightarrow (u = u_1 u_2 \text{ et } v_1 w_1 = h(u_1) \text{ et } w_2 v_2 = h(u_2))] \end{aligned}$$

et nous notons cela $w = w_1|_U w_2$, ou simplement $w = w_1|w_2$ s'il n'y a pas d'ambiguïté quant au choix de U. Si h est *injectif sur U*, c'est-à-dire que deux éléments distincts de U ont des images distinctes par h, cette définition est équivalente à

$$\forall v_1 \in \Sigma'^*, \forall v_2 \in \Sigma'^*, [v_1 w v_2 \in h(U) \Longrightarrow (v_1 w_1 \in h(U) \text{ et } w_2 v_2 \in h(U))]$$

et on retrouve en particulier la définition de la section précédente quand $U = \Sigma^*$.

On dit que le morphisme h est *circulaire sur U* si h est injectif sur U et tout mot assez long de Σ'^* a au moins un point de synchronisation sur U. Un morphisme est donc circulaire si et seulement s'il est circulaire sur Σ^*. On peut également définir le *délai de synchronisation sur U* d'un morphisme.

On note $S_h(U)$ l'ensemble des mots non vides qui ont au plus un point de synchronisations sur U. Si h est circulaire sur U de délai L, alors $S_h(U)$ est fini et tout mot $w \in F(h(U)) \setminus (S_h(U) \cup \{\varepsilon\})$ se décompose sous la forme $w = d_w|_U h(c_w)|_U e_w$ avec $c_w \in U$, $d_w \in \Sigma'^*$ et $e_w \in \Sigma'^*$ tels que $c_w \neq \varepsilon$, $|d_w| \leqslant L$ et $|e_w| \leqslant L$.

Ainsi, si on prend pour U l'ensemble des mots de $\{a,b\}^*$ qui ne contiennent ni a^3 ni b^3, on peut dire que le morphisme de Thue-Morse est circulaire sur U

de délai 4 : il y a en effet des points de synchronisation supplémentaires $ab|_U ab$ et $ba|_U ba$, ce qui donne $S_h(U) = \{a, b, aa, ab, ba, bb, aba, bab\}$.

Pour finir, remarquons que

Proposition 6.2. *Si U et U' sont deux sous-langages factoriels de Σ^* tels que $U' \subset U$, et $h\colon \Sigma^* \to \Sigma'^*$ est un morphisme, alors tout point de synchronisation par h sur U est aussi un point de synchronisation sur U'. Si h est circulaire sur U de délai L, alors h est aussi circulaire sur U', de délai inférieur ou égal à L.* ■

6.1.3 L-systèmes circulaires

Dans le cas des DOL-systèmes, un langage U s'impose : l'ensemble des facteurs du langage engendré par le DOL-système lui-même. Aussi, on dira qu'un DOL-système $G = (\Sigma, h, w_0)$ est *circulaire* si le morphisme h est circulaire sur $F(L(G))$. On retrouve ainsi les notions de *DOL-langage circulaire* de Mignosi et Séébold [40] et de *substitution reconnaissable* de Mossé [42], du moins quand le morphisme est injectif.

Pour un HDOL-système, c'est l'ensemble des facteurs du langage engendré par le DOL-système sous-jacent qui va jouer le rôle de U : on dit que le HDOL-système $G = (\Sigma, h, w_0, \Sigma', g)$ est *circulaire* si les morphismes h et g sont tous les deux circulaires sur $F(L(G^0))$.

Exemples:

Soit $\Sigma = \{a, b\}$ et θ le morphisme de Thue-Morse. On sait que le langage engendré par le DOL-système $G = (\Sigma, \theta, a)$ est sans cube, et donc qu'il est contenu dans $U = \Sigma^* \setminus \Sigma^*\{a^3, b^3\}\Sigma^*$. On a vu plus haut que θ est circulaire sur U, et donc d'après la proposition 6.2 il est aussi circulaire sur $F(L(G))$: le DOL-système G est donc circulaire.

Soit $G = (\Sigma_3, \mu, a, \Sigma_2, \psi)$ le HDOL-système utilisé par Roth pour montrer que $ABAAB$ est 2-évitable (voir page 34). Le premier morphisme, μ, est circulaire; le second,

$$\begin{array}{rrcl} \psi\colon & \{a, b, c\} & \longrightarrow & \{a, b\} \\ & a & \longmapsto & aaa \\ & b & \longmapsto & bbb \\ & c & \longmapsto & ababab \end{array}$$

n'est pas circulaire, mais comme $U = F(L(G^0))$ est un langage sans carré on peut montrer que ψ est circulaire sur U, de délai 7. On en déduit donc que G est circulaire.

Nous pouvons maintenant énoncer le résultat principal de ce chapitre, qui sera démontré à la section 6.3 :

Théorème 6.1. *Il existe un algorithme qui, étant donnés un HDOL-système* *circulaire et expansif G et un motif avec constantes p, décide si le langage engendré par G évite le motif p.*

Dans le cas des motifs sans constantes, ce résultat peut être renforcé en utilisant un théorème de Mignosi et Séébold [40] : tout DOL-système non répétitif est circulaire.

Théorème 6.2. *Il existe un algorithme qui, étant donnés un HDOL-système $G = (\Sigma, h, a, \Sigma', g)$ tel que a est une lettre de Σ et g est circulaire sur $F(L(G^0))$, et un motif sans constantes p, décide si le langage $L(G)$ évite le motif p.*

Preuve : Si h n'est pas circulaire sur $U = F(L(G^0))$, le langage $L(G^0)$ est répétitif d'après [40, Theorem 1, Corollary], donc $L(G)$ est également répétitif puisque g est non-effaçant (sinon g ne pourrait pas être circulaire sur U), donc $L(G)$ contient le motif p d'après la proposition 1.5. On se ramène donc au cas où G est circulaire; il reste à montrer qu'on peut le supposer expansif. Pour cela, on utilise la construction de la proposition 3.19; il faut encore montrer qu'après cette opération le morphisme de codage reste circulaire (sur un nouveau langage factoriel $\widetilde{U}$) : c'est l'objet de la section 6.4, dans laquelle nous devons supposer que l'axiome est une lettre. ■

Pour les DOL-systèmes, on a donc :

Corollaire 6.1. *Il existe un algorithme qui, étant donnés un DOL-système G dont l'axiome est une lettre et un motif sans constantes p, décide si le langage $L(G)$ évite le motif p.*

La seule restriction qui subsiste est que l'axiome doit être une lettre, ce qui n'est pas bien gênant au vu de la proposition 3.5. Notons que Carpi a obtenu indépendamment un résultat similaire [12, Proposition 7.4], valable uniquement pour les DOL-systèmes expansifs, par une méthode assez différente reposant sur les *systèmes de représentation*, une généralisation des systèmes de numération qui permet aussi, entre autres, de formaliser des constructions comme le codage des mots sans chevauchement que nous étudions au chapitre 5.

6.2 L'image inverse d'un motif par un morphisme circulaire

Notation: Soit $p \in (\Delta \cup \Sigma')^*$ un motif avec constantes sur un alphabet Σ'. On note $[p]_{\Sigma'}$ l'ensemble des mots de Σ'^* qui contiennent une occurrence de p. Si aucune ambiguïté n'est possible, on note cet ensemble simplement $[p]$.

Avec cette notation, un langage $L \subset \Sigma'^*$ évite le motif p si et seulement si $F(L) \cap [p] = \emptyset$, et p est évitable sur Σ' si et seulement si $\Sigma'^* \setminus \Sigma'^*[p]\Sigma'^*$ est infini.

On appelle *valuation* sur un alphabet Σ un morphisme de monoïde de Σ^* dans le groupe additif $(\mathbb{R}, +)$, et on note habituellement $\|u\|$ l'image d'un mot u par une valuation. Les valuations usuelles sont la longueur $|u|$ et le nombre d'occurrences $|u|_a$ d'une lettre a, et toute valuation est combinaison linéaire de celles-ci. Par exemple, si Σ' est une partie de Σ, l'application qui à u associe $|u|_{\Sigma'} = \sum_{a \in \Sigma'} |u|_a$ est une valuation. Une valuation à valeurs dans $\mathbb{R}^+$ est dite *positive*.

Proposition 6.3. *Soient Σ et Σ' deux alphabets munis de valuations positives, $U \in \mathscr{P}(\Sigma^*)$ un langage factoriel, $h\colon \Sigma^* \to \Sigma'^*$ un morphisme circulaire sur U, m et M deux réels tels que $0 < m\,\|x\|_\Sigma \leqslant \|h(x)\|_{\Sigma'} \leqslant M$ pour tout $x \in \Sigma$, L un réel tel que tout mot de Σ'^* de valuation au moins L ait un point de synchronisation sur U et tel que les éléments de $S_h(U)$ soient de longueur au plus $2L$, et $p \in (\Delta \cup \Sigma')^*$ un motif avec constantes sur Σ'. Alors il existe un ensemble fini $Q \subset (\Delta \cup \Sigma)^*$ de motifs avec constantes sur Σ tel que :*

(i) $h^{-1}([p]_{\Sigma'}) \cap U = \bigcup\limits_{q \in Q} [q]_\Sigma \cap U$

(ii) $\forall q \in Q, |q|_\Delta \leqslant |p|_\Delta$ et $\|q\|_\Sigma \leqslant \frac{2M}{m} + \frac{1}{m}\|p\|_{\Sigma'} + \frac{2L}{m}|p|_\Delta$

où les valuations de Σ et Σ' sont étendues respectivement à $\Delta \cup \Sigma$ et $\Delta \cup \Sigma'$ par $\|X\|_\Sigma = \|X\|_{\Sigma'} = 0$ pour $X \in \Delta$.

En général on prendra $\|u\| = |u|$ et L sera le délai de synchronisation de h sur U. Si de plus $U = \Sigma^*$, on obtient la version simplifiée suivante :

Proposition 6.4. *Soient $h\colon \Sigma^* \to \Sigma'^*$ un morphisme circulaire de délai de synchronisation L, m et M deux entiers tels que $1 \leqslant m \leqslant |h(x)| \leqslant M$ pour tout $x \in \Sigma$, et $p \in (\Delta \cup \Sigma')^*$ un motif avec constantes sur Σ'. Alors il existe un ensemble fini $Q \subset (\Delta \cup \Sigma)^*$ de motifs avec constantes sur Σ tel que :*

(i) $h^{-1}([p]_{\Sigma'}) = \bigcup\limits_{q \in Q} [q]_\Sigma$

(ii) $\forall q \in Q, |q|_\Delta \leqslant |p|_\Delta$ et $|q|_\Sigma \leqslant \frac{2M}{m} + \frac{1}{m}|p|_{\Sigma'} + \frac{2L}{m}|p|_\Delta$

■

Preuve : Soit $w \in h^{-1}([p]_{\Sigma'}) \cap U$. Supposons que w est minimal, c'est-à-dire qu'aucun facteur propre de w n'est dans $h^{-1}([p]_{\Sigma'})$. Si $p = u_0X_1u_1X_2\ldots X_ru_r$ avec $X_i \in \Delta$ et $u_i \in \Sigma'^*$, alors

$$h(w) = v_1u_0\varphi(X_1)u_1\varphi(X_2)u_2\ldots\varphi(X_r)u_rv_2$$

avec $v_i \in \Sigma'^*$ et $\varphi\colon \Delta \to \Sigma'^+$. Comme w est minimal, $\|v_i\|_{\Sigma'} < M$.

Soit $\Delta' = \{ X \in \Delta \mid \varphi(X) \notin S_h(U) \}$. Soient i_1,...,i_s les indices i tels que $X_i \in \Delta'$, $i_0 = 0$, $i_{s+1} = r+1$, $Y_j = X_{i_j}$ et

$$u'_j = u_{i_j}\varphi\left(X_{i_j+1}\right)u_{i_j+1}\varphi\left(X_{i_j+2}\right)u_{i_j+2}\ldots\varphi\left(X_{i_{j+1}-1}\right)u_{i_{j+1}-1}$$

pour $0 \leqslant j \leqslant s$. Alors $h(w)$ s'écrit aussi

$$h(w) = v_1u'_0\varphi\left(Y_1\right)u'_1\varphi\left(Y_2\right)u'_2\ldots\varphi\left(Y_s\right)u'_sv_2 \ .$$

Comme $\varphi(Y_j) \in F(h(U)) \setminus (S_h(U) \cup \{\varepsilon\})$, on a

$$\varphi\left(Y_j\right) = d_{\varphi(Y_j)}h\left(c_{\varphi(Y_j)}\right)e_{\varphi(Y_j)} = d_jh\left(\psi\left(Y_j\right)\right)e_j$$

où $\psi\colon \Delta' \to \Sigma^+$ est le morphisme défini par $\psi(Y) = c_{\varphi(Y)}$. À cause des points de synchronisation $d_j|_Uh\left(\psi\left(Y_j\right)\right)|_Ue_j$, les $s+1$ mots $v_1u'_0d_1$, $e_ju'_jd_{j+1}$ (pour $1 \leqslant j \leqslant s-1$) et $e_su'_sv_2$ sont dans $h(U)$. Nous pouvons donc appliquer h^{-1}, qui est bien défini sur $h(U)$ puisque h est injectif sur U :

$$w = h^{-1}\left(v_1u'_0d_1\right)\psi(Y_1)h^{-1}\left(e_1u'_1d_2\right)\ldots\psi(Y_s)h^{-1}\left(e_su'_sv_2\right)$$

donc $w \in [q]_\Sigma$ avec

$$q = h^{-1}\left(v_1u'_0d_1\right)Y_1h^{-1}\left(e_1u'_1d_2\right)\ldots Y_sh^{-1}\left(e_su'_sv_2\right) \ .$$

Si $w' \in h^{-1}\left([p]_{\Sigma'}\right)$ n'est pas minimal et contient w, alors on a aussi $w' \in [q]_\Sigma$.

Réciproquement, si $w' \in [q]_\Sigma$, ce mot s'écrit

$$w' = v'_1h^{-1}\left(v_1u'_0d_1\right)\psi'(Y_1)h^{-1}\left(e_1u'_1d_2\right)\ldots\psi'(Y_s)h^{-1}\left(e_su'_sv_2\right)v'_2$$

avec $\psi'\colon \Delta' \to \Sigma^+$, et alors $h(w') \in [p]_{\Sigma'}$ puisque $h(w')$ contient $\varphi'(p)$ où $\varphi'\colon \Delta \to \Sigma'^+$ est défini par

$$\left\{\begin{array}{lcll} \varphi'(X) & = & \varphi(X) & \text{si } X \notin \Delta', \\ \varphi'(X) & = & d_{\varphi(X)}h\left(\psi'(X)\right)e_{\varphi(X)} & \text{si } X \in \Delta'. \end{array}\right.$$

Notons Q l'ensemble de tous les motifs q obtenus par cette méthode pour tous les choix possibles de w (cet ensemble peut *a priori* être infini) : *(i)* est clairement vérifié.

D'après la construction de q,

$$\begin{array}{rcl} |q|_\Delta & = & s \leqslant r = |p|_\Delta \\ \|q\|_\Sigma & = & \left\|h^{-1}\left(v_1u'_0d_1e_1u'_1d_2e_2u'_2\ldots d_se_su'_sv_2\right)\right\|_\Sigma \\ & \leqslant & \dfrac{1}{m}\left\|v_1u'_0d_1e_1u'_1d_2e_2u'_2\ldots d_se_su'_sv_2\right\|_{\Sigma'} \\ & \leqslant & \dfrac{1}{m}\left(\|v_1v_2\|_{\Sigma'} + \|d_1e_1d_2e_2\ldots d_se_s\|_{\Sigma'} + \|u_0u_1\ldots u_r\|_{\Sigma'} + \displaystyle\sum_{\substack{1\leqslant i\leqslant r\\ X_i\notin\Delta'}} \|\varphi\left(X_i\right)\|_{\Sigma'}\right) \\ & \leqslant & \dfrac{1}{m}\left(2M + 2sL + \|p\|_{\Sigma'} + (r-s)2L\right) \end{array}$$

ce qui démontre *(ii)*.

Comme le nombre de motifs satisfaisant *(ii)* est fini, l'ensemble Q est nécessairement fini. ■

L'ensemble Q est noté $Q = h_U^{-1}(p)$, ou simplement $Q = h^{-1}(p)$, et est appelé *image inverse* du motif p par le morphisme h dans U. Cet ensemble n'est pas défini de manière unique, mais ce qui est important est que nous avons une méthode effective pour construire un tel ensemble. En particulier, si un élément de Q est contenu dans un autre élément de Q, il est possible de supprimer ce dernier sans changer les propriétés *(i)* et *(ii)*. En pratique, nous chercherons à minimiser la taille de l'ensemble Q de manière à réduire le temps de calcul.

Si P est un ensemble fini de motifs avec constantes sur Σ', nous notons naturellement $h_U^{-1}(P)$ l'ensemble $h_U^{-1}(P) = \bigcup_{p \in P} h_U^{-1}(p)$, qui est aussi un ensemble fini. Nous notons aussi $[P]_{\Sigma'}$ l'ensemble $\bigcup_{p \in P} [p]_{\Sigma'}$.

6.3 L'algorithme

Nous pouvons maintenant formuler notre algorithme. Rappelons que les données sont les suivantes :

- $G = (\Sigma, h, w_0, \Sigma', g)$, un HDOL-système circulaire et expansif;
- $p \in (\Delta \cup \Sigma')^*$, un motif avec constantes.

et que l'algorithme doit indiquer si p est évité par le HDOL-langage engendré par G. Dans cette section, il sera sous-entendu que toutes les images inverses de motifs par des morphismes sont des images inverses dans $U = F(L(G^0))$.

Pour tout entier $j \in \mathbb{N}$, nous appelons Q_j l'ensemble fini $Q_j = h^{-j}(g^{-1}(p))$, qui se calcule en partant de $Q_0 = g^{-1}(p)$ par la formule de récurrence $Q_{j+1} = h^{-1}(Q_j)$ pour tout $j \in \mathbb{N}$. Soit $Q = \bigcup_{j=0}^{\infty} Q_j$, de telle sorte que

$$[Q]_\Sigma \cap U = \bigcup_{j=0}^{\infty} h^{-j}\left(g^{-1}([p]_{\Sigma'})\right) \cap U \ .$$

En utilisant le fait que le morphisme h est expansif, nous pouvons démontrer que l'ensemble Q est toujours fini. Supposons d'abord que h est partout croissant : soient m et M des entiers tels que pour toute lettre $x \in \Sigma$, $2 \leqslant m \leqslant |h(x)| \leqslant M$, et soit

$$\alpha = \max_{q \in Q_0} |q|_\Sigma$$

et

$$\beta = \max\left(\alpha, \frac{2M + 2L\,|p|_\Delta}{m-1}\right) \quad .$$

Prouvons par récurrence sur j que

$$\forall j \in \mathbb{N}, \forall q \in Q_j, |q|_\Delta \leqslant |p|_\Delta \text{ et } |q|_\Sigma \leqslant \beta \quad .$$

Cette inégalité est évidente quand $j = 0$, et si elle est vraie pour j, alors d'après l'inégalité *(ii)* de la proposition 6.3, en prenant la longueur comme valuation sur Σ, on a pour tout $q \in Q_{j+1} = h^{-1}(Q_j)$:

$$|q|_\Delta \leqslant |p|_\Delta \text{ et } |q|_\Sigma \leqslant \frac{2M}{m} + \frac{1}{m}\beta + \frac{2L}{m}|p|_\Delta \leqslant \beta \quad .$$

Par conséquent,

$$\forall q \in Q, |q|_\Delta \leqslant |p|_\Delta \text{ et } |q|_\Sigma \leqslant \beta \quad ,$$

ce qui implique que Q est fini.

Si h est expansif mais non partout croissant, il existe une valuation sur Σ telle que $\|h(x)\| > \|x\|$ pour tout $x \in \Sigma$. Supposons en effet que h^n est partout croissant. Pour toute lettre x, il existe un entier n_x, compris entre 0 et $n-1$, tel que $|h^{n_x}(x)| = 1$ et $|h^{n_x+1}(x)| \geqslant 2$. Posons $\|x\| = 1 - \frac{n_x}{2n}$. Si $n(x) > 0$, on a

$$\|h(x)\| = 1 - \frac{n_{h(x)}}{2n} = 1 - \frac{n_x - 1}{2n} = \|x\| + \frac{1}{2n}$$

et si $n(x) = 0$, le mot $h(x)$ contient au moins deux lettres y et z et

$$\|h(x)\| \geqslant \|y\| + \|z\| = 2 - \frac{n_y + n_z}{2n} \geqslant 2 - \frac{2(n-1)}{2n} = 1 + \frac{1}{n} \quad .$$

Dans les deux cas, on a $\|h(x)\| \geqslant m\,\|x\|$ avec $m = 1 + \frac{1}{2n}$. On peut ensuite procéder comme ci-dessus en remplaçant la longueur par la valuation que nous venons de construire. Comme $m > 1$, on conclut également que Q est fini.

Pour calculer l'ensemble Q, on part de Q_0 et on calcule successivement Q_1, Q_2, etc. au moyen de la relation de récurrence $Q_{j+1} = h^{-1}(Q_j)$, jusqu'à ce que $Q_{j+1} \subset \bigcup_{i=0}^{j} Q_i$, et alors $Q = \bigcup_{i=0}^{j} Q_i$.

Si un motif $q \in Q$ est contenu dans w_0, cela veut dire qu'il existe un indice j tel que $w_0 \in h^{-j}(g^{-1}([p]_{\Sigma'}))$. Alors $g(h^j(w_0))$ contient p, et nous pouvons conclure que $L(G)$ n'évite pas p. Réciproquement, si w_0 évite tous les éléments de Q, cela veut dire qu'aucun des mots $g(h^j(w_0))$ ne contient p, c'est-à-dire que $L(G)$ évite p. Comme Q est fini, on peut donc effectivement tester si $L(G)$ évite p.

En pratique, on peut procéder comme indiqué à la figure 6.1. La variable R est une pile dans laquelle on stocke les motifs qui restent à traiter. À la fin

Algorithme *test-HDOL-circulaire*
Entrée : un HDOL-système circulaire expansif $G = (\Sigma, h, w_0, \Sigma', g)$
un motif avec constantes $p \in (\Delta \cup \Sigma')^*$
Sortie : un booléen, qui indique si le HDOL-langage $L(G)$ évite p
Variables : deux ensembles de motifs avec constantes Q et R

Début
$Q \leftarrow \emptyset$
$R \leftarrow g^{-1}(p)$
Tant que $R \neq \emptyset$:
Choisir un $q \in R$ et l'enlever de R
Si q n'est divisible par aucun élément de Q :
Si q divise w_0, rendre *faux*
Sinon, ajouter q à Q et $h^{-1}(q)$ à R
Rendre *vrai*
Fin.

FIG. 6.1 - *Algorithme pour tester si un HDOL-langage évite un motif*

de la boucle, c'est-à-dire quand il ne reste plus de motif à traiter, la variable Q contient bien l'ensemble $Q = \bigcup_{i=0}^{\infty} Q_i$. On a ajouté deux améliorations : si un motif est divisible par un motif déjà dans Q, il est inutile de le traiter car il ne fournira que des motifs divisibles par des motifs déjà obtenus (pour que ce soit efficace, il faut avoir une stratégie pour choisir un motif dans R qui favorise les motifs qui ont le plus de chances d'en diviser d'autres : le plus simple est de commencer par les motifs les plus courts); et si on trouve un motif qui divise w_0, il est inutile de continuer. Nous montrons à la section 6.6 le fonctionnement de cet algorithme sur un exemple, en particulier la manière dont peuvent être calculées les images inverses.

6.4 Cas des morphismes non expansifs

Il est parfois utile de considérer des HDOL-systèmes $G = (\Sigma, f, w_0, \Sigma', g)$ où f n'est pas expansif; c'est le cas en particulier s'il y a une lettre $b \in \Sigma$ telle que $f(b) = b$. Bien sûr, l'algorithme peut toujours être utilisé dans ce cas, mais il n'y a pas de garantie qu'il s'arrêtera au bout d'un temps fini.

Une solution est de construire, au moyen de la proposition 3.19, un nouveau HDOL-système $\widetilde{G} = (\widetilde{\Sigma}, \widetilde{f}, \widetilde{w}_0, \Sigma', \widetilde{g})$ équivalent à G dans le sens que les langages engendrés ont les mêmes facteurs, et tel que $\widetilde{f}$ est partout croissant, et d'appliquer l'algorithme à ce nouveau système.

Nous allons montrer que, si G est un HDOL-système circulaire satisfaisant les conditions de la proposition 3.18, et que de plus la première lettre de w_0 n'est pas de rang 0 (cette dernière condition permet d'utiliser la version simplifiée de la construction, et n'est pas très restrictive puisqu'en général w_0 est égal à une lettre non de rang 0; on a vu au théorème 3.2 que pour prouver qu'un motif p sans constante est k-DOL-évitable ou k-HDOL-évitable, il suffit de considérer des L-systèmes de ce type), alors les nouveaux morphismes $\widetilde{f}$ et $\widetilde{g}$ sont circulaires sur $F(L(\widetilde{G}^0))$, donc l'algorithme est applicable au HDOL-système $\widetilde{G}$.

Remarque : Pour pouvoir appliquer la proposition 3.19 à un HDOL-système quelconque, il faut lui avoir fait subir toutes les transformations nécessaires pour qu'il satisfasse la proposition 3.18. Il faudrait donc vérifier que, si le HDOL-système de départ est circulaire, il le reste après chacune de ces transformations. Il est déjà clair que le DOL-système sous-jacent reste circulaire, sinon on obtiendrait un langage répétitif. Le morphisme de codage étant en général peu transformé, il est assez facile de montrer qu'il reste circulaire, en utilisant au besoin les lemmes 6.1 et 6.2 (la seule transformation qui pose problème est le passage à un morphisme élémentaire, proposition 3.7).

Lemme 6.1. *Soient $f\colon \Sigma^* \to \Sigma'^*$ et $g\colon \Sigma'^* \to \Sigma''^*$ deux morphismes. Supposons que f est circulaire sur un langage factoriel $U \in \mathscr{P}(\Sigma^*)$ et que g est circulaire sur $V = F(f(U))$. Alors $g \circ f$ est circulaire sur U.*

Preuve : Soit M la longueur maximale d'une image de lettre par g, et L_f et L_g les délais de synchronisation de f et g sur U et V. Nous allons montrer que $g \circ f$ est circulaire sur U de délai $L = 2L_g + ML_f$. Soit w un mot de Σ''^L. On a $w = d_w|_V g(c_w)|_V e_w$ avec $|d_w| \leqslant L_g$, $|e_w| \leqslant L_g$ et $|c_w| \geqslant L_f$. Le mot c_w a donc un point de synchronisation $c_w = u_1|_U u_2$ pour f, et on en déduit facilement que w a un point de synchronisation $w = d_w g(u_1)|_U g(u_2) e_w$ pour $g \circ f$. Comme de plus $g \circ f$ est injectif sur U, on en déduit que $g \circ f$ est circulaire sur U. ■

Lemme 6.2. *Soient $f\colon \Sigma^* \to \Sigma'^*$ et $g\colon \Sigma'^* \to \Sigma''^*$ deux morphismes. Supposons que $g \circ f$ est circulaire sur un langage factoriel $U \in \mathscr{P}(\Sigma^*)$, et que g est injectif sur $V = F(f(U))$. Alors f est circulaire sur U et g est circulaire sur V.*

Preuve : Soit L le délai de synchronisation de $g \circ f$ sur U. Soit u un mot de Σ^* suffisamment grand pour que $|g(f(u))| \geqslant L$; le mot $g(f(u))$ a alors un point de synchronisation $g(f(u)) = z_1|_U z_2$ par $g \circ f$, donc u se factorise en $u = u_1 u_2$ de telle sorte que $z_i = g(f(u_i))$.

On peut alors dire que $g(f(u)) = z_1|_V z_2$ pour g. Supposons en effet qu'on a $v \in V$ et $w_1, w_2 \in W = F(g(v))$ tels que $g(v) = w_1 g(f(u)) w_2$; v est lui-même facteur d'un mot de $f(U)$, disons $f(u') = v_1 v v_2$. Alors $g(f(u')) = g(v_1) w_1 z_1 z_2 w_2 g(v_2)$ donc il existe $u'_1, u'_2 \in U$ tels que $u' = u'_1 u'_2$, $g(f(u'_1)) = g(v_1) w_1 z_1$ et $g(f(u'_2)) = z_2 w_2 g(v_2)$. La longueur de $f(u'_i)$ est supérieure à celle de v_i, donc il existe $v'_1, v'_2 \in V$ tels que $f(u'_1) = v_1 v'_1$ et $f(u'_2) = v'_2 v_2$. Finalement, on a $g(v'_1) = w_1 z_1$ et $g(v'_2) = z_2 w_2$ avec $v = v'_1 v'_2$, ce qui est bien ce qu'on cherchait.

Si maintenant w est un mot suffisamment long de Σ''^*, soit w n'est pas dans W et on peut mettre un point de synchronisation pour g sur V n'importe où dans w, soit w contient un mot de la forme $g(f(u))$ et on a vu qu'un tel mot avait un point de synchronisation pour g. Le morphisme g est donc circulaire sur V.

Montrons de même que $f(u) = f(u_1)|_U f(u_2)$ pour f. Supposons qu'on a $u' \in U$ et $v_1, v_2 \in V$ tels que $f(u') = v_1 f(u) v_2$. Alors $g(f(u')) = g(v_1) g(f(u)) g(v_2)$ donc il existe $u'_1, u'_2 \in U$ tels que $u' = u'_1 u'_2$, $g(f(u'_1)) = g(v_1) g(f(u_1))$ et $g(f(u'_2)) = g(f(u_2)) g(v_2)$. Comme g est injectif sur V, on en déduit que $f(u'_1) = v_1 f(u_1)$ et $f(u'_2) = f(u_2) v_2$. On conclut comme ci-dessus que f est circulaire sur U. ∎

Passons maintenant à la preuve de la circularité de $\widetilde{G}$. Rappelons que $\widetilde{f}$ et $\widetilde{g}$ sont définis sur un nouvel alphabet $\widetilde{\Sigma} = \Sigma_1 \Sigma_0^* \cap F(L(G^0)\Sigma_1)\Sigma_1^{-1}$, où Σ_0 est l'ensemble des lettres de rang 0 et Σ_1 son complémentaire, et satisfont le diagramme commutatif suivant :

$$\begin{tikzcd}
\widetilde{\Sigma}^* \arrow[r, "\widetilde{f}"] \arrow[d, "i"] & \widetilde{\Sigma}^* \arrow[d, "i"] \arrow[dr, "\widetilde{g}"] & \\
\Sigma^* \arrow[r, "f"] & \Sigma^* \arrow[r, "g"] & \Sigma'^*
\end{tikzcd}$$

où $i \colon \widetilde{\Sigma}^* \to \Sigma^*$ est tel que $i([av]) = av$.

Le morphisme i est clairement circulaire (sur $\widetilde{\Sigma}^*$), à cause du rôle de marqueur joué par les lettres de Σ_1. D'après le lemme 6.1, les morphismes $i \circ \widetilde{f} = f \circ i$ et $\widetilde{g} = g \circ i$ sont circulaires sur $\widetilde{U} = F(L(\widetilde{G}^0))$. On applique ensuite le lemme 6.2 aux deux morphismes $\widetilde{f}$ et i (i est injectif) pour montrer que $\widetilde{f}$ est circulaire sur $\widetilde{U}$.

6.5 Conclusion

Nous avons construit un algorithme qui peut traiter une large classe de HDOL-systèmes. Nous avons pu, au prix d'une certaine complication due à la nécessité de toujours préciser l'ensemble U, nous affranchir de quasiment toutes les limitations sur le morphisme f (rappelons tout de même que f doit être expansif quand le motif contient des constantes ou quand l'axiome n'est pas une lettre). Cela permet de traiter des morphismes courants comme celui de Thue-Morse qui ne sont pas circulaires. Dans le cas particulier du morphisme de Thue-Morse θ, on aurait pu contourner le problème en utilisant un HDOL-système équivalent dont les morphismes sont circulaires : $(\{a,b\},\theta,a)$ est équivalent à $(\{a,b,c\},\mu,a,\{a,b\},\pi)$ où

$$\begin{array}{rrcl} \mu: & \{a,b,c\}^* & \longrightarrow & \{a,b,c\}^* \\ & a & \longmapsto & abc \\ & b & \longmapsto & ac \\ & c & \longmapsto & b \end{array}$$

et

$$\begin{array}{rrcl} \pi: & \{a,b,c\}^* & \longrightarrow & \{a,b\}^* \\ & a & \longmapsto & abb \\ & b & \longmapsto & ab \\ & c & \longmapsto & a \end{array} .$$

Mais rien ne garantissait qu'on aurait pu en faire autant avec d'autres morphismes (c'est peut-être possible, en utilisant des méthodes similaires à celles de la section 3.4).

Toutefois, il est plutôt ennuyeux que le morphisme h doive être injectif sur $U = F(L(G^0))$: le cas le plus courant et en effet celui du morphisme lettre à lettre, qui ne peut pas être injectif quand l'alphabet d'arrivée est plus petit que celui de départ. Peut-être est-il possible, de même qu'on s'est ramené à g non effaçant (proposition 3.14), de transformer g en un morphisme injectif sur U, ou du moins presque injectif, au sens qu'un mot est uniquement déterminé par son image à l'exception de bords de longueur bornée.

Tel qu'il est, cet algorithme nous a déjà permis d'établir la k-évitabilité d'un grand nombre de motifs (voir annexe A). Mais si on parvient à le généraliser à tous les HDOL-systèmes, il permettra certainement de démontrer des résultats généraux sur la HDOL-évitabilité. En particulier, cela impliquerait que la k-HDOL-évitabilité est semidécidable (puisque l'ensemble des HDOL-systèmes sur k lettres est dénombrable) et donc, si par ailleurs la conjecture 8.1 est démontrée, cela impliquerait la décidabilité de la k-évitabilité.

6.6 Exemple

Le but de cette section est d'étudier la mise en pratique de notre algorithme sur un exemple : montrer que le DOL-langage $L(\Sigma,\mu,a)$ évite le motif $p = AA$, où $\Sigma = \{a, b, c\}$ et μ est le morphisme

$$\begin{array}{rrcl} \mu: & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & abc \\ & b & \longmapsto & ac \\ & c & \longmapsto & b \end{array} .$$

On notera que μ est expansif car μ^2 est partout croissant.

6.6.1 μ est circulaire de délai de synchronisation 2

On a en effet les synchronisations suivantes :

$$|a,\ c|,\ |b|b|\,.$$

et tout mot de longueur au moins 2 contient a, c ou bb comme facteur.

6.6.2 L'ensemble S_μ

S_μ contient 9 *petits mots* (0 ou 1 point de synchronisation) :

$$|a,\ b,\ c|,\ |ab,\ bc|,\ c|a,\ bc|a,\ c|ab,\ bc|ab\,.$$

6.6.3 Le complémentaire de S_μ

Les autres mots non vides, s'ils sont facteurs de mots de C^*, se factorisent de l'une des 9 manières suivantes (*grands mots*) :

$$\begin{gathered} |\mu(w)|,\ |\mu(w)|a,\ |\mu(w)|ab,\ c|\mu(w)|,\ c|\mu(w)|a, \\ c|\mu(w)|ab,\ bc|\mu(w)|,\ bc|\mu(w)|a,\ bc|\mu(w)|ab\,. \end{gathered}$$

6.6.4 Boucle principale

Initialisation de Q et R

$$Q \leftarrow \emptyset, R \leftarrow \{AA\}\,.$$

Motif à traiter : $q = AA$

1. On réécrit chaque variable (ici A seulement) de chacune des 18 manières déterminées en 6.6.2 et 6.6.3 :

$$\begin{gathered} |a|a,\ bb,\ c|c|,\ |ab|ab,\ bc|bc|, \\ c|ac|a,\ bc|abc|a,\ c|abc|ab,\ bc|abbc|ab, \\ |A|A|,\ |A|a|A|a,\ |A|ab|A|ab,\ c|A|c|A|,\ c|A|ac|A|a, \\ c|A|abc|A|ab,\ cb|A|bc|A|,\ bc|A|abc|A|a,\ bc|A|abbc|A|ab\,. \end{gathered}$$

2. On calcule les antécédents par μ de ces mots. Il peut ne pas y en avoir (s'ils contiennent des facteurs interdits, c'est-à-dire des mots qui ne sont facteurs d'aucun mot de C^*) ou y en avoir plusieurs (à cause d'ambiguïtés aux extrémités des mots) :

$$\begin{gathered}\emptyset,\ \{cc\},\ \emptyset,\ \emptyset,\ \emptyset,\\ \{aba, abb, bba, bbb\},\ \{aaa, aab\},\ \{aaa, baa\},\ \emptyset,\\ \{AA\},\ \emptyset,\ \emptyset,\ \emptyset,\ \{aAbAa, aAbAb, bAbAa, bAbAb\},\\ \{aAaAa, bAaAa\},\ \emptyset,\ \{aAaAa, aAaAb\},\ \emptyset\,.\end{gathered}$$

3. On élimine les motifs déjà obtenus (AA) ou qui contiennent un tel motif (par exemple $aAbAb$). Il reste :

$$aba,\ aAbAa\,.$$

4. On ajoute ces motifs ($\mu^{-1}(q)$) à R et on transfère q de R à Q :

$$Q \leftarrow \{AA\}, R \leftarrow \{aba, aAbAa\}\,.$$

Motif à traiter : $q = aba$

1. Pas de variable, donc pas de réécriture, on garde aba.
2. Pas d'antécédent, car aba est un facteur interdit.
3. Donc pas de nouveau motif.
4. On transfère q de R à Q :

$$Q \leftarrow \{AA, aba\}, R \leftarrow \{aAbAa\}\,.$$

Motif à traiter : $q = aAbAa$

1. La réécriture de A donne :

$$\begin{gathered}a|ab|aa,\ abbba,\ ac|bc|a,\ a|abb|aba,\ abc|bbc|a,\\ ac|abc|aa,\ abc|abbc|aa,\ ac|abbc|aba,\ abc|abbbc|aba,\\ a|A|b|A|a,\ a|A|ab|A|aa,\ a|A|abb|A|aba,\ ac|A|bc|A|a,\ ac|A|abc|A|aa,\\ ac|A|abbc|A|aba,\ abc|A|bbc|A|a,\ abc|A|abbc|A|aa,\ abc|A|abbbc|A|aba\,.\end{gathered}$$

2. Pas d'antécédent.
3. Donc pas de nouveau motif.
4. On transfère q de R à Q :

$$Q \leftarrow \{AA, aba, aAbAa\}, R \leftarrow \emptyset\,.$$

Fin de la boucle : $R = \emptyset$

6.6.5 Résultat

On a obtenu les motifs suivants :

$$AA\,,\ aba\,,\ aAbAa\,.$$

Cela signifie que $Q = \{AA, aba, aAbAa\}$ est invariant par μ^{-1}. Comme la lettre a évite tous ces motifs, on peut conclure :

$$L(\Sigma, \mu, a) \text{ évite } p\,.$$

Chapitre 7

Une autre application des morphismes circulaires : calculs de complexités de suites

Dans ce chapitre, nous étudions un problème assez différent de celui de l'évitabilité des motifs, mais auquel la notion de morphisme circulaire utilisée au chapitre précédent apporte un éclairage nouveau.

7.1 Introduction

Soit $\mathbf{u}$ un suite infinie sur un alphabet (fini) Σ. On appelle *complexité* de $\mathbf{u}$ la fonction $p\colon \mathbb{N} \to \mathbb{N}$ où, pour tout $n \geqslant 0$, $p(n)$ est le nombre de facteurs de $\mathbf{u}$ dont la longueur est n.

J.-P. Allouche et J. Berstel ont posé le problème suivant : pour quels entiers α et β existe-t-il des suites dont la complexité est la fonction affine $p(n) = \alpha n + \beta$? Si on veut que ce soit vrai pour tout $n \geqslant 0$, la seule possibilité est $\beta = 1$ et $\alpha = \#\,\mathrm{Alph}\,(\mathbf{u}) - 1$ (sur un alphabet binaire, cela correspond aux suites sturmiennes, de complexité $p(n) = n + 1$). On se contente donc de fonctions de complexité ultimement affines, c'est-à-dire qu'il existe un rang n_0 tel que pour tout $n \geqslant n_0$ on a $p(n) = \alpha n + \beta$.

Allouche a développé plusieurs techniques pour construire, à partir d'une suite de complexité ultimement $\alpha n + \beta$, des suites de complexités différentes, par exemple $\alpha n + \beta + 1$. Cependant il reste des couples (α, β) qui ne peuvent être obtenus par ces techniques en partant des quelques suites de complexité ultimement affine déjà connues. Il est donc nécessaire de construire de nouveaux points de départ. C'est dans cet objectif que cette étude a été faite, mais celui-ci a été dépassé puisqu'on obtient directement toutes les complexités ultimement affines possibles.

7.2 Suites ou langages?

À toute suite $\mathbf{u}$ on associe le langage de ses facteurs $F(\mathbf{u})$. C'est un langage factoriel et prolongeable à droite.

Réciproquement, soit $L \subset \Sigma^*$ un langage factoriel prolongeable à droite et *transitif*, c'est-à-dire tel que

$$\forall x, y \in L, \exists z \in L, xzy \in L \ .$$

Alors il existe une suite $\mathbf{u}$ telle que $L = F(\mathbf{u})$. Mais en fait on n'a pas besoin d'une telle suite pour étudier la complexité. Par contre, il est utile de supposer le langage L *prolongeable* :

$$\forall x \in L, \exists a, b \in \Sigma, axb \in L$$

ce qui interdit les facteurs apparaissant uniquement en début de suite.

On s'intéresse donc à la classe des langages factoriels prolongeables, qui est stable par union et par application de morphisme suivie d'une clôture factorielle (partant du langage $L \subset \Sigma^*$, on construit $F(f(L)) \subset \Sigma'^*$ pour tout morphisme $f \colon \Sigma^* \to \Sigma'^*$). Les langages résultant de nos constructions étant tous transitifs, on pourra donc les voir indifféremment comme des constructions de langages ou des constructions de suites.

À tout morphisme $f \colon \Sigma^* \to \Sigma^*$, on associe le langage

$$FL(f) = F\left(\{\ f^n(x) \ \mid n \in \mathbb{N} \text{ et } x \in \Sigma \ \}\right)$$

(c'est la clôture factorielle d'un *DOL-langage sans axiome*, c'est-à-dire un DOL-langage où toutes les lettres de l'alphabet servent d'axiome), qui est prolongeable pourvu que toutes les lettres apparaissent comme facteurs internes, c'est-à-dire

$$\forall x \in \Sigma, \exists a, b \in \Sigma, axb \in FL(f)$$

ce qui exclut les morphismes du type $(a \mapsto ab, b \mapsto bb)$. Si de plus il existe une lettre $a \in \Sigma$ telle que $f(a) \in a\Sigma^*$ et que $\bigcup_{n \in \mathbb{N}} \mathrm{Alph}\,(f^n(a)) = \Sigma$, la suite de mots $(f^n(a))$ a pour limite un mot infini $\mathbf{u} = f^\omega(a)$ dont l'ensemble des facteurs est exactement $FL(f)$.

7.3 Facteurs spéciaux et bispéciaux

7.3.1 Facteurs spéciaux

On suppose pour le moment que tous les mots sont sur l'alphabet $\Sigma = \{a, b\}$. Soit L un langage factoriel prolongeable sur Σ.

Un mot $u \in L$ est dit *spécial à droite* si les mots ua et ub sont tous deux dans L, *spécial à gauche* si au et bu sont dans L. Un facteur spécial correspond à un choix de prolongement, ce qui se voit aisément dans les *arbres de facteurs* et les *graphes de Rauzy* de L, dont nous allons détailler la construction.

L'*arbre des facteurs à droite* (resp. *à gauche*) de L est un arbre infini dont les sommets sont étiquetés par les éléments de L et les arêtes par les lettres, de sorte qu'il y a une arête d'étiquette x entre les sommets u et v si et seulement si $v = ux$ (resp. $v = xu$). On le représente conventionnellement en plaçant la racine, étiquetée par le mot vide, à gauche (resp. à droite), si bien que quand on lit de la gauche vers la droite les étiquettes du chemin entre la racine et le sommet étiqueté u on retrouve le mot u; on peut alors se contenter d'indiquer les étiquettes des arêtes, ou la dernière lettre des étiquettes de sommet (cf. figure 7.1). Dans cet arbre, chaque facteur spécial à droite (resp. à gauche) correspond à une bifurcation.

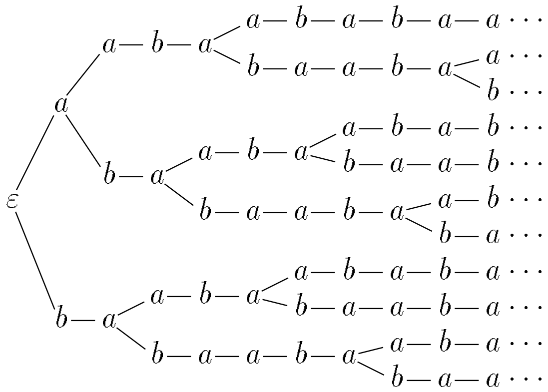

FIG. 7.1 - *Arbre à droite des facteurs de la suite de Fibonacci*

Pour tout entier n, le *graphe de Rauzy d'ordre n* de L est le graphe fini dont les sommets sont étiquetés par les mots de L de longueur n et tel qu'il y a une arête entre les sommets u et v si et seulement s'il existe deux lettres x et y telles que $ux = yv$. Un mot spécial à droite correspond à un sommet d'où partent deux arêtes, et un mot spécial à gauche à un sommet où arrivent deux arêtes (cf. figure 7.2).

Le nombre de facteurs spéciaux de longueur n à gauche ou à droite est le même : c'est $s(n) = sg(n) = sd(n) = p(n+1) - p(n)$.

7.3.2 Facteurs bispéciaux

Lorsqu'un facteur u est spécial à la fois à gauche et à droite, on dit qu'il est *bispécial*. Trois cas peuvent se présenter, selon le nombre d'éléments de $L \cap \Sigma u \Sigma$: si cet ensemble a quatre éléments (le maximum), on dit que u est *bispécial strict*. S'il n'en a que trois, u est *bispécial ordinaire*. Enfin, il est

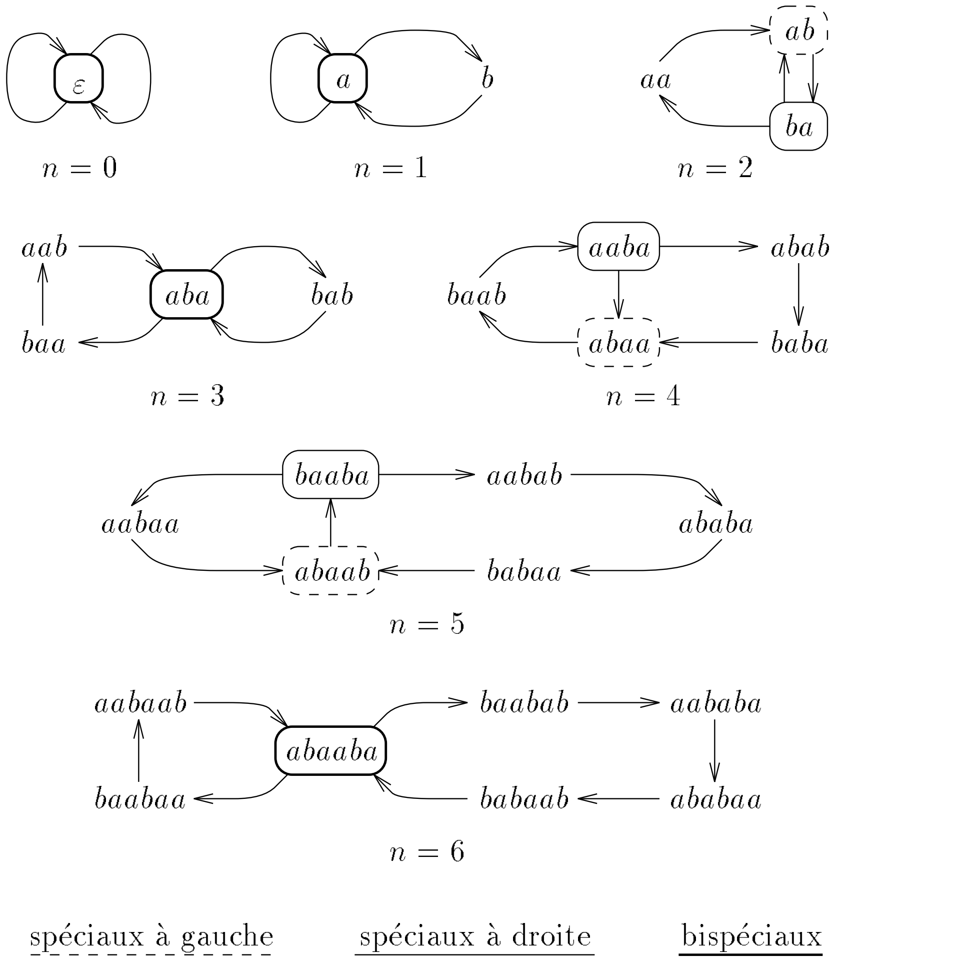

FIG. 7.2 - *Graphes de Rauzy d'ordre 0 à 6 de la suite de Fibonacci*

possible que cet ensemble ait deux éléments ($\{aua, bub\}$ ou $\{aub, bua\}$) et on dit dans ce cas que u est *bispécial faible*.

Quand on prolonge à droite un facteur spécial à gauche u, on obtient un ou deux facteurs qui peuvent être spéciaux à gauche ou non, en fonction de la nature de u :

- aucun n'est spécial à gauche si u est bispécial faible;
- un seul est spécial à gauche si u est non bispécial ou bispécial ordinaire;
- les deux sont spéciaux à gauche si u est bispécial strict.

Réciproquement, un préfixe de facteur spécial à gauche est spécial à gauche, donc on peut construire l'*arbre des facteurs spéciaux à gauche*, qui est un sous-arbre de l'arbre des facteurs à droite, dans lequel on ne garde que les sommets dont les étiquettes sont spéciales à gauche. Dans cet arbre, un bispécial strict indique un embranchement et un bispécial faible indique une fin de branche. Cet arbre est souvent plus facile à décrire que l'arbre de tous les facteurs, le cas extrême étant celui des suites sturmiennes pour lesquelles il est filiforme. On peut bien entendu aussi construire l'arbre des facteurs spéciaux à droite, qui est orienté vers la gauche. Pour bien indiquer qu'il s'agit de facteurs spéciaux, on indique à la racine les deux lettres a et b par lesquelles les facteurs spéciaux peuvent se prolonger. (figures 7.3 et 7.4).

$$\cdots\ b - a - b - a - a - b - a - a - b - a - b - a - a - b - a - \varepsilon \;\left|\; \begin{matrix} a \\ b \end{matrix}\right.$$

FIG. 7.3 - *Arbre des facteurs spéciaux à droite pour la suite de Fibonacci*

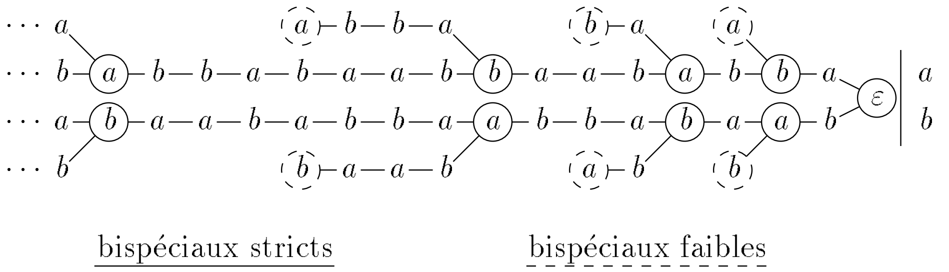

FIG. 7.4 - *Arbre des facteurs spéciaux à droite pour la suite de Thue-Morse*

Les différents types de facteurs bispéciaux interviennent également lors de l'expansion des graphes de Rauzy : pour passer du graphe des facteurs de longueur n à celui des facteurs de longueur $n+1$, on remplace chaque arête par un sommet et chaque sommet par une, deux, trois ou quatre arêtes selon que le mot qui étiquette ce sommet est non spécial, spécial non bispécial ou bispécial faible, bispécial ordinaire, bispécial strict (cf. figure 7.5). En particulier, si aucun facteur de longueur n n'est bispécial, il n'y a qu'une expansion possible.

Si on note $bs(n)$, $bo(n)$ et $bf(n)$ le nombre de facteurs bispéciaux de longueur n de chaque type, on établit aisément la relation $bs(n) - bf(n) = s(n+1) - s(n)$.

Pour calculer $p(n)$, il est donc suffisant de connaître $bs(n)$ et $bf(n)$.

On peut également utiliser des séries génératrices : à la fonction p on associe la série $P(X) = \sum_{n=0}^{\infty} p(n)X^n$, et de même $S(X)$, $BS(X)$ et $BF(X)$ pour s, bs et bf. En supposant que $s(0) = 1$, c'est-à-dire que les lettres a et b sont toutes

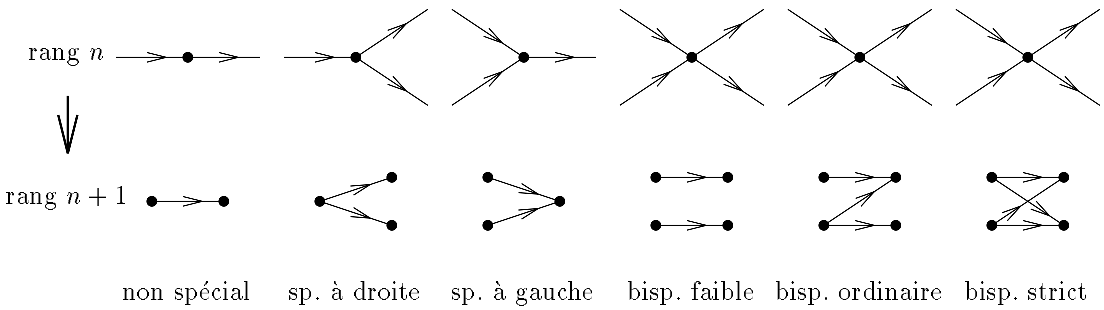

FIG. 7.5 - *Expansion des graphes de Rauzy*

deux utilisées (sinon le mot vide ne serait pas spécial), on a alors les relations suivantes :

$$P(X) = \frac{1}{1-X}\,(1 + XS(X))$$

$$S(X) = \frac{1}{1-X}\,(1 + X(BS(X) - BF(X)))$$

soit

$$S(X) = \frac{1}{(1-X)^2}\Big(1 + X^2(BS(X) - BF(X))\Big) \quad .$$

7.3.3 Le cas des alphabets plus grands

Quand l'alphabet Σ a plus de deux éléments, la situation se complique quelque peu, surtout en ce qui concerne les facteurs bispéciaux. Supposons donc que Σ a k éléments, et soit comme ci-dessus L un langage factoriel prolongeable sur Σ.

Nous dirons qu'un mot $u \in L$ est *spécial à droite d'ordre* m, où m est un entier strictement positif, si le mot u se prolonge à droite de $m+1$ manières différentes, c'est-à-dire si l'ensemble $\{\ x \in \Sigma \mid ux \in L\ \}$ a exactement $m+1$ éléments. Sur l'alphabet binaire, tous les facteurs spéciaux à droite sont d'ordre 1, mais si l'alphabet est plus grand ils n'ont pas nécessairement tous le même ordre. Les facteurs spéciaux à gauche d'ordre m sont définis de la même manière.

Comme avec un alphabet binaire, les facteurs spéciaux correspondent à des embranchements dans les arbres de facteurs et les graphes de Rauzy. Le nombre de branches est égal à l'ordre du facteur spécial, augmenté d'une unité.

Pour calculer la complexité, ce n'est plus le nombre de facteurs spéciaux qui est important, car il faut les compter avec multiplicité : si $sd_m(n)$ est le nombre de facteurs spéciaux à droite de longueur n et d'ordre m, et de même $sg_m(n)$

à gauche, alors on a

$$s(n) = \sum_{m=1}^{k-1} m.sg_m(n) = \sum_{m=1}^{k-1} m.sd_m(n) = p(n+1) - p(n) \ .$$

Un facteur u spécial à droite et à gauche est encore dit *bispécial.* A tout facteur bispécial sont associés deux entiers m_d et m_g, que nous appellerons son *ordre à droite* et son *ordre à gauche*, et qui correspondent à son ordre en tant que facteur à droite et à gauche respectivement. Ces deux nombres n'ont aucune raison d'être égaux. Nous allons encore classer les facteurs bispéciaux en fonction du nombre d'éléments de l'ensemble $L \cap \Sigma u \Sigma$. Celui-ci a au plus $(m_d+1)(m_g+1)$ éléments, et si c'est le cas on dit que u est *bispécial strict*; il a au moins $\max(m_d, m_g)+1$ éléments, car chaque prolongement de u à droite ou à gauche doit correspondre à au moins un prolongement bilatéral : c'est le cas que nous avons appelé *bispécial faible* pour un alphabet binaire. Mais cette fois tous les intermédiaires sont possibles entre ces deux extrêmes : on peut quantifier cela au moyen d'un troisième entier m, l'*ordre* du facteur bispécial, défini ainsi :

$$m = \#\,(L \cap \Sigma u \Sigma) - m_d - m_g - 1 \ .$$

L'ordre m est un entier relatif, compris entre $-\min(m_d, m_g)$ et $m_d m_g$, donc entre $-(k-1)$ et $(k-1)^2$; si $k=2$, on retrouve les trois catégories de facteurs bispéciaux, qui correspondent aux trois ordres possibles, -1, 0 et 1.

Quand on prolonge à droite un facteur spécial à gauche u, la nature de u permet de déterminer, non le nombre de facteurs spéciaux à gauche obtenus, mais leur ordre total :

- si u n'est pas bispécial, on obtient un facteur spécial à gauche de même ordre que u;

- si u est bispécial d'ordre m, on obtient un certain nombre de facteurs spéciaux à gauche dont l'ordre total est l'ordre à gauche de u augmenté de l'ordre de u.

Cela donne la relation suivante :

$$b(n) = \sum_{m=-(k-1)}^{(k-1)^2} m.b_m(n) = s(n+1) - s(n)$$

où $b_m(n)$ est le nombre de facteurs bispéciaux d'ordre m et de longueur n. Si $P(X)$, $S(X)$ et $B(X)$ sont les séries génératrices associées à $p(n)$, $s(n)$ et $b(n)$, on a les formules suivantes :

$$P(X) = \frac{1}{1-X}\,(1 + XS(X))$$

$$S(X) = \frac{1}{1-X}\left(s(0) + XB(X)\right)$$

(où $s(0) = p(1) - 1$ vaut $k-1$ si toutes les lettres sont utilisées), soit

$$S(X) = \frac{1}{(1-X)^2}\Big(1 + (p(1)-2)X + X^2B(X)\Big) \quad .$$

Pour pouvoir parler d'arbre de facteurs spéciaux, il faut un critère plus fin que l'ordre : étant donné une partie Σ' de Σ à au moins deux éléments, on construit l'*arbre des facteurs spéciaux à gauche de type* Σ' en ne conservant dans l'arbre des facteurs à droite que ceux qui se prolongent à gauche par chacune des lettres de Σ'. L'ensemble des facteurs spéciaux à gauche est donc décrit par une famille d'arbres (un pour chaque partie Σ'), mais en pratique on peut se limiter aux arbres non vides avec Σ' maximal (i.e. aucune partie plus grande ne donne le même arbre), ce qui en général donne un petit nombre d'arbres, qu'on pourra appeler *base* de la famille des arbres des facteurs spéciaux à gauche. Dans ces arbres, tous les embranchements et toutes les extrémités de branches sont étiquetés par des facteurs bispéciaux, et réciproquement tous les facteurs bispéciaux d'ordre non nul donnent un embranchement ou une extrémité sur au moins l'un des arbres de base : on peut donc déterminer les facteurs bispéciaux qui contribuent à la complexité par simple observation de la base, et il est alors facile de déterminer leur ordre.

7.4 Action d'un morphisme circulaire

Soit $f\colon \Sigma^* \to \Sigma'^*$ un morphisme circulaire de délai de synchronisation l (voir page 89), qu'on supposera de plus préfixe (si $f(u)$ est préfixe de $f(v)$, alors u est préfixe de v) pour simplifier. En fait, ce qui compte est que f est à délai de déchiffrage borné, et c'est le cas de tout morphisme circulaire.

Soit $L \subset \Sigma^*$ un langage factoriel prolongeable, et $L' = F(f(L))$ la clôture factorielle de son image par f. Nous allons montrer qu'il est possible de construire l'arbre (ou les arbres) des facteurs spéciaux à droite de L' en fonction de celui (ceux) de L.

Soit $u \in \Sigma'^*$ spécial à droite dans L' de longueur supérieure ou égale à l. Le mot u peut s'écrire $u = u_1|f(u_2)|u_3$ (rappelons que les barres de mesure indiquent les points de synchronisation) où u_1 et u_3 sont des mots de L' de longueur au plus l, et u_2 est un mot de L. En fait, puisqu'on a supposé f préfixe, u_3 est un préfixe strict d'image de lettre.

Comme u est spécial à droite, il se prolonge au moins de deux manières ua et ub. Les mots u_3a et u_3b ne peuvent être préfixes de la même image de lettre, donc u_2 se prolonge par au moins deux lettres différentes x et y, avec u_3a préfixe de $f(x)$ et u_3b préfixe de $f(y)$, c'est-à-dire que u_2 est spécial à droite dans L; son ordre est supérieur ou égal à celui de u.

Réciproquement, si v est un facteur spécial à droite de L qui se prolonge en vx et vy, soit p le plus grand préfixe commun à $f(x)$ et $f(y)$ (qui est un préfixe strict de ces deux mots). Alors $f(v)p$ est spécial à droite dans L'. Plus précisément, si v est d'ordre m, il se prolonge de $m+1$ manières différentes vx_0, vx_1, ..., vx_m. L'ensemble des préfixes des $f(x_i)$ forme un arbre à droite fini, dans lequel chaque embranchement correspond à un plus grand préfixe commun à au moins deux des $f(x_i)$. Si on note p_1, ..., p_r ces préfixes, et pour tout j compris entre 1 et r on note H_j l'ensemble des lettres $a \in \Sigma'$ telles que $p_j a$ est préfixe de l'un des $f(x_i)$, on trouve que $f(v)p_j$ est spécial à droite d'ordre au moins $\#H_j - 1$ dans L'. La somme des ordres des r facteurs spéciaux obtenus est donc supérieure ou égale à l'ordre de v. Il peut ne pas y avoir égalité si $f(v)$ est facteur de l'image par f d'un mot qui ne contient pas v.

On voit donc que, connaissant les facteurs spéciaux à droite dans L, on peut calculer les facteurs spéciaux à droite dans L' en appliquant f et en ajoutant un ou plusieurs suffixes p qui ne dépendent que des lettres par lesquelles le facteur considéré se prolonge. On obtient ainsi tous les facteurs spéciaux à droite dans L', sauf peut-être ceux de longueur inférieure à l, qu'il faut déterminer à la main, et que nous appellerons *facteurs spéciaux exceptionnels.*

On obtient de la même manière les facteurs spéciaux à gauche si f est un morphisme suffixe, et tous les types de facteurs bispéciaux si f est bipréfixe.

7.5 Application : suites de complexité affine

7.5.1 Suites de complexité ultimement affine

Revenons au problème initial : on cherche à construire des suites de complexité ultimement affine $p(n) = \alpha n + \beta$ sur un alphabet binaire $\Sigma = \{\alpha, \beta\}$. Pour que ce soit possible, il faut que $(\alpha, \beta) \in \{0,1\} \times (\mathbb{N} \setminus \{0\}) \cup (\mathbb{N} \setminus \{0,1\}) \times \mathbb{Z}$. On va voir que cette condition est suffisante.

Théorème 7.1. *Soient $j \geqslant 0$, $k \geqslant 1$ et $l \geqslant 1$ trois entiers. Soit $\Sigma_k = \{c_1, \ldots, c_k\}$ un alphabet à k lettres. On considère les deux morphismes f et g définis ainsi :*

$$f : \begin{array}{rcl} \Sigma_k^* & \longrightarrow & \Sigma_k^* \\ c_i & \longmapsto & c_1 c_{i+1} \text{ si } i \neq k \\ c_k & \longmapsto & c_1 \end{array} \qquad \text{et} \qquad g : \begin{array}{rcl} \Sigma_k^* & \longrightarrow & \Sigma^* \\ c_i & \longmapsto & a^l b^{i+j} \end{array}$$

pour tout i compris entre 1 et k.

La suite $\mathbf{u} = g(f^\omega(c_1))$ a pour complexité

$$p(n) = (k-1)n - \frac{k(k-1)}{2} - j(k-2) + l + 1$$

pour $n \geqslant \max(l, j+k-1, j+1)$.

Preuve : On construit tout d'abord la base des arbres des facteurs spéciaux à gauche de $FL(f)$, qui ne contient qu'un seul arbre, de type Σ_k, puisque chaque facteur spécial peut être prolongé par chacune des k lettres de l'alphabet : comme f est un morphisme circulaire suffixe, cet arbre est un point fixe du procédé décrit à la section 7.4, qui ne peut être que filiforme puisqu'il n'y a pas de facteurs spéciaux exceptionnels. Il n'y a donc aucun facteur bispécial d'ordre non nul. En quelque sorte, $f^\omega(c_1)$ est une suite de Fibonacci généralisée, et sa complexité est $(k-1)n+1$.

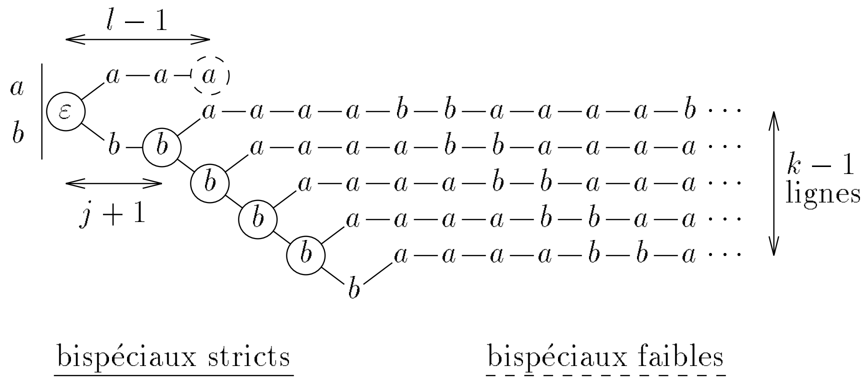

FIG. 7.6 - *Arbre des facteurs spéciaux à gauche de $g(f^\omega(c_1))$ pour $j=1$, $k=6$, $l=4$*

On applique ensuite le morphisme g, qui est circulaire et suffixe, pour obtenir l'arbre des facteurs spéciaux à gauche de $F(g(FL(f)))$ (figure 7.6). Cet arbre comporte $k-1$ branches infinies (chaque facteur spécial pour $FL(f)$, d'ordre $k-1$, donne un élément dans chacune des branches) et une branche finie de longueur $l-1$, contenant les facteurs spéciaux exceptionnels. Il ne reste plus qu'à compter les facteurs bispéciaux pour obtenir la complexité cherchée : il y a un bispécial faible, a^{l-1}, et $k-1$ bispéciaux stricts, ε, b^{j+1}, b^{j+2}, ..., b^{j+k-2}. ■

Corollaire 7.1. *Pour tout couple d'entiers $(\alpha,\beta) \in \{0,1\} \times (\mathbb{N} \setminus \{0\}) \cup (\mathbb{N} \setminus \{0,1\}) \times \mathbb{Z}$, il existe une suite binaire de complexité ultimement $\alpha n + \beta$.*

Preuve : La suite $\mathbf{u}$ définie dans le théorème 7.1 fournit une solution dans tous les cas, avec $k = \alpha + 1 \geqslant 1$ et j et l choisis en fonction de α et β.

Si $k = 1$, la suite $\mathbf{u}$ a pour complexité ultimement $j + l + 1$. Si on prend $j = \left\lfloor \frac{\beta-1}{2} \right\rfloor$ et $l = \left\lfloor \frac{\beta}{2} \right\rfloor$, on a $p(n) = \beta$ pour $n \geqslant \left\lfloor \frac{\beta+1}{2} \right\rfloor$.

Si $k = 2$, la suite $\mathbf{u}$ a pour complexité ultimement $n + l$. On prend donc $l = \beta$ et (par exemple) $j = 0$, et on a $p(n) = n + \beta$ pour $n \geqslant \beta$.

Si $k > 2$, soit par exemple

$$j = \max\left(0, \left\lfloor \frac{-k^2+3k-2-2\beta}{2k-4} \right\rfloor\right)$$

et

$$l = \beta + \frac{k(k-1)}{2} + j(k-2) - 1 \geqslant 1 \ .$$

La suite $\mathbf{u}$ a alors pour complexité $p(n) = \alpha n + \beta$ pour

$$n \geqslant \max\left(\alpha, \frac{\alpha}{2} - \frac{\beta}{\alpha-1}, \beta + \frac{\alpha(\alpha+1)}{2} - 1\right) \ .$$

■

7.5.2 Complexité $\alpha n + \beta$ pour $n \geqslant 1$

La question suivante, variante du problème ci-dessus, a été posée par Pascal Alessandri : pour quelles valeurs de (α, β) existe-t-il une suite de complexité $p(n) = \alpha n + \beta$ pour tout $n \geqslant 1$? Elle peut être résolue par une construction similaire.

Théorème 7.2. *Soit $(\alpha, \beta) \in \mathbb{N} \times \mathbb{Z}$. Il existe une suite de complexité $p(n) = \alpha n + \beta$ pour tout $n \geqslant 1$ (et $p(0) = 1$) si et seulement si $\alpha + \beta \geqslant 1$ et $2\alpha + \beta \leqslant (\alpha + \beta)^2$.*

Preuve : Ces conditions sont évidemment nécessaires : $p(1) = \alpha + \beta$ est le nombre de lettres distinctes apparaissant dans la suite, donc est un nombre strictement positif; et si la suite contient $\alpha + \beta$ lettres distinctes, elle ne peut contenir plus de $(\alpha+\beta)^2$ facteurs de longueur 2, donc $2\alpha + \beta = p(2) \leqslant p(1)^2 = (\alpha + \beta)^2$.

Supposons maintenant que (α, β) satisfait les deux conditions, et que $\alpha > 0$, le cas $\alpha = 0$ étant facile à résoudre avec des suites périodiques. Soit $k = \alpha + 1$ et $l = \alpha + \beta$: on va construire un HDOL-système $G = (\Sigma_k, f, c_1, \Sigma_l, h)$ où $\Sigma_k = \{c_1, \ldots, c_k\}$ et $\Sigma_l = \{d_1, \ldots, d_l\}$, de telle sorte que la suite associée $\mathbf{u} = h(f^\omega(c_1))$ aura la complexité voulue.

Le DOL-système sous-jacent $G^0 = (\Sigma_k, f, c_1)$ est le même que pour le théorème 7.1. La suite $f^\omega(c_1)$ est donc une suite de Fibonacci généralisée, de complexité $\alpha n + 1$ pour tout n.

La construction du morphisme h dépend du signe de β. Supposons tout d'abord que $l \geqslant k$, c'est-à-dire que $\beta > 0$. Si $l = k$, la suite $f^\omega(c_1)$ a déjà la complexité cherchée et il suffit de prendre $h = Id$. Sinon, on définit h par $h(c_1) = d_1 d_{k+1} d_{k+2} \ldots d_l$ et $h(c_i) = d_i$ pour $2 \leqslant i \leqslant k$: le morphisme h

est circulaire et bipréfixe. On voit facilement que la base des arbres de facteurs spéciaux à gauche de $F(\mathbf{u})$ a deux éléments, un arbre filiforme de type $\{d_1, \ldots, d_k\}$ (image par h de l'arbre de $FL(f)$) et un arbre de type Σ_l contenant le seul nœud ε, seul facteur spécial exceptionnel, et seul facteur bispécial d'ordre non nul, $k-l$. On en déduit immédiatement que la complexité est bien $p(n) = \alpha n + \beta$ pour $n \geqslant 1$.

Supposons maintenant que $l < k$, c'est-à-dire $\beta \leqslant 0$. On définit h de la manière suivante : $h(c_1) = d_1$, $h(c_i) = d_1 d_i$ pour $2 \leqslant i \leqslant l$, et $h(c_{l+1})$, ..., $h(c_k)$ sont $k-l$ éléments pris parmi l'ensemble $E = d_1\{d_2, \ldots, d_l\}^2$, qui en compte $(l-1)^2$; comme $k-l = 1-\beta \leqslant (\alpha+\beta)^2 - 2\alpha - 2\beta + 1 = (l-1)^2$, c'est toujours possible. L'ordre des images n'a pas d'importance.

Le morphisme h est suffixe et circulaire, de délai de synchronisation 2, grâce au rôle de marqueur joué par d_1. Nous pouvons donc déterminer les facteurs spéciaux à gauche de $F(\mathbf{u})$ en fonction de ceux de $FL(f)$.

Soit u un facteur spécial à gauche pour $FL(f)$: il se prolonge par chacune des lettres de Σ_k. Les suffixes communs des $f(c_i)$ sont ε et toutes les lettres d_i telles que $E \cap d_1 \Sigma_k d_i \neq \emptyset$; les facteurs spéciaux à gauche pour $F(\mathbf{u})$ obtenus sont donc $f(u)$, d'ordre $l-1$ (il se prolonge par chacune des lettres de $H_1 = \Sigma_l$), et $d_i f(u)$, d'ordre $\#(E \cap d_1 \Sigma_k d_i)$ quand cet ensemble est non vide (il se prolonge par $H_i = \{d_1\} \cup \{\, d_j \mid d_1 d_j d_i \in E \,\}$). Les seuls mots non synchronisés sont ε et certains d_i, mais ils ne donnent pas de facteurs spéciaux exceptionnels. Finalement, la base contient un arbre pour chaque ensemble H_i, avec une ou plusieurs branches filiformes partant de ε (une branche pour chaque j tel que $H_i \subset H_j$). Comme précédemment, ε est le seul facteur bispécial d'ordre non nul, et son ordre vaut $k-l$ (l'ordre de ε est toujours égal à $p(2) - 2p(1) + 1$). ■

7.6 Quelques exemples de calculs de complexité de suites définies par morphisme

7.6.1 Suite de Thue-Morse

La suite de Thue-Morse est le point fixe du morphisme

$$\begin{array}{rccc} \theta\colon & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & ab \\ & b & \longmapsto & ba \end{array}.$$

La figure 7.4 montre l'arbre des facteurs spéciaux à droite de cette suite. Les facteurs bispéciaux sont

<u>bispéciaux stricts :</u> ε, $\theta^m(ab)$, $\theta^m(ba)$ pour $m \geqslant 0$;

<u>bispéciaux ordinaires :</u> a, b;

bispéciaux faibles : $\theta^m(aba)$, $\theta^m(bab)$) pour $m \geqslant 0$.

On a donc :

$$BS(X) - BF(X) = 1 + 2\sum_{n=0}^{\infty}\left(X^{2.2^n} - X^{3.2^n}\right)$$

et on en déduit immédiatement les séries $S(X)$ et $P(X)$. On retrouve ainsi les formules classiques [11] :

$$s(n) = \begin{cases} 1 & \text{si } n = 0 \\ 2 & \text{si } 0 < n \leqslant 2 \\ 4 & \text{si } 2.2^m < n \leqslant 3.2^m \\ 2 & \text{si } 3.2^m < n \leqslant 4.2^m \end{cases}$$

$$p(n) = \begin{cases} 1 & \text{si } n = 0 \\ 2 & \text{si } n = 1 \\ 4 & \text{si } n = 2 \\ 4n - 2.2^m - 4 & \text{si } 2.2^m < n \leqslant 3.2^m \\ 2n + 4.2^m - 2 & \text{si } 3.2^m < n \leqslant 4.2^m \end{cases}$$

où m est un entier positif ou nul.

7.6.2 Un exemple non primitif

Considérons le morphisme suivant :

$$\begin{array}{rccc} f\colon & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & aba \\ & b & \longmapsto & bb \end{array} .$$

Ce morphisme n'est pas primitif, car les images successives de b ne contiennent aucun a (un morphisme f est dit primitif s'il existe un entier n tel que l'image de chaque lettre par f^n contient toutes les lettres). La suite qu'il engendre,

$$f^\omega(a) = ababbababbbbababbababbbbbbbbababbababbbbababbababbbbbbbbbbbbbbbbab\ldots$$

est récurrente (tout facteur apparaît une infinité de fois), mais pas uniformément récurrente puisqu'il existe des facteurs arbitrairement longs qui ne contiennent pas de a.

Les facteurs bispéciaux sont

bispéciaux stricts : $f^m(b)$, de longueur 2^m, pour $m \geqslant 0$;

bispéciaux ordinaires : ε;

bispéciaux faibles : $f^m(bab)$, de longueur $2^m(3+m/2)$, pour $m \geqslant 0$.

On peut donc exprimer BS et BF comme des sommes. Il est également possible d'écrire des équations fonctionnelles :

$$BS(X) = BS(X^2) + X \ ;$$

$$BF(X) = \varphi(X,X) \text{ avec } \varphi(X,Y) = \varphi(XYX, YY) + YXY \ .$$

On peut enfin écrire que $s(n) = 1 + q + r$, où $q = \lceil \log_2 n \rceil$ est le nombre de facteurs bispéciaux stricts de longueur inférieure à n, et $r = \left\lceil \frac{W(128n \log 2)}{\log 2} - 6 \right\rceil$ est le nombre de facteurs bispéciaux faibles de longueur inférieure à n ($W(x)$ est l'unique fonction analytique réelle sur $]-1/e, +\infty[$ telle que $W(x)e^{W(x)} = x$). On calcule ensuite :

$$\begin{aligned} p(n) &= 1 + \sum_{m=0}^{n-1} s(m) \\ &= 1 + n + \sum_{m=0}^{q-1} (n - 1 - 2^m) - \sum_{m=0}^{r-1} (n - 1 - 2^m(3 + m/2)) \\ &= 1 + n + (n-1)q - (2^q - 1) - (n-1)r + (2^{r+1} + 2^{r-1}r - 2) \ . \end{aligned}$$

Dans cette expression, les termes dominants sont nq et nr, soit

$$p(n) = n \frac{\log n - W(128 n \log 2)}{\log 2} + O\left(n\right) \ .$$

On peut alors, en utilisant le fait que $W(x) = \log x - \log\log x + O\left(\frac{\log\log x}{\log x}\right)$ obtenir un équivalent de la complexité de la suite étudiée : $p(n) \sim n \log_2 \log_2 n$.

Conclusion

Nous avons au cours de cette thèse abordé un certain nombre d'aspects de la théorie des motifs évitables, dans lesquels ils est apparu que les L-systèmes tiennent une place privilégiée. D'autres aspects non moins intéressants ont été laissés de côté, par exemple les propriétés topologiques de l'ensemble des mots qui évitent un motif donné, actuellement étudiées par James Currie (voir aussi [5, 57, 58, 59]), la recherche d'algorithmes efficaces pour tester si un mot contient un motif [44, 45, 46], ou encore l'étude des mots maximaux évitant un motif inévitable donné.

Pour finir, voici quelques questions que nous jugeons intéressantes (certaines de ces questions figurent aussi dans [4, 5, 17], avec plusieurs autres).

1. Nous pensons que la conjecture suivante est vraie :

   Conjecture 8.1. *Tout motif k-évitable est k-HDOL-évitable.*

   Comme on l'a vu page 101, elle pourrait permettre de démontrer la décidabilité de la k-évitabilité. Une voie possible pour attaquer cette conjecture serait de montrer d'une part que pour tout motif p il existe un ensemble fini de mots F tel que l'ensemble des mots infiniment biprolongeables évitant $p + F$ soit uniformément récurrent (mais non vide !), et d'autre part qu'un ensemble de ce type est l'ensemble des facteurs d'un HDOL-langage. C'est par exemple ce qui se passe pour $p = AA$ sur $\Sigma = \{a, b, c\}$: si on prend $F = \{aba, cbc\}$, l'ensemble des mots infiniment biprolongeables qui évitent $p + F$ est exactement l'ensemble des facteurs du DOL-langage (Σ, μ, a) [62].

2. L'ensemble des couples (k, k') tel que p est k, k'-HDOL-évitable est-il toujours égal à l'ensemble des couples supérieurs ou égaux (pour l'ordre partiel) à $(\alpha^{\mathrm{DOL}}(p), \alpha^{\mathrm{HDOL}}(p))$, ou peut-il avoir une autre forme ?

3. Préciser la structure de l'algèbre des formules et de son quotient par équivalence. Établir une classification des formules binaires.

4. Trouver d'autres motifs à croissance polynomiale, et leur adapter les calculs faits au chapitre 5 pour les chevauchements.

5. Décrire tous les motifs évités par un L-système donné.

6. Trouver un bon équivalent des motifs verrouillés qui rende compte des facteurs de longueur 3 (ou plus) du motif. Cela permettrait peut-être de se faire une meilleure idée sur l'existence éventuelle de motifs évitables 4-inévitables; en effet, deux points de vue opposés sont possibles actuellement :

   – Les motifs verrouillés sont parmi les plus difficiles à éviter, donc comme ils sont 4-évitables tous les motifs évitables sont 4-évitables. Baker suggère même [4] que les motifs évitables qui ne sont pas divisibles par un motif verrouillé sont tous 3-évitables.

   – Les motifs verrouillés ne sont pas assez difficiles à éviter, il faut concevoir d'autres familles de motifs pour espérer en trouver un 4-inévitable.

Annexe A

Classification partielle des motifs ternaires

Une fois la classification des motifs binaires achevée, nous avons entrepris d'en faire autant avec les motifs ternaires. Silke Nilgens [47] avait déjà montré que presque tous les motifs ternaires évitables sont 3-évitables. En utilisant les techniques développées dans cette thèse, et en particulier l'algorithme présenté au chapitre 6, ainsi que quelques résultats existants, nous avons pu compléter ses résultats et statuer sur la 2-évitabilité d'un grand nombre de motifs. Comme pour les motifs binaires, nous avons aussi essayé de déterminer ceux des motifs ternaires 2-évitables qui sont 2-DOL-inévitables, et ceux des motifs 3-évitables qui sont 3-DOL-inévitables.

Nous présentons ici l'état actuel de cette classification. À permutation des lettres près, il reste à classer : 1 motif pour la 3-évitabilité (de longueur 7), 5 motifs pour la 3-DOL-évitabilité (de longueur 6 et 7), 103 motifs pour la 2-évitabilité (de longueur 6 à 10), et 193 motifs pour la 2-DOL-évitabilité (de longueur 7 à 13), soit en tout 274 motifs (certains sont dans plusieurs listes).

On constate que la longueur des motifs 2-inévitables connus ne dépasse pas 11, celle des motifs 2-DOL-inévitables ne dépasse pas 14, et celle des motifs 3-DOL-inévitables ne dépasse pas 7. Comme aucun motif inconnu n'est de longueur supérieure, ces trois valeurs sont bien les longueurs maximales.

A.1 Les résultats de Silke Nilgens

Pour prouver la 3-évitabilité de certains motifs ternaires, S. Nilgens utilise le morphisme suivant, sur $\Sigma = \{a, b, c\}$:

$$\begin{array}{rrcl} \delta : & \Sigma^* & \longrightarrow & \Sigma^* \\ & a & \longmapsto & ab \\ & b & \longmapsto & bc \\ & c & \longmapsto & ca \end{array} .$$

Le mot infini qu'il engendre, $\delta^\omega(a)$, peut être vu comme une généralisation de la suite de Thue-Morse; comme ce dernier, c'est un mot sans chevauchement. Le fait que le morphisme δ soit uniforme facilite les démonstrations d'évitabilité, puisqu'on peut utiliser des arguments de parité.

S. Nilgens montre que $\delta^\omega(a)$ évite les motifs $ABCABAC$, $ABCBABCA$, $ABACBCA$ et $ABCACBA$. Comme on sait par ailleurs que AA est 3-évitable, elle peut donc conclure que tous les motifs ternaires de longueur supérieure ou égale à 8, qui sont divisibles par l'un des motifs ci-dessus (ou par leur image miroir), sont 3-évitables. La borne ainsi obtenue est optimale, car $ABACABA$, de longueur 7, est inévitable. En fait, la plupart des motifs restant sont 3-inévitables; seuls 5 motifs, à permutation des lettres et image miroir près, échappent à cet argument : $ABACBC$, $ABCACB$, $ABACABC$, $ABACBAB$ et $ABCBABC$.

A.2 Les 5 motifs restants

Tous les motifs dont la 3-évitabilité a été établie par S. Nilgens sont en fait 3-DOL-évitables. Pour les motifs restants, nous allons devoir utiliser des HDOL-langages, car $ABCBABC$ et $ABACBAB$ sont 3-DOL-inévitables et il n'a pas été possible de trouver de DOL-langage pour les trois autres.

Soient $\Sigma_4 = \{a, b, c, d\}$ et $\Sigma_3 = \{a, b, c\}$, et les morphismes suivants :

$$\begin{array}{llcl} f\colon & \Sigma_4^* & \longrightarrow & \Sigma_4^* \quad , \\ & a & \longmapsto & ab \\ & b & \longmapsto & ac \\ & c & \longmapsto & dc \\ & d & \longmapsto & db \end{array}$$

$$\begin{array}{llcl} g_1\colon & \Sigma_4^* & \longrightarrow & \Sigma_3^* \quad , \\ & a & \longmapsto & ab \\ & b & \longmapsto & bc \\ & c & \longmapsto & aa \\ & d & \longmapsto & cb \end{array} \qquad \begin{array}{llcl} g_2\colon & \Sigma_4^* & \longrightarrow & \Sigma_3^* \quad . \\ & a & \longmapsto & ab \\ & b & \longmapsto & ac \\ & c & \longmapsto & bc \\ & d & \longmapsto & cb \end{array}$$

Au moyen de l'algorithme présenté au chapitre 6, on montre que le HDOL-langage $L(\Sigma_4, f, a, \Sigma_3, g_1)$ évite les motifs $ABACBC$ et $ABACABC$, et que le HDOL-langage $L(\Sigma_4, f, a, \Sigma_3, g_2)$ évite les motifs $ABACABC$, $ABACBAB$ et $BCACBA$ (image miroir de $ABCACB$). Quatre des cinq motifs étudiés sont donc 3-HDOL-évitables.

Le dernier motif, $ABCBABC$, pose nettement plus de problèmes. Les résultats du *backtracking* portent à croire qu'il est 3-évitable, et même 2-évitable. Nous pensons qu'il est évité par les HDOL-systèmes $G_1 = (\Sigma_5, f_1, a, \Sigma_3, g_1)$ et

$G_2 = (\Sigma_5, f_2, a, \Sigma_2, g_2)$ où $\Sigma_5 = \{a, b, c, d, e\}$, $\Sigma_3 = \{a, b, c\}$, $\Sigma_2 = \{a, b\}$ et

$$f_1\colon \begin{array}{ccc} \Sigma_5 & \longrightarrow & \Sigma_5 \\ a & \longmapsto & abad \\ b & \longmapsto & cbce \\ c & \longmapsto & bda \\ d & \longmapsto & bcd \\ e & \longmapsto & edbde \end{array} \quad , \qquad g_1\colon \begin{array}{ccc} \Sigma_5 & \longrightarrow & \Sigma_3 \\ a & \longmapsto & aa \\ b & \longmapsto & ccca \\ c & \longmapsto & acbb \\ d & \longmapsto & cab \\ e & \longmapsto & accb \end{array} \quad ,$$

$$f_2\colon \begin{array}{ccc} \Sigma_5 & \longrightarrow & \Sigma_5 \\ a & \longmapsto & ab \\ b & \longmapsto & ac \\ c & \longmapsto & ad \\ d & \longmapsto & ce \\ e & \longmapsto & be \end{array} \quad , \qquad g_2\colon \begin{array}{ccc} \Sigma_5 & \longrightarrow & \Sigma_2 \\ a & \longmapsto & aaab \\ b & \longmapsto & bbba \\ c & \longmapsto & babb \\ d & \longmapsto & abaa \\ e & \longmapsto & baaa \end{array} \quad .$$

Le premier a été obtenu en essayant des morphismes aléatoires; le second, bien plus agréable car uniforme et sur l'alphabet binaire, a été découvert par Kirby Baker. En théorie, il suffirait d'appliquer notre algorithme pour vérifier ces conjectures; mais s'il marche bien avec 2 ou 3 lettres, l'explosion combinatoire fait que les implémentations actuelles n'ont pas de ressources suffisantes pour traiter un alphabet à 5 lettres.

A.3 Résultats de 2-inévitabilité

Par *backtracking*, on arrive à montrer qu'un certain nombre de motifs ternaires sont 2-inévitables. Il peut s'avérer très intéressant de transformer auparavant le motif en une formule équivalente. Ainsi, pour le motif $AABCBABA$, on trouve 224924033 mots qui évitent ce motif, les plus longs étant de longueur 84, par exemple :

$$abababababababaaaaababaaababbaababbaabbbaabbbaaabbbaaabbbaabbaaabbbaabbabbbaaabbbabb$$

(plusieurs jours de calcul sur SPARCstation 10 ont été nécessaires pour terminer le *backtracking*). Par contre, avec la formule équivalente $AAB.BABA$, on ne trouve plus que 719 mots qui l'évitent, de longueur au plus 18, par exemple $abaabbaaabbbaabbab$, et le calcul dure quelques fractions de seconde.

On montre ainsi que les motifs suivants sont 2-inévitables :
$AABABCBAA$, $AABACABB$, $AABACBB$, $AABACBBA$, $AABBCBAA$, $AABBCBBA$, $AABCABBA$, $AABCBABA$, $AABCCAC$, $AABCCB$, $ABAACBAAB$, $ABACABBA$, $ABACCB$, $ABBACAAB$, $ABBCCA$, $ABCBCA$, $ABCCAB$, $ABCCBA$.

A.4 Résultats de DOL-inévitabilité

Les motifs suivants sont 2-DOL-inévitables car simultanément 2-inévitables avec AAA et divisant le multicube C_3 (équivalent à $(A(BCCCD)^3E)^3$, voir page 59) :
$AABAACABAABA$, $AABAACBAAB$, $AABABCBABA$, $AABABCC$, $AABACAABB$, $AABACAABBA$, $AABACABBA$, $AABACBABA$, $AABACBC$, $AABBACAABA$, $AABBACABA$, $AABBACC$, $AABBCAABA$, $AABBCABA$, $AABBCBAB$, $AABBCBC$, $AABBCCB$, $AABCACAB$, $AABCACB$, $AABCBAC$, $AABCBBAB$, $AABCBBC$, $AABCCAB$, $AABCCBA$, $AABCCBC$, $ABAACABAB$, $ABAACABBA$, $ABAACBBA$, $ABAACCAB$, $ABABBCBABA$, $ABABBCBBA$, $ABABBCCB$, $ABABCAAC$, $ABABCAC$, $ABABCBABA$, $ABABCBBA$, $ABABCCB$, $ABACAABBA$, $ABACAACB$, $ABACABC$, $ABACACB$, $ABACACBA$, $ABACBABC$, $ABACBAC$, $ABACBBC$, $ABACBCA$, $ABACCAAB$, $ABACCAB$, $ABACCABA$, $ABBACCA$, $ABBCABC$, $ABBCBAC$, $ABBCBCA$, $ABBCBCAB$, $ABBCCAB$, $ABBCCBA$, $ABCAACB$, $ABCABABC$, $ABCACAB$, $ABCACABC$, $ABCACBA$, $ABCACCAB$, $ABCBBAC$, $ABCBBCBA$.

Ceux-ci sont 2-DOL-inévitables, d'après le théorème 3.4, car contenus dans la suite de Thue-Morse et divisant le multichevauchement O_3.
$AABABCAABCA$, $AABABCACAA$, $AABACABACCA$, $AABACBCA$, $AABACBCC$, $AABACCABAA$, $AABACCABACA$, $AABACCBA$, $AABBCAABC$, $AABBCABBCB$, $AABBCABCA$, $AABBCACA$, $AABCAABABCAAB$, $AABCABAC$, $AABCACABCAC$, $AABCCBAA$, $AABCCBAC$, $AABCCBCA$, $ABAABACABAC$, $ABAABACAC$, $ABAABACBAABAC$, $ABAABACBAC$, $ABAABACCA$, $ABAABCCA$, $ABAACAABAACA$, $ABAACACBAAC$, $ABAACBAACAC$, $ABAACBABAAC$, $ABAACCB$, $ABABBCBC$, $ABABCBABCA$, $ABABCBCA$, $ABACAACABACAAC$, $ABACABAABACABA$, $ABACABBACABA$, $ABBCAABBCAB$, $ABCAACBA$, $ABCABCBA$, $ABCBACAB$.

Enfin, $ABBCABBC$ et $ABACABAC$ sont 2-DOL-inévitables car simultanément 2-inévitables avec AA et contenus dans les grands carrés, et $ABACBAB$ et $ABCBABC$ sont 3-DOL-inévitables pour la même raison (voir 3.5).

A.5 Résultats d'évitabilité

Nous avons trouvé un grand nombre de motifs évités par le mot de Thue-Morse et une vingtaine d'autres DOL-systèmes binaires simples (voir le dictionnaire pour plus de détails). De même, une douzaine de morphismes de $\{a, b, c\}$

dans $\{a, b\}$, tous appliqués au DOL-langage sans carré classique ($f(a) = abc$, $f(b) = ac$, $f(c) = b$) suffisent à éviter 55 motifs.

Tous ces résultats ont été vérifiés au moyen de l'algorithme présenté au chapitre 6.

A.6 Dictionnaire des motifs

Pour finir, voici un « dictionnaire » contenant tous les motifs ternaires, sauf ceux qui sont 2-DOL-évitables car ils contiennent un motif 2-DOL-évitable figurant dans la liste : le théorème 4.1 assure que cette liste est finie. Pour l'utiliser, il faut tout d'abord mettre le motif étudié p sous *forme normale*, c'est-à-dire permuter les variables de sorte que $p \in A^+B\Delta^*$. On recherche ensuite p dans la liste : s'il y figure, on y lit ce qui est connu sur p. Sinon, p est 2-DOL-évitable; pour trouver le DOL-langage qui évite p, on cherche le plus grand préfixe q de p qui figure dans la liste. Si celui-ci est 2-DOL-évitable, on a trouvé un DOL-langage qui évite p. Sinon, soit p' l'image miroir du préfixe de p de longueur $|q| + 1$: la forme normale de p', ou un de ses préfixes 2-DOL-évitable, devrait être dans la liste.

On a indiqué, pour chaque propriété, comment celle-ci pouvait être démontrée : soit en se ramenant à un autre motif (moyennant parfois le passage par des formules équivalentes : ainsi « $AABCABB$ divise $AABB$ » doit être compris comme « $AABCABB$ est équivalent à $AAB.ABB$ qui divise $AABB$ »), soit par *backtracking* (la longueur maximale des mots qui évitent le motif est alors indiquée, si elle est connue), soit en construisant un L-système (Σ, f, a) ou $(\Sigma, f, a, \Sigma', g)$, auquel cas les morphismes f et éventuellement g sont indiqués sous la forme $f = (abc, ac, b)$ si $f(a) = abc$, $f(b) = ac$ et $f(c) = b$, et les alphabets sont sous-entendus. Pour les motifs k-DOL-inévitables, voir A.4.

A
inévitable
AA
2-inévitable *(longueur maximale: 3)*
3-DOL-évitable *($f = (abc, ac, b)$)*
AAA
2-DOL-évitable *($f = (ab, ba)$)*
AAB
2-inévitable *(longueur maximale: 4)*
3-DOL-évitable *(contient AA)*
AABA
2-inévitable *(longueur maximale: 9)*
3-DOL-évitable *(contient AA)*
AABAA
2-inévitable *(longueur maximale: 18)*
3-DOL-évitable *(contient AA)*
AABAAB
2-DOL-évitable *($f = (ab, ba)$)*
AABAAC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACAAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACAABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACAABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACAABAAC
2-DOL-évitable *(contient $AABAAB$)*
AABAACAABAB
2-DOL-évitable *(contient $AABAACBAB$)*
AABAACAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACABAAB
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $AABAACABAABA$)*
3-DOL-évitable *(contient AA)*
AABAACABAABA
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABAACABAAC
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABAACABAB
2-DOL-évitable *(contient $AABAACBAB$)*
AABAACABB
2-DOL-évitable *(contient $AABAACBB$)*
AABAACAC
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABAACACB
2-DOL-évitable *($f = (ab, ba)$)*
AABAACB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACBA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACBAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABAACBAAB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABAACBAABA
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $AABAACABAABA$)*
3-DOL-évitable *(contient AA)*
AABAACBAAC
2-DOL-évitable *($f = (ab, ba)$)*
AABAACBAB
2-DOL-évitable *($f = (ab, ba)$)*
AABAACBAC
2-évitable *(contient $ABBCABC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABAACBACA
2-DOL-évitable *($f = (aabab, baab)$, ou $f = (abaab, babba)$)*
AABAACBACB
2-DOL-évitable *(contient $AABACBCB$)*
AABAACBB
2-DOL-évitable *($f = (ab, ba)$)*
AABAACBC
2-DOL-évitable *($f = (ab, ba)$)*
AABAACC
2-inévitable *(divise $AABB$)*
3-DOL-évitable *(contient AA)*
AABAACCA
2-évitable *(contient $AABBA$)*
2-DOL-inévitable *(divise $AABBA$)*
3-DOL-évitable *(contient AA)*
AABAACCAB
2-DOL-évitable *($f = (ab, ba)$)*
AABAACCAC
2-DOL-évitable *(contient $AABACCAC$)*
AABAACCB
2-DOL-évitable *($f = (ab, ba)$)*
AABAB
2-inévitable *(longueur maximale: 38)*
3-DOL-évitable *(contient AA)*
AABABB
2-DOL-évitable *($f = (aab, bba)$)*
AABABC
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABABCA
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABABCAA
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABABCAAB
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABABCAABA
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABABCAABAB
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABABCAABABB
2-DOL-évitable *(contient $AABACBABB$)*
AABABCAABABC
2-DOL-évitable *(contient $AABAAB$)*
AABABCAABAC
2-DOL-évitable *(contient $ABBCABAC$)*
AABABCAABB
2-DOL-évitable *(contient $AABABCBB$)*
AABABCAABC
2-évitable *(contient $ABBCABC$)*
2-DOL-inévitable *(divise $AABABCAABCA$)*
3-DOL-évitable *(contient AA)*
AABABCAABCA
2-évitable *(contient $ABBCABC$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABABCAABCAB
2-DOL-évitable *(contient $AABABCABCB$)*
AABABCAABCAC
2-DOL-évitable *(contient $ABBCABCAC$)*
AABABCAABCB
2-DOL-évitable *(contient $AABACAACB$)*
AABABCAABCC
2-DOL-évitable *(contient $ABBCABCC$)*
AABABCAAC
2-évitable *(contient $ABACBBC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABABCAACA
2-évitable *(contient $ABACBBC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABABCAACC
2-DOL-évitable *(contient $ABBCAACC$)*

AABABCAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABABCABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABABCABAA
2-évitable *(contient ABABCABAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABABCABAAC
2-DOL-évitable *(contient ABBCBAAC)*
AABABCABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABABCABABB
2-DOL-évitable *(contient AABACBABB)*
AABABCABB
2-DOL-évitable *(contient AABABCBB)*
AABABCABC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise AABABCAABCA)*
3-DOL-évitable *(contient AA)*
AABABCABCB
2-DOL-évitable $(f = (ab, aa))$
AABABCAC
2-évitable *(contient ABABCAC)*
2-DOL-inévitable *(divise AABABCACAA)*
3-DOL-évitable *(contient AA)*
AABABCACA
2-évitable *(contient ABABCAC)*
2-DOL-inévitable *(divise AABABCACAA)*
3-DOL-évitable *(contient AA)*
AABABCACAA
2-évitable *(contient ABABCAC)*
2-DOL-inévitable *(symétrique de AABABCACAA)*
3-DOL-évitable *(contient AA)*
AABABCACAAB
2-DOL-évitable *(contient ABBCACAB)*
AABABCACAB
2-DOL-évitable *(contient AABABCCB)*
AABABCB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABABCBA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABABCBAA
2-inévitable *(équivalent à AABAB.BAA)*
3-DOL-évitable *(contient AA)*
AABABCBAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABABCBABA
2-évitable *(contient ABABCBABA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABABCBABAA
2-DOL-évitable $(f = (aab, abb))$
AABABCBABB
2-DOL-évitable *(contient AABACBABB)*
AABABCBABC
2-DOL-évitable $(f = (ab, ba))$
AABABCBB
2-DOL-évitable $(f = (ab, aa))$
AABABCBC
2-DOL-évitable $(f = (ab, ba))$
AABABCC
2-évitable $(f = (abc, ac, b)$ *et* $g = (aaab, abab, bb))$
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABABCCA
2-évitable *(contient AABABCC)*
2-DOL-inévitable *(divise AABABCACAA)*
3-DOL-évitable *(contient AA)*
AABABCCAA
2-DOL-évitable *(contient ABBCCAA)*
AABABCCAB
2-évitable *(contient AABABCC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABABCCABB
2-DOL-évitable *(contient AABABCBB)*
AABABCCB
2-DOL-évitable $(f = (ab, ba))$
AABAC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACAAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACAABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACAABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACAABAAC
2-DOL-évitable *(contient AABCAABAC)*
AABACAABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABACAABABB
2-DOL-évitable *(contient AABACBABB)*
AABACAABAC
2-DOL-évitable *(contient AABAAB)*
AABACAABB
2-évitable *(contient AABBCABA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABACAABBA
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABACAABBAA
2-DOL-évitable *(contient AABCABBAA)*
AABACAABBAB
2-DOL-évitable *(contient AABACCAC)*
AABACAABBAC
2-DOL-évitable *(contient AABCAABBC)*
AABACAABC
2-DOL-évitable $(f = (ab, aa))$
AABACAAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
AABACAACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
AABACAACAB
2-DOL-évitable $(f = (aab, abb))$
AABACAACB
2-DOL-évitable $(f = (ab, ba))$
AABACAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABACABAAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
AABACABAABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
AABACABAABAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
AABACABAABACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
AABACABAABACAB
2-DOL-évitable *(contient AABAAB)*
AABACABAABACB
2-DOL-évitable *(contient AABACAABC)*
AABACABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABACABABB
2-DOL-évitable *(contient AABACBABB)*
AABACABAC
2-évitable *(contient ABCABC)*

2-DOL-inévitable *(divise $AABACABACCA$)*
3-DOL-évitable *(contient AA)*
$AABACABACB$
2-DOL-évitable *($f = (ab, aa)$)*
$AABACABACC$
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(divise $AABACABACCA$)*
3-DOL-évitable *(contient AA)*
$AABACABACCA$
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
$AABACABACCAB$
2-DOL-évitable *(contient $ABACBACCB$)*
$AABACABB$
2-inévitable *(équivalent à $AABA.ABB$)*
3-DOL-évitable *(contient AA)*
$AABACABBA$
2-évitable *($f = (abc, ac, b)$ et $g = (abbba, abba, a)$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
$AABACABBAA$
2-DOL-évitable *(contient $AABCABBAA$)*
$AABACABBAB$
2-DOL-évitable *(contient $AABACCAC$)*
$AABACABBAC$
2-DOL-évitable *(contient $AABCABBC$)*
$AABACABBC$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACABC$
2-DOL-évitable *($f = (aab, abb)$)*
$AABACAC$
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
$AABACACB$
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACACBA$
2-évitable *($f = (abc, ac, b)$ et $g = (abbb, bbaa, aba)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACACBAC$
2-évitable *(contient $AABACACBA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACACBACA$
2-évitable *(contient $AABACACBA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACACBACAA$
2-DOL-évitable *(contient $AABCCBCAA$)*
$AABACACBACAB$
2-DOL-évitable *(contient $AABCCBCAB$)*
$AABACACBACAC$
2-DOL-évitable *(contient $AABCACBCAC$)*
$AABACACBB$
2-DOL-évitable *($f = (aab, abb)$)*
$AABACACBC$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACACC$
2-DOL-évitable *(symétrique de $AABABCBB$)*
$AABACB$
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
$AABACBA$
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
$AABACBAA$
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
$AABACBAAB$
2-inévitable *(divise $ABBACBBAB$)*
3-DOL-évitable *(contient AA)*
$AABACBAABA$
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABAABA$)*
3-DOL-évitable *(contient AA)*
$AABACBAABAC$
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABAABACBAABAC$)*
3-DOL-évitable *(contient AA)*
$AABACBAABACB$
2-DOL-évitable *(contient $AABAAB$)*
$AABACBAABACC$
2-DOL-évitable *(contient $AABCBABCC$)*
$AABACBAAC$
2-évitable *(contient $ABCABBC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACBAACA$
2-DOL-évitable *($f = (ab, aa)$)*
$AABACBAACC$
2-DOL-évitable *(contient $AABCBACC$)*
$AABACBAB$
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
$AABACBABA$
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
$AABACBABAA$
2-évitable *(contient $AABACBABA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACBABAAC$
2-DOL-évitable *(contient $AABCBABAC$)*
$AABACBABAC$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACBABB$
2-DOL-évitable *($f = (ab, aa)$)*
$AABACBABC$
2-DOL-évitable *($f = (ab, aa)$)*
$AABACBAC$
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(divise $ABAABACBAC$)*
3-DOL-évitable *(contient AA)*
$AABACBACA$
2-DOL-évitable *($f = (aab, abb)$)*
$AABACBACC$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACBB$
2-inévitable *(longueur maximale: 66, équivalent à $AABA.BB$)*
3-DOL-évitable *(contient AA)*
$AABACBBA$
2-inévitable *(équivalent à $AABA.BBA$)*
3-DOL-évitable *(contient AA)*
$AABACBBAA$
2-DOL-évitable *($f = (aaabb, baabb)$, ou $f = (aabbb, aabba)$)*
$AABACBBAB$
2-DOL-évitable *($f = (ab, aa)$)*
$AABACBBAC$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACBBC$
2-DOL-évitable *($f = (aab, abbb)$)*
$AABACBC$
2-évitable *($f = (abc, ac, b)$ et $g = (abbb, bbaa, aba)$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
$AABACBCA$
2-évitable *(contient $AABACBC$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
$AABACBCAA$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACBCAB$
2-DOL-évitable *($f = (aab, abb)$)*
$AABACBCAC$
2-DOL-évitable *($f = (aabab, baab)$)*
$AABACBCB$
2-DOL-évitable *($f = (ab, ba)$)*
$AABACBCC$
2-évitable *(contient $AABACBC$)*
2-DOL-inévitable *(symétrique de $AABACBCC$)*
3-DOL-évitable *(contient AA)*
$AABACBCCB$
2-DOL-évitable *($f = (ab, baa)$)*
$AABACC$
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
$AABACCA$
2-inévitable *(divise $AABCABBA$)*
3-DOL-évitable *(contient AA)*
$AABACCAA$
2-évitable *(contient $ABBAA$)*
2-DOL-inévitable *(divise $ABBAA$)*
3-DOL-évitable *(contient AA)*
$AABACCAAB$
2-évitable *(contient $ABBAA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
$AABACCAABA$
2-évitable *(contient $ABBAA$)*

2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABACCAABAC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABACCAABACC
2-DOL-évitable *(contient AABAAB)*
AABACCAB
2-évitable *($f = (abc, ac, b)$ et $g = (aaa, aba, bbb)$)*
2-DOL-inévitable *(divise AABACCABAA)*
3-DOL-évitable *(contient AA)*
AABACCABA
2-évitable *($f = (abc, ac, b)$ et $g = (aa, aba, abbb)$)*
2-DOL-inévitable *(divise AABACCABAA)*
3-DOL-évitable *(contient AA)*
AABACCABAA
2-évitable *(contient AABACCAB)*
2-DOL-inévitable *(symétrique de AABACCABAA)*
3-DOL-évitable *(contient AA)*
AABACCABAAB
2-DOL-évitable *($f = (abb, baa)$)*
AABACCABAC
2-évitable *(contient AABACCAB)*
2-DOL-inévitable *(divise AABACCABACA)*
3-DOL-évitable *(contient AA)*
AABACCABACA
2-évitable *(contient AABACCAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABACCABACAB
2-DOL-évitable *(contient ABACCBACB)*
AABACCABC
2-DOL-évitable *($f = (ab, ba)$)*
AABACCAC
2-DOL-évitable *($f = (ab, aaa)$)*
AABACCB
2-DOL-inévitable *(divise AABACCBA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABACCBA
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABACCBAA
2-DOL-évitable *($f = (ab, ba)$)*
AABACCBAB
2-évitable *(contient ABACCBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABACCBABB
2-DOL-évitable *(contient AABACBABB)*
AABACCBAC
2-DOL-évitable *($f = (ab, ba)$)*
AABACCBB
2-DOL-évitable *($f = (ab, aa)$)*
AABACCBC
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABACCBCA
2-DOL-évitable *($f = (ab, ba)$)*
AABB
2-inévitable *(longueur maximale: 11)*
3-DOL-évitable *(contient AA)*
AABBA
2-évitable *($f = (abc, ac, b)$ et $g = (aa, aba, abbb)$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABBAA
2-DOL-évitable *($f = (aab, bba)$)*
AABBAB
2-DOL-évitable *($f = (aab, bba)$)*
AABBAC
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACAA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACAAB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACAABA
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABBACAABAC
2-DOL-évitable *($f = (aab, abb)$)*
AABBACAABB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACAABBA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACAABBAA
2-DOL-évitable *(contient AABCABBAA)*
AABBACAABBAB
2-DOL-évitable *(contient AABBACBAB)*
AABBACAABBAC
2-DOL-évitable *(contient AABAAB)*
AABBACAABBC
2-DOL-évitable *(contient AABACAABC)*
AABBACAAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBACAACA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBACAACAB
2-DOL-évitable *(contient AABBACACB)*
AABBACAB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACABA
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABBACABAA
2-DOL-évitable *(contient AABBCABAA)*
AABBACABAC
2-DOL-évitable *($f = (aab, abb)$)*
AABBACABB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACABBA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACABBAA
2-DOL-évitable *(contient AABCABBAA)*
AABBACABBAB
2-DOL-évitable *(contient AABBACBAB)*
AABBACABBAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBACABC
2-DOL-évitable *($f = (ab, baa)$)*
AABBACAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBACACB
2-DOL-évitable *($f = (aab, abb)$)*
AABBACB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACBA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABBACBAA
2-DOL-évitable *($f = (aab, bba)$)*
AABBACBAB
2-DOL-évitable *($f = (ab, aaa)$)*
AABBACBAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBACBACA
2-évitable *(contient AABBA)*

2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABBACBB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*

AABBACBBA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*

AABBACBBAA
2-DOL-évitable *(contient AABBCBBAA)*

AABBACBBAB
2-DOL-évitable *(contient AABBACBAB)*

AABBACBC
2-DOL-évitable $(f = (ab, aaa))$

AABBACC
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBACCA
2-DOL-évitable $(f = (aab, bba))$

AABBACCB
2-DOL-évitable $(f = (aab, bba))$

AABBC
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCA
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCAA
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCAAB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCAABA
2-évitable *(contient AABBCABA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCAABB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCAABBA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*

AABBCAABBAA
2-DOL-évitable *(contient AABCABBAA)*

AABBCAABBAB
2-DOL-évitable *(contient AABACCAC)*

AABBCAABBC
2-DOL-évitable *(contient AABAAB)*

AABBCAABC
2-évitable *(contient ABBCABC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCAABCB
2-DOL-évitable *(contient AABCAACB)*

AABBCAABCC
2-DOL-évitable *(contient ABBCABCC)*

AABBCAAC
2-DOL-évitable $(f = (ab, ba))$

AABBCAB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCABA
2-évitable $(f = (abc, ac, b)$ *et* $g = (abbba, abba, a))$
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCABAA
2-DOL-évitable *(symétrique de AABACBBAA)*

AABBCABAB
2-DOL-évitable $(f = (ab, aa))$

AABBCABB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCABBA
2-DOL-inévitable *(divise AABBA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABBCABBAA
2-DOL-évitable *(contient AABCABBAA)*

AABBCABBAB
2-DOL-évitable *(contient AABACCAC)*

AABBCABBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise AABBCABBCB)*
3-DOL-évitable *(contient AA)*

AABBCABBCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCABBCBC
2-DOL-évitable $(f = (ab, ba))$

AABBCABBCC
2-DOL-évitable $(f = (ab, ba))$

AABBCABC
2-évitable *(contient ABBCABC)*
2-DOL-inévitable *(divise AABBCABCA)*
3-DOL-évitable *(contient AA)*

AABBCABCA
2-évitable *(contient ABBCABC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCABCAA
2-DOL-évitable $(f = (aab, abb))$

AABBCABCB
2-DOL-évitable $(f = (ab, aa))$

AABBCAC
2-DOL-inévitable *(divise AABBCACA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABBCACA
2-évitable *(contient ABBCACA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCACAA
2-DOL-évitable *(contient AABBCCA)*

AABBCACB
2-DOL-évitable $(f = (abb, baa))$

AABBCACC
2-DOL-évitable *(symétrique de AABACCBB)*

AABBCB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCBA
2-inévitable *(divise AABBCBBA)*
3-DOL-évitable *(contient AA)*

AABBCBAA
2-inévitable *(équivalent à AABB.BAA)*
3-DOL-évitable *(contient AA)*

AABBCBAAB
2-DOL-inévitable *(divise ABBAA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABBCBAABB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*

AABBCBAABBC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABBCBAABBCB
2-DOL-évitable *(contient AABAAB)*

AABBCBAABBCC
2-DOL-évitable *(contient ABBCBABBCC)*

AABBCBAAC
2-DOL-évitable $(f = (ab, ba))$

AABBCBAB
2-évitable *(contient ABACBBAA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABBCBABA
2-DOL-évitable $(f = (ab, aa))$

AABBCBABB
2-évitable *(contient ABACBBAA)*
2-DOL-inévitable *(symétrique de AABACAABB)*
3-DOL-évitable *(contient AA)*

AABBCBABBA
2-DOL-évitable *(contient AABCACCA)*

AABBCBABBC
2-DOL-évitable $(f = (aab, abb))$

AABBCBABC
2-DOL-évitable $(f = (ab, aa))$

AABBCBAC
2-DOL-évitable $(f = (aab, bba))$

AABBCBB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABBCBBA
2-inévitable *(longueur maximale: 91, équivalent à AABB.BBA)*
3-DOL-évitable *(contient AA)*

AABBCBBAA
2-DOL-évitable *($f = (aab, abb)$)*
AABBCBBAB
2-DOL-évitable *($f = (aaabb, baabb)$, ou $f = (aabbb, aabba)$)*
AABBCBC
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBCBCA
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBCBCAB
2-DOL-évitable *($f = (ab, aa)$)*
AABBCC
2-DOL-inévitable *(divise $AABBCACA$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABBCCA
2-DOL-évitable *($f = (ab, ba)$)*
AABBCCB
2-évitable *(contient $AABBA$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABBCCBA
2-DOL-évitable *($f = (aab, abb)$)*
AABC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCAAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCAABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCAABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCAABAB
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABCAABABC
2-évitable *(contient $ABCAABABC$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABCAABABCA
2-évitable *(contient $ABCAABABC$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABCAABABCAA
2-évitable *(contient $ABCAABABC$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABCAABABCAAB
2-évitable *(contient $ABCAABABC$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABCAABABCAC
2-DOL-évitable *(contient $ABCABBCAC$)*
AABCAABAC
2-DOL-évitable *($f = (aab, abb)$)*
AABCAABB
2-inévitable *(divise $AABB$)*
3-DOL-évitable *(contient AA)*
AABCAABBA
2-évitable *(contient $AABBA$)*
2-DOL-inévitable *(divise $AABBA$)*
3-DOL-évitable *(contient AA)*
AABCAABBAA
2-DOL-évitable *(contient $AABCABBAA$)*
AABCAABBAB
2-DOL-évitable *(contient $AABACCAC$)*
AABCAABBC
2-DOL-évitable *($f = (ab, ba)$)*
AABCAABC
2-DOL-évitable *(contient $AABAAB$)*
AABCAAC
2-inévitable *(divise $ABBA$)*
3-DOL-évitable *(contient AA)*
AABCAACA
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABAABA$)*
3-DOL-évitable *(contient AA)*
AABCAACAB
2-DOL-évitable *($f = (aab, abb)$)*
AABCAACB
2-DOL-évitable *($f = (aab, abb)$)*
AABCAACC
2-évitable *(contient $ABBAA$)*
2-DOL-inévitable *(divise $ABBAA$)*
3-DOL-évitable *(contient AA)*
AABCAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCABAAB
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $ABAABA$)*
3-DOL-évitable *(contient AA)*
AABCABAABA
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $ABAABA$)*
3-DOL-évitable *(contient AA)*
AABCABAABC
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $ABBABCBABBABCB$)*
3-DOL-évitable *(contient AA)*
AABCABAABCA
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $ABBABCBABBABCB$)*
3-DOL-évitable *(contient AA)*
AABCABAABCAB
2-DOL-évitable *(contient $AABAAB$)*
AABCABAABCB
2-DOL-évitable *(contient $AABACAABC$)*
AABCABAAC
2-DOL-évitable *($f = (abbab, baaba)$, ou $f = (abaab, babba)$)*
AABCABAB
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
AABCABABB
2-DOL-évitable *(contient $AABACACC$)*
AABCABABC
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABCABABCA
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABCABABCAA
2-DOL-évitable *($f = (aab, abb)$)*
AABCABABCAB
2-évitable *(contient $ABAAB$)*
2-DOL-inévitable *(divise $AABCAABABCAAB$)*
3-DOL-évitable *(contient AA)*
AABCABABCABB
2-DOL-évitable *($f = (ab, aa)$)*
AABCABABCB
2-DOL-évitable *(contient $AABACABC$)*
AABCABAC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABCABACB
2-DOL-évitable *($f = (aab, abb)$)*
AABCABB
2-inévitable *(divise $AABB$)*
3-DOL-évitable *(contient AA)*
AABCABBA
2-inévitable *(équivalent à $ABBA.AAB$)*
3-DOL-évitable *(contient AA)*
AABCABBAA
2-DOL-évitable *(symétrique de $AABBACBAA$)*
AABCABBAB
2-DOL-évitable *(contient $AABACCAC$)*
AABCABBAC
2-DOL-évitable *($f = (ab, bbaa)$)*
AABCABBC
2-DOL-évitable *($f = (ab, ba)$)*
AABCABC
2-évitable *(contient $ABCABC$)*

2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
AABCABCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
AABCABCBA
2-DOL-évitable *($f = (aab, abb)$)*
AABCABCBB
2-DOL-évitable *(symétrique de AABACBACC)*
AABCABCBC
2-DOL-évitable *(contient AABABB)*
AABCABCC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
AABCABCCA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise AABACABACCA)*
3-DOL-évitable *(contient AA)*
AABCABCCAA
2-DOL-évitable *(symétrique de AABBCABCAA)*
AABCABCCB
2-DOL-évitable *($f = (ab, ba)$)*
AABCAC
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABCACA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
AABCACAA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
AABCACAAB
2-évitable *($f = (abc, ac, b)$ et $g = (aabbb, baaa, bbabab)$)*
2-DOL-inévitable *(divise ABAACBABAAC)*
3-DOL-évitable *(contient AA)*
AABCACAABA
2-évitable *(contient AABCACAAB)*
2-DOL-inévitable *(divise ABBCABABBCB)*
3-DOL-évitable *(contient AA)*
AABCACAABAB
2-DOL-évitable *(contient ABCACABB)*
AABCACAABAC
2-DOL-évitable *(contient ABCACABAC)*
AABCACAABB
2-DOL-évitable *(contient AABCCABB)*
AABCACAABC
2-évitable *(contient AABCACAAB)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*
AABCACAABCA
2-évitable *(contient AABCACAAB)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*
AABCACAABCAA
2-évitable *(contient AABCACAAB)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*
AABCACAABCAB
2-DOL-évitable *(contient AABCCABCB)*
AABCACAABCAC
2-DOL-évitable *(contient AABAAB)*
AABCACAB
2-évitable *($f = (abc, ac, b)$ et $g = (aaa, aba, bbb)$)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABCACABA
2-DOL-évitable *($f = (aab, abb)$)*
AABCACABC
2-évitable *(contient AABCACAB)*
2-DOL-inévitable *(divise AABCACABCAC)*
3-DOL-évitable *(contient AA)*
AABCACABCA
2-évitable *(contient AABCACAB)*
2-DOL-inévitable *(divise AABCACABCAC)*
3-DOL-évitable *(contient AA)*
AABCACABCAA
2-DOL-évitable *(symétrique de AABCABABCAA)*
AABCACABCAC
2-évitable *(contient AABCACAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABCACABCC
2-DOL-évitable *(contient ABCACBCC)*
AABCACB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCACBA
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCACBAA
2-DOL-évitable *($f = (aab, abb)$)*
AABCACBAB
2-DOL-évitable *($f = (ab, bba)$)*
AABCACBAC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCACBACA
2-évitable *(contient AABCACBAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCACBB
2-DOL-évitable *($f = (aab, bba)$)*
AABCACBC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCACBCA
2-évitable *($f = (abc, ac, b)$ et $g = (aabb, bbaa, baba)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCACBCAA
2-DOL-évitable *($f = (ab, ba)$)*
AABCACBCAC
2-DOL-évitable *($f = (ab, ba)$)*
AABCACC
2-inévitable *(symétrique de AABACBB)*
3-DOL-évitable *(contient AA)*
AABCACCA
2-DOL-évitable *($f = (ab, aaa)$)*
AABCACCB
2-DOL-évitable *($f = (ab, ba)$)*
AABCB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCBA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCBAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
AABCBAAB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
AABCBAABA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
AABCBAABB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
AABCBAABBC
2-DOL-évitable *(contient AABCAACB)*
AABCBAABC
2-évitable *($f = (abc, ac, b)$ et $g = (aabbb, baaa, bbabab)$)*
2-DOL-inévitable *(divise ABAABACBAABAC)*
3-DOL-évitable *(contient AA)*
AABCBAABCB
2-DOL-évitable *(contient AABAAB)*
AABCBAAC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCBAACA
2-DOL-évitable *($f = (ab, ba)$)*
AABCBAACB
2-DOL-évitable *($f = (aab, abb)$)*
AABCBAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
AABCBABA
2-inévitable *(longueur maximale : 84, équivalent à ABAB.BBA)*
3-DOL-évitable *(contient AA)*
AABCBABAA
2-inévitable *(symétrique de AABABCBAA)*
3-DOL-évitable *(contient AA)*

AABCBABAC
2-DOL-évitable *(*$f = (aab, bba)$*)*

AABCBABB
2-inévitable *(symétrique de AABACABB)*
3-DOL-évitable *(contient AA)*

AABCBABBA
2-DOL-évitable *(contient AABCACCA)*

AABCBABBC
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCBABC
2-évitable *(*$f = (abc, ac, b)$ *et*
$g = (aabb, bbaa, baba)$*)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*

AABCBABCA
2-évitable *(contient AABCBABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBABCAA
2-DOL-évitable *(symétrique de AABCACBCAA)*

AABCBABCB
2-évitable *(contient AABCBABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*

AABCBABCBB
2-évitable *(contient AABCBABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*

AABCBABCBBA
2-DOL-évitable *(contient AABCABCBA)*

AABCBABCBBC
2-DOL-évitable *(*$f = (ab, aaa)$*)*

AABCBABCBC
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCBABCC
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCBAC
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBACA
2-DOL-évitable *(*$f = (abaab, babba)$*)*

AABCBACAA
2-DOL-évitable *(symétrique de AABACBCAA)*

AABCBACAB
2-DOL-évitable *(*$f = (abb, aab)$*)*

AABCBACB
2-DOL-évitable *(*$f = (aab, abb)$*)*

AABCBACC
2-DOL-évitable *(symétrique de AABCACBB)*

AABCBB
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABCBBA
2-inévitable *(divise AABACBBA)*
3-DOL-évitable *(contient AA)*

AABCBBAA
2-inévitable *(symétrique de AABBCBAA)*
3-DOL-évitable *(contient AA)*

AABCBBAAB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*

AABCBBAABC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBAABCB
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBAABCBB
2-DOL-évitable *(contient AABAAB)*

AABCBBAB
2-inévitable *(divise AABACBBA)*
3-DOL-évitable *(contient AA)*

AABCBBABA
2-DOL-évitable *(contient AABCCACA)*

AABCBBABB
2-DOL-évitable *(contient AABCCACC)*

AABCBBABC
2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aa, ab, bb)$*)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBABCB
2-évitable *(contient AABCBBABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBABCBC
2-DOL-évitable *(contient AABCCABB)*

AABCBBABCC
2-DOL-évitable *(*$f = (ab, aa)$*)*

AABCBBAC
2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aaab, abab, bb)$*)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBACA
2-DOL-évitable *(*$f = (abaab, babba)$*)*

AABCBBACB
2-évitable *(contient AABCBBAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBACBA
2-DOL-évitable *(*$f = (abaab, babba)$*, contient*
AABCBACA)

AABCBBC
2-évitable *(contient ABAAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

AABCBBCA
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBCAA
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCBBCAB
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBCABA
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCBBCABB
2-DOL-évitable *(contient AABCBACC)*

AABCBBCABC
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCBBCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBACAACA)*
3-DOL-évitable *(contient AA)*

AABCBBCBA
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBBCBAA
2-DOL-évitable *(contient AABBAA)*

AABCBBCBAB
2-DOL-évitable *(contient AABCBCAB)*

AABCBC
2-DOL-inévitable *(divise AABCBCC)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBCA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBCAA
2-DOL-évitable *(contient AABBAA)*

AABCBCAB
2-DOL-évitable *(*$f = (aab, bba)$*)*

AABCBCAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

AABCBCACA
2-DOL-évitable *(*$f = (aab, abb)$*)*

AABCBCACB
2-DOL-évitable *(*$f = (ab, bba)$*)*

AABCBCACC
2-DOL-évitable *(symétrique de AABACACBB)*

AABCBCC
2-évitable *(symétrique de AABABCC)*
2-DOL-inévitable *(symétrique de AABABCC)*
3-DOL-évitable *(contient AA)*

AABCBCCA
2-DOL-évitable *(*$f = (ab, ba)$*)*

AABCC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*

AABCCA
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABCCAA
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*

AABCCAAB
2-DOL-inévitable *(divise ABBCAABBCAB)*

2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCAABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABBCAABBCAB)*
3-DOL-évitable *(contient AA)*
AABCCAABCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABBCAABBCAB)*
3-DOL-évitable *(contient AA)*
AABCCAABCAC
2-DOL-évitable *(contient ABCCABCAC)*
AABCCAABCC
2-DOL-évitable *(contient AABAAB)*
AABCCAAC
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
AABCCAB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCABA
2-DOL-inévitable *(divise AABACCABAA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCABAA
2-DOL-évitable *(symétrique de AABACCBAA)*
AABCCABAB
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCABABB
2-DOL-évitable *(contient AABACACC)*
AABCCABAC
2-DOL-évitable $(f = (aab, abb))$
AABCCABB
2-DOL-évitable $(f = (aaba, babb))$
AABCCABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise AABACCABACA)*
3-DOL-évitable *(contient AA)*
AABCCABCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise AABACCABACA)*
3-DOL-évitable *(contient AA)*
AABCCABCB
2-DOL-évitable $(f = (aab, bba))$
AABCCAC
2-inévitable *(longueur maximale: 104, équivalent à AABA.BB)*
3-DOL-évitable *(contient AA)*
AABCCACA
2-DOL-évitable $(f = (ab, aa))$
AABCCACB
2-DOL-évitable $(f = (ab, aa))$
AABCCACC
2-DOL-évitable *(symétrique de AABAACBB)*
AABCCB
2-inévitable *(longueur maximale: 21)*
3-DOL-évitable *(contient AA)*
AABCCBA
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCBAA
2-évitable $(f = (abc, ac, b)$ *et* $g = (aab, aba, abbb))$
2-DOL-inévitable *(symétrique de AABCCBAA)*
3-DOL-évitable *(contient AA)*
AABCCBAAB
2-DOL-évitable $(f = (aab, abb))$
AABCCBAAC
2-DOL-évitable $(f = (ab, ba))$
AABCCBAB
2-DOL-évitable $(f = (ab, bbaa))$
AABCCBAC
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCBACB
2-DOL-évitable $(f = (aab, abb))$
AABCCBB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de AABBACC)*
3-DOL-évitable *(contient AA)*
AABCCBBA
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
AABCCBBAB
2-DOL-évitable $(f = (ab, ba))$
AABCCBC
2-évitable *(contient ABBAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABCCBCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
AABCCBCAA
2-DOL-évitable *(symétrique de AABCBBCAA)*
AABCCBCAB
2-DOL-évitable $(f = (ab, baa))$
AABCCBCAC
2-DOL-évitable $(f = (ab, ba))$
AB
inévitable
ABA
inévitable
ABAA
2-inévitable *(symétrique de AABA)*
3-DOL-évitable *(contient AA)*
ABAAB
2-évitable $(f = (abc, ac, b)$ *et* $g = (aaa, bbb, ababab))$
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABAA
2-DOL-évitable *(contient ABABA)*
ABAABAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACAA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACAAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACAABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACAABAA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de AABAACABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACAABB
2-DOL-évitable *(contient ABAACAABB)*
ABAABACAABC
2-DOL-évitable *(contient ABABCBAC)*
ABAABACAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACABAA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACABAAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACABAABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAABACABAABAC
2-DOL-évitable *(contient AABAAB)*

ABAABACABAABB
- 2-DOL-évitable *(contient ABAACAABB)*

ABAABACABAABC
- 2-DOL-évitable *(contient ABABCABAC)*

ABAABACABAAC
- 2-DOL-évitable *(contient ABABCABAC)*

ABAABACABAC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABAABACABACB
- 2-DOL-évitable *(contient ABAABCABCB)*

ABAABACABB
- 2-DOL-évitable *(contient ABAABCBB)*

ABAABACAC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABAABACB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABACBA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABACBAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABACBAAB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABACBAABA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABACBAABAC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABAABACBAABB
- 2-DOL-évitable *(contient ABAACAABB)*

ABAABACBAAC
- 2-DOL-évitable *(contient ABABCBAC)*

ABAABACBAB
- 2-DOL-évitable *(contient ABAABCBAB)*

ABAABACBAC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABAABACBACA
- 2-DOL-évitable *(contient ABAABCBCA)*

ABAABACBACC
- 2-DOL-évitable *(contient ABABCBCC)*

ABAABACBB
- 2-DOL-évitable *(contient ABAABCBB)*

ABAABACBC
- 2-DOL-évitable $(f = (ab, ba))$

ABAABACC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABACCA)*
- 3-DOL-évitable *(contient AA)*

ABAABACCA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABAABACCAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABAABACCAAB
- 2-DOL-évitable *(contient ABABCCBA)*

ABAABACCAB
- 2-évitable *(contient ABAAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABAABACCABA
- 2-évitable *(contient ABAAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABAABACCABAA
- 2-DOL-évitable *(contient ABABCCABA)*

ABAABACCABAC
- 2-DOL-évitable *(contient AABBAB)*

ABAABACCABB
- 2-DOL-évitable *(contient ABAABCBB)*

ABAABACCABC
- 2-DOL-évitable *(contient ABABCCAC)*

ABAABB
- 2-DOL-évitable *(symétrique de AABBAB)*

ABAABC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCAAB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCAABA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCAABAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise AABAACABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCAABB
- 2-DOL-évitable *(contient ABAACAABB)*

ABAABCAB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCABA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCABAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCABAAB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCABAABA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCABAABB
- 2-DOL-évitable *(contient ABAACAABB)*

ABAABCABAABC
- 2-DOL-évitable $(f = (ab, ba))$

ABAABCABB
- 2-DOL-évitable *(contient ABAABCBB)*

ABAABCABC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABACABAC)*
- 3-DOL-évitable *(contient AA)*

ABAABCABCB
- 2-DOL-évitable $(f = (ab, aa))$

ABAABCAC
- 2-évitable *(contient ABAAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABAABCACA
- 2-évitable *(contient ABAAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABAABCACAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABAABCB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCBA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCBAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABAABCBAAB
- 2-évitable *(contient ABAAB)*

2-D0L-inévitable *(divise $ABAABA$)*
3-D0L-évitable *(contient AA)*
ABAABCBAABA
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(divise $ABAABA$)*
3-D0L-évitable *(contient AA)*
ABAABCBAABB
2-D0L-évitable *(contient $ABAACAABB$)*
ABAABCBAABC
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(divise $ABAABACBAABAC$)*
3-D0L-évitable *(contient AA)*
ABAABCBAB
2-D0L-évitable *($f = (ab, aa)$)*
ABAABCBAC
2-D0L-évitable *($f = (abb, baa)$)*
ABAABCBB
2-D0L-évitable *(symétrique de $AABACCAC$)*
ABAABCBC
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(divise $ABAABACBAC$)*
3-D0L-évitable *(contient AA)*
ABAABCBCA
2-D0L-évitable *($f = (aab, abb)$)*
ABAABCC
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(symétrique de $AABCCBC$)*
3-D0L-évitable *(contient AA)*
ABAABCCA
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*
ABAABCCB
2-évitable *(contient $ABAAB$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABAABCCBA
2-évitable *(contient $ABAAB$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABAABCCBAB
2-D0L-évitable *(contient $ABAABCBAB$)*
ABAABCCBC
2-D0L-évitable *($f = (aab, abb)$)*
ABAAC
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACAA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACAAB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACAABA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACAABAA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACAABAAC
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(divise $ABAACAABAACA$)*
3-D0L-évitable *(contient AA)*
ABAACAABAACA
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(symétrique de $ABAACAABAACA$)*
3-D0L-évitable *(contient AA)*
ABAACAABAACAA
2-D0L-évitable *(contient $ABABA$)*
ABAACAABAACC
2-D0L-évitable *(contient $ABBCBABBCC$)*
ABAACAABAB
2-évitable *(contient $ABAACABAB$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABAACAABABC
2-D0L-évitable *(contient $ABBCBAAC$)*
ABAACAABB
2-D0L-évitable *(symétrique de $AABBCBBAB$)*
ABAACAB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACABA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACABAA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABAACABAAB
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(divise $ABAABA$)*
3-D0L-évitable *(contient AA)*
ABAACABAABA
2-évitable *(contient $ABAAB$)*
2-D0L-inévitable *(divise $ABAABA$)*
3-D0L-évitable *(contient AA)*
ABAACABAABAC
2-D0L-évitable *(contient $ABACABABC$)*
ABAACABAABB
2-D0L-évitable *(contient $ABAACAABB$)*
ABAACABAAC
2-évitable *(contient $ABCABC$)*
2-D0L-inévitable *(divise $AABCAABABCAAB$)*
3-D0L-évitable *(contient AA)*
ABAACABAACA
2-D0L-évitable *(contient $ABABA$)*
ABAACABAB
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*
ABAACABABB
2-D0L-évitable *(contient $ABAACBABB$)*
ABAACABAC
2-D0L-évitable *($f = (ab, aa)$)*
ABAACABB
2-inévitable *(divise $ABBCABAA$)*
3-D0L-évitable *(contient AA)*
ABAACABBA
2-évitable *(contient $ABBACABAA$)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*
ABAACABBAA
2-évitable *(contient $ABBAA$)*
2-D0L-inévitable *(symétrique de $AABBACAABA$)*
3-D0L-évitable *(contient AA)*
ABAACABBAAB
2-D0L-évitable *(contient $ABACBBAAB$)*
ABAACABBAAC
2-évitable *(contient $ABBAA$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABAACABBAACA
2-évitable *(contient $ABBAA$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABAACABBAACAB
2-évitable *(contient $ABBAA$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABAACABBAACABA
2-D0L-évitable *(contient $ABAACBBAACBA$)*
ABAACABBC
2-D0L-évitable *($f = (ab, baa)$)*
ABAACABC
2-D0L-évitable *($f = (aabb, aaab)$)*
ABAACAC
2-inévitable *(divise $AABAB$)*
3-D0L-évitable *(contient AA)*
ABAACACB
2-évitable *($f = (abc, ac, b)$ et
$g = (aabb, bbaa, baba)$)*
2-D0L-inévitable *(divise $ABAACACBAAC$)*
3-D0L-évitable *(contient AA)*
ABAACACBA
2-évitable *(contient $ABAACACB$)*
2-D0L-inévitable *(divise $ABAACACBAAC$)*
3-D0L-évitable *(contient AA)*
ABAACACBAA
2-évitable *(contient $ABAACACB$)*
2-D0L-inévitable *(divise $ABAACACBAAC$)*
3-D0L-évitable *(contient AA)*
ABAACACBAAC
2-évitable *(contient $ABAACACB$)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*
ABAACACBAACA
2-D0L-évitable *($f = (ab, ba)$)*
ABAACACBAB
2-D0L-évitable *($f = (abb, baa)$)*

ABAACB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABAACBA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABAACBAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABAACBAAB
2-inévitable *(équivalent à AABA.BAAB)*
3-DOL-évitable *(contient AA)*
ABAACBAABA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABAACBAABB
2-DOL-évitable *(contient ABAACAABB)*
ABAACBAAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABAACBAACC)*
3-DOL-évitable *(contient AA)*
ABAACBAACA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABABBCABBCB)*
3-DOL-évitable *(contient AA)*
ABAACBAACAB
2-DOL-évitable *(contient ABACBACAB)*
ABAACBAACAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABAACBAACACB
2-DOL-évitable *(contient ABACBACCB)*
ABAACBAACACC
2-DOL-évitable *(contient ABACBACACC)*
ABAACBAACC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de AABBCABBCB)*
3-DOL-évitable *(contient AA)*
ABAACBAACCA
2-DOL-évitable *($f = (aab, abb)$)*
ABAACBAB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABAACBABA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABAACBABAA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABAACBABAAC
2-évitable *(contient AABCACAAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABAACBABAACA
2-DOL-évitable *(symétrique de ABAACACBAACA)*
ABAACBABAACB
2-DOL-évitable *($f = (ab, ba)$)*
ABAACBABAACC
2-DOL-évitable *(contient ABACBBACC)*
ABAACBABB
2-DOL-évitable *(symétrique de AABACBBAB)*
ABAACBAC
2-évitable *(contient ABBCABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACBACA
2-DOL-évitable *($f = (aabab, baab)$)*
ABAACBACAA
2-DOL-évitable *(symétrique de AABACBAACA)*
ABAACBACB
2-DOL-évitable *($f = (ab, aa)$)*
ABAACBB
2-inévitable *(symétrique de AABCCAC)*
3-DOL-évitable *(contient AA)*
ABAACBBA
2-inévitable *(divise AABCBABB)*
3-DOL-évitable *(contient AA)*
ABAACBBAA
2-évitable *(contient ABACBBAA)*
2-DOL-inévitable *(symétrique de AABBCAABA)*
3-DOL-évitable *(contient AA)*
ABAACBBAAB
2-DOL-évitable *(contient ABACBBAAB)*
ABAACBBAAC
2-évitable *(contient ABACBBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACBBAACB
2-évitable *(contient ABACBBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACBBAACBA
2-DOL-évitable *($f = (aab, abb)$)*
ABAACBBAB
2-DOL-évitable *($f = (ab, aa)$)*
ABAACBBAC
2-DOL-évitable *($f = (ab, aa)$)*
ABAACBBC
2-DOL-évitable *($f = (ab, baa)$)*
ABAACBC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACBCA
2-évitable *($f = (abc, ac, b)$ et $g = (aaa, aba, bbb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACBCAA
2-DOL-évitable *(symétrique de AABCBAACA)*
ABAACBCAC
2-DOL-évitable *($f = (ab, ba)$)*
ABAACBCB
2-DOL-évitable *($f = (aaba, aabb)$)*
ABAACBCC
2-évitable *(symétrique de AABACCBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACBCCA
2-DOL-évitable *($f = (aabab, baab)$, ou $f = (abbab, baaba)$, ou $f = (abaab, babba)$)*
ABAACC
2-inévitable *(divise AABB)*
3-DOL-évitable *(contient AA)*
ABAACCA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
ABAACCAB
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABAACCABA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACCABAA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACCABAAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACCABAACA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACCABAACAB
2-DOL-évitable *(contient ABBCCABBCA)*
ABAACCABAACC
2-DOL-évitable *(contient ABBABB)*
ABAACCABAB
2-DOL-évitable *(contient ABBCCAA)*
ABAACCABB
2-DOL-évitable *(symétrique de AABCCBBAB)*
ABAACCB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABAACCBA
2-DOL-évitable *($f = (ab, ba)$)*
ABAACCBC
2-DOL-évitable *($f = (ab, ba)$)*
ABAB
2-inévitable *(longueur maximale : 18)*
3-DOL-évitable *(contient AA)*
ABABA
2-DOL-évitable *($f = (ab, ba)$)*
ABABB
2-inévitable *(symétrique de AABAB)*
3-DOL-évitable *(contient AA)*
ABABBA
2-DOL-évitable *($f = (aab, bba)$)*

ABABBC
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCAA
2-DOL-évitable *(symétrique de AABCCACA)*
ABABBCAB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCABA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCABAA
2-DOL-évitable *(contient ABABBCAA)*
ABABBCABAB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCABABB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCABABBA
2-DOL-évitable *(contient ABACBABBA)*
ABABBCABABBC
2-DOL-évitable *(contient AABAAB)*
ABABBCABABC
2-DOL-évitable *(contient ABABCABAC)*
ABABBCABB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCABBA
2-DOL-évitable *(contient ABABCABBA)*
ABABBCABBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABABBCABBCB)*
3-DOL-évitable *(contient AA)*
ABABBCABBCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABAACBAACAC)*
3-DOL-évitable *(contient AA)*
ABABBCABBCBC
2-DOL-évitable *(contient AABABB)*
ABABBCABBCC
2-DOL-évitable *(symétrique de AABBCABBCBC)*
ABABBCABC
2-DOL-évitable *($f = (aab, bba)$)*
ABABBCAC
2-DOL-évitable *($f = (ab, bbaa)$)*
ABABBCB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCBA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCBAA
2-DOL-évitable *(contient ABABBCAA)*
ABABBCBAB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCBABA
2-évitable *(contient ABABCBABA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABABBCBABAA
2-DOL-évitable *(contient ABABBCAA)*
ABABBCBABB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCBABBA
2-DOL-évitable *(contient ABACBABBA)*
ABABBCBABBC
2-DOL-évitable *($f = (ab, ba)$)*
ABABBCBB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABABBCBBA
2-inévitable *(divise AABCBABAA)*
3-DOL-évitable *(contient AA)*
ABABBCBBAA
2-DOL-évitable *(contient ABABBCAA)*
ABABBCBBAB
2-évitable *(contient ABABCBBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABABBCBBABA
2-DOL-évitable *($f = (aab, abb)$)*
ABABBCBBABB
2-DOL-évitable *(contient ABACBBABB)*
ABABBCBC
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$, ou $f = (abc, ac, b)$ et $g = (aa, ab, ba)$)*
2-DOL-inévitable *(symétrique de ABABBCBC)*
3-DOL-évitable *(contient AA)*
ABABBCBCA
2-évitable *(contient ABABBCBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABABBCBCAA
2-DOL-évitable *(contient ABABBCAA)*
ABABBCBCAB
2-évitable *(contient ABABBCBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABABBCBCABA
2-DOL-évitable *(contient ABABCCABA)*
ABABBCBCABB
2-DOL-évitable *(contient AABCBCAB)*
ABABBCC
2-DOL-inévitable *(symétrique de AABBCBC)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABABBCCA
2-DOL-évitable *($f = (ab, ba)$)*
ABABBCCB
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABABBCCBA
2-DOL-évitable *($f = (ab, aaa)$)*
ABABC
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCAA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCAAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCAABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCAABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCAABB
2-DOL-évitable *(symétrique de AABBCABAB)*
ABABCAABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise AABABCAABCA)*
3-DOL-évitable *(contient AA)*
ABABCAABCA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise AABABCAABCA)*
3-DOL-évitable *(contient AA)*
ABABCAABCAB
2-DOL-évitable *($f = (ab, aa)$)*
ABABCAABCB
2-DOL-évitable *($f = (ab, ba)$)*
ABABCAAC
2-évitable *(contient ABACBBC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABABCAACA
2-évitable *(contient ABACBBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABABCAACAB
2-DOL-évitable *(contient ABABCACB)*
ABABCAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABABCABAA
2-évitable *(symétrique de AABACBABA)*
2-DOL-inévitable *(symétrique de AABACBABA)*
3-DOL-évitable *(contient AA)*
ABABCABAAB
2-DOL-évitable *(contient ABABCBAAB)*

ABABCABAB
- 2-inévitable *(divise AABAB)*
- 3-DOL-évitable *(contient AA)*

ABABCABABB
- 2-inévitable *(divise ABABB)*
- 3-DOL-évitable *(contient AA)*

ABABCABABBA
- 2-DOL-évitable *(contient ABACBABBA)*

ABABCABABBC
- 2-DOL-évitable *(contient ABACABABC)*

ABABCABABC
- 2-DOL-évitable *(contient AABAAB)*

ABABCABAC
- 2-DOL-évitable *(*$f = (ab, bba)$*)*

ABABCABB
- 2-inévitable *(divise ABABB)*
- 3-DOL-évitable *(contient AA)*

ABABCABBA
- 2-DOL-évitable *(*$f = (ab, aa)$*)*

ABABCABBC
- 2-DOL-évitable *(*$f = (ab, aa)$*)*

ABABCABC
- 2-évitable *(contient ABCABC)*
- 2-DOL-inévitable *(divise ABABBCABBCB)*
- 3-DOL-évitable *(contient AA)*

ABABCABCB
- 2-évitable *(contient ABCABC)*
- 2-DOL-inévitable *(divise ABABBCABBCB)*
- 3-DOL-évitable *(contient AA)*

ABABCABCBC
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABABCABCC
- 2-DOL-évitable *(contient AABABB)*

ABABCAC
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aab, aba, abbb)$*)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABABCACA
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aa, ab, bb)$*)*
- 2-DOL-inévitable *(divise AABABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABABCACAA
- 2-évitable *(contient ABABCAC)*
- 2-DOL-inévitable *(divise AABABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABABCACAAB
- 2-évitable *(contient ABABCAC)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABABCACAABA
- 2-DOL-évitable *(contient ABABCCABA)*

ABABCACAABC
- 2-DOL-évitable *(contient AABCBCAB)*

ABABCACB
- 2-DOL-évitable *(*$f = (abb, baa)$*)*

ABABCACC
- 2-DOL-évitable *(symétrique de AABACBCB)*

ABABCB
- 2-inévitable *(divise AABAB)*
- 3-DOL-évitable *(contient AA)*

ABABCBA
- 2-inévitable *(divise AABAB)*
- 3-DOL-évitable *(contient AA)*

ABABCBAA
- 2-inévitable *(symétrique de AABCBABA)*
- 3-DOL-évitable *(contient AA)*

ABABCBAAB
- 2-DOL-évitable *(*$f = (ab, aa)$*)*

ABABCBAAC
- 2-DOL-évitable *(*$f = (aab, bba)$*)*

ABABCBAB
- 2-inévitable *(divise AABAB)*
- 3-DOL-évitable *(contient AA)*

ABABCBABA
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aa, ab, bb)$*)*
- 2-DOL-inévitable *(symétrique de ABABCBABA)*
- 3-DOL-évitable *(contient AA)*

ABABCBABAA
- 2-évitable *(contient ABABCBABA)*
- 2-DOL-inévitable *(symétrique de AABABCBABA)*
- 3-DOL-évitable *(contient AA)*

ABABCBABAAB
- 2-DOL-évitable *(contient ABABCBAAB)*

ABABCBABB
- 2-inévitable *(divise ABABB)*
- 3-DOL-évitable *(contient AA)*

ABABCBABBA
- 2-DOL-évitable *(contient ABACBABBA)*

ABABCBABC
- 2-évitable *(contient ABCABC)*
- 2-DOL-inévitable *(divise ABAABACBAABAC)*
- 3-DOL-évitable *(contient AA)*

ABABCBABCA
- 2-évitable *(contient ABCABC)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABABCBABCAB
- 2-DOL-évitable *(contient ABACBACAB)*

ABABCBABCC
- 2-DOL-évitable *(symétrique de AABCBABCBC)*

ABABCBAC
- 2-DOL-évitable *(*$f = (aab, bba)$*)*

ABABCBB
- 2-inévitable *(divise ABABB)*
- 3-DOL-évitable *(contient AA)*

ABABCBBA
- 2-inévitable *(divise AABCBABAA)*
- 3-DOL-évitable *(contient AA)*

ABABCBBAA
- 2-DOL-évitable *(symétrique de AABBCBABA)*

ABABCBBAB
- 2-évitable *(symétrique de ABAACABAB)*
- 2-DOL-inévitable *(symétrique de ABAACABAB)*
- 3-DOL-évitable *(contient AA)*

ABABCBBABA
- 2-évitable *(contient ABABCBBAB)*
- 2-DOL-inévitable *(symétrique de ABABBCBABA)*
- 3-DOL-évitable *(contient AA)*

ABABCBBABB
- 2-DOL-évitable *(contient ABACBBABB)*

ABABCBBABC
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABABCBBAC
- 2-DOL-évitable *(*$f = (ab, aa)$*)*

ABABCBBC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABABCBBCB)*
- 3-DOL-évitable *(contient AA)*

ABABCBBCA
- 2-DOL-évitable *(*$f = (ab, bba)$*)*

ABABCBBCB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(symétrique de ABAABACAC)*
- 3-DOL-évitable *(contient AA)*

ABABCBBCBA
- 2-DOL-évitable *(contient ABACBCBA)*

ABABCBC
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aaab, abab, bb)$*)*
- 2-DOL-inévitable *(divise ABAABACBAC)*
- 3-DOL-évitable *(contient AA)*

ABABCBCA
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aa, ab, bb)$*)*
- 2-DOL-inévitable
- 3-DOL-évitable *(contient AA)*

ABABCBCAA
- 2-DOL-évitable *(symétrique de AABCBCACA)*

ABABCBCAB
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABABCBCAC
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABABCBCC
- 2-DOL-évitable *(symétrique de AABABCBC)*

ABABCC
- 2-DOL-inévitable *(divise AABABCACAA)*
- 2-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABABCCA
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aa, b, ba)$*)*
- 2-DOL-inévitable *(divise AABABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABABCCAA
- 2-évitable *(contient ABABCCA)*
- 2-DOL-inévitable *(symétrique de AABBCACA)*
- 3-DOL-évitable *(contient AA)*

ABABCCAB
- 2-évitable *(contient ABABCCA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABABCCABA
- 2-DOL-évitable *(*$f = (aab, bba)$*)*

ABABCCABB
- 2-évitable *(contient ABABCCA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABABCCABC
- 2-DOL-évitable *(contient AABBAB)*

ABABCCAC
2-DOL-évitable *(symétrique de ABAACBCB)*
ABABCCB
2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aa, ab, bb)$*)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABABCCBA
2-DOL-évitable *(*$f = (aab, bba)$*)*
ABABCCBB
2-évitable *(contient ABABCCB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABABCCBBA
2-DOL-évitable *(*$f = (ab, ba)$*)*
ABABCCBC
2-évitable *(contient ABABCCB)*
2-DOL-inévitable *(divise ABCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABAC
inévitable
ABACA
inévitable
ABACAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABACAAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABACAABA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABACAABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABACAABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACAABAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACAABACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACAABACAA
2-DOL-évitable *(contient ABABA)*
ABACAABACC
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACAABACCA
2-DOL-évitable *(contient ABCAABCCA)*
ABACAABB
2-évitable *(contient AABBCABA)*
2-DOL-inévitable *(symétrique de AABBCBAB)*
3-DOL-évitable *(contient AA)*
ABACAABBA
2-évitable *(contient AABBA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABACAABBAB
2-DOL-évitable *(contient ABACABBAB)*
ABACAABBAC
2-DOL-évitable *(*$f = (aab, abb)$*)*
ABACAABC
2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aaab, abab, bb)$*)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACAABCA
2-DOL-évitable *(*$f = (aab, abb)$*)*
ABACAABCB
2-DOL-évitable *(*$f = (abbab, baaba)$*, ou* $f = (abaab, babba)$*)*
ABACAAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABACAACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABACAACAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACAACABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACAACABAA
2-DOL-évitable *(contient ABACACBAA)*
ABACAACABAB
2-DOL-évitable *(contient ABCACABAB)*
ABACAACABAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACAACABACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACAACABACAA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABACAACABACAAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABACAACABACAACA
2-DOL-évitable *(contient ABABA)*
ABACAACABACAC
2-DOL-évitable *(contient ABAACABAC)*
ABACAACABACC
2-DOL-évitable *(contient ABAACABC)*
ABACAACABB
2-DOL-évitable *(contient ABCACABB)*
ABACAACB
2-évitable *(contient ABAAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABACAACBA
2-DOL-évitable *(*$f = (aab, abb)$*)*
ABACAACBB
2-DOL-évitable *(symétrique de AABCCBCAC)*
ABACAB
inévitable
ABACABA
inévitable
ABACABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABACABAAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABACABAABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABACABAABAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACABAABACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACABAABACAA
2-DOL-évitable *(contient ABCBAABCA)*
ABACABAABACAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACABAABACABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABACABAABACABAA
2-DOL-évitable *(contient ABABA)*
ABACABAABACC
2-DOL-évitable *(contient ABAABB)*
ABACABAABB
2-DOL-évitable *(contient ABACBAABB)*
ABACABAAC
2-DOL-évitable *(*$f = (aab, abb)$*)*
ABACABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACABABB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABACABABBA
2-DOL-évitable *(contient ABACBABBA)*

ABACABABC
2-DOL-évitable *($f = (aab, bba)$)*
ABACABAC
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(cf. A.4)*
3-DOL-évitable *(contient AA)*
ABACABACA
2-DOL-évitable *(contient $ABABA$)*
ABACABACB
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(divise $ABABCBABCA$)*
3-DOL-évitable *(contient AA)*
ABACABACBB
2-DOL-évitable *(symétrique de $AABCACBCAC$)*
ABACABACC
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(divise $AABACABACCA$)*
3-DOL-évitable *(contient AA)*
ABACABACCA
2-évitable *(contient $ABCABC$)*
2-DOL-inévitable *(divise $AABACABACCA$)*
3-DOL-évitable *(contient AA)*
ABACABACCAC
2-DOL-évitable *(contient $ABCABCCAC$)*
ABACABB
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
ABACABBA
2-inévitable *(équivalent à $ABBA.ABA$)*
3-DOL-évitable *(contient AA)*
ABACABBAA
2-évitable *(contient $ABBAA$)*
2-DOL-inévitable *(symétrique de $AABBACABA$)*
3-DOL-évitable *(contient AA)*
ABACABBAAB
2-DOL-évitable *(contient $ABACBBAAB$)*
ABACABBAB
2-DOL-évitable *(symétrique de $ABAABCBAB$)*
ABACABBAC
2-évitable *(contient $ABCABBC$)*
2-DOL-inévitable *(divise $ABACABBACABA$)*
3-DOL-évitable *(contient AA)*
ABACABBACA
2-évitable *(contient $ABCABBC$)*
2-DOL-inévitable *(divise $ABACABBACABA$)*
3-DOL-évitable *(contient AA)*
ABACABBACAA
2-évitable *(contient $ABCABBC$)*
2-DOL-inévitable *(symétrique de $AABACCABACA$)*
3-DOL-évitable *(contient AA)*
ABACABBACAAC
2-DOL-évitable *(contient $ABCABBCAC$)*
ABACABBACAB
2-évitable *(contient $ABCABBC$)*
2-DOL-inévitable *(divise $ABACABBACABA$)*
3-DOL-évitable *(contient AA)*
ABACABBACABA
2-évitable *(contient $ABCABBC$)*
2-DOL-inévitable *(symétrique de $ABACABBACABA$)*
3-DOL-évitable *(contient AA)*
ABACABBACABAC
2-DOL-évitable *(contient $ABCABBCABC$)*
ABACABBC
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACABBCA
2-DOL-évitable *($f = (abbab, baaba)$, ou $f = (abaab, babba)$)*
ABACABBCB
2-DOL-évitable *(symétrique de $ABAACBCAC$)*
ABACABC
2-DOL-inévitable
2-évitabilité inconnue
3-évitable *($f = (ab, ac, dc, db)$ et $g = (ab, ac, bc, cb)$, ou $f = (ab, ac, dc, db)$ et $g = (ab, bc, aa, cb)$)*
3-DOL-évitabilité inconnue
4-DOL-évitable *($f = (ab, ac, dc, db)$)*
ABACABCA
2-DOL-évitable *($f = (aab, abb)$)*
ABACABCB
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient $ABCBACA$)*
ABACABCBA
2-DOL-évitable *($f = (aab, abb)$)*
ABACABCBB
2-DOL-évitable *(symétrique de $AABACBCAC$)*
ABACABCBBC
2-DOL-évitable *($f = (ab, ba)$)*
ABACABCC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACAC
2-inévitable *(divise $AABAB$)*
3-DOL-évitable *(contient AA)*
ABACACB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACACBA
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACACBAA
2-DOL-évitable *(symétrique de $AABCACABA$)*
ABACACBAC
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABACABAABACABA$)*
3-DOL-évitable *(contient AA)*
ABACACBACA
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABACAACABACAAC$)*
3-DOL-évitable *(contient AA)*
ABACACBACC
2-DOL-évitable *(contient $ABCACBCC$)*
ABACACBB
2-évitable *(contient $ABBAA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACACBBC
2-DOL-évitable *($f = (ab, aa)$)*
ABACACBC
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACACBCA
2-DOL-évitable *($f = (abb, baa)$)*
ABACACBCB
2-DOL-évitable *(symétrique de $ABABCBCAC$)*
ABACACBCC
2-DOL-évitable *(symétrique de $AABACACBC$)*
ABACACC
2-inévitable *(divise $ABABB$)*
3-DOL-évitable *(contient AA)*
ABACACCB
2-DOL-évitable *($f = (ab, ba)$)*
ABACB
inévitable
ABACBA
inévitable
ABACBAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABACBAAB
2-inévitable *(divise $ABAACBAAB$)*
3-DOL-évitable *(contient AA)*
ABACBAABA
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABAABA$)*
3-DOL-évitable *(contient AA)*
ABACBAABAC
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABACABAABACABA$)*
3-DOL-évitable *(contient AA)*
ABACBAABACA
2-DOL-évitable *(contient $ABCBAABCA$)*
ABACBAABACB
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABACABAABACABA$)*
3-DOL-évitable *(contient AA)*
ABACBAABACBA
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABACABAABACABA$)*
3-DOL-évitable *(contient AA)*
ABACBAABACBAA
2-DOL-évitable *(contient $ABABA$)*
ABACBAABB
2-DOL-évitable *(symétrique de $AABBACBAB$)*
ABACBAABC
2-DOL-évitable *($f = (aab, abb)$)*

ABACBAAC
2-évitable *(contient ABCABBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACBAACA
2-DOL-évitable *(symétrique de ABAACBACA)*
ABACBAACAA
2-DOL-évitable *(symétrique de AABAACBACA)*
ABACBAACAB
2-DOL-évitable $(f = (abb, aab))$
ABACBAACB
2-DOL-évitable $(f = (aab, abb))$
ABACBAACC
2-DOL-évitable *(symétrique de AABBCABCB)*
ABACBAB
2-inévitable *(divise AABAB)*
3-évitable $(f = (ab, ac, dc, db)$ *et* $g = (ab, ac, bc, cb))$
3-DOL-inévitable *(cf. A.4)*
4-DOL-évitable $(f = (ab, ac, dc, db))$
ABACBABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACBABAA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABACBABAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAACBABAAC)*
3-DOL-évitable *(contient AA)*
ABACBABACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
ABACBABACAA
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACBABACB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
ABACBABACBA
2-DOL-évitable *(contient ABABA)*
ABACBABACBB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de AABCACABCAC)*
3-DOL-évitable *(contient AA)*
ABACBABB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABACBABBA
2-DOL-évitable $(f = (ab, aa))$
ABACBABBC
2-DOL-évitable $(f = (abb, baa))$
ABACBABC
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient ABCABAC)*
ABACBABCA
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABCABAC)*
ABACBABCAA
2-évitable *(contient ABCABACBB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient ABCABAC)*
ABACBAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABACBACA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABACBACAA
2-DOL-évitable *(symétrique de AABACBACA)*
ABACBACAB
2-DOL-évitable $(f = (aab, abb))$
ABACBACAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABAACBAACAC)*
3-DOL-évitable *(contient AA)*
ABACBACACC
2-DOL-évitable *(symétrique de AABABCABCB)*
ABACBACC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABAACBAACC)*
3-DOL-évitable *(contient AA)*
ABACBACCA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise AABACABACCA)*
3-DOL-évitable *(contient AA)*
ABACBACCAA
2-évitable *(contient ABCABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACBACCAC
2-DOL-évitable *(symétrique de ABAABCABCB)*
ABACBACCB
2-DOL-évitable $(f = (ab, aa))$
ABACBB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACBBA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACBBAA
2-évitable *(symétrique de AABBCABA)*
2-DOL-inévitable *(symétrique de AABBCABA)*
3-DOL-évitable *(contient AA)*
ABACBBAAB
2-DOL-évitable $(f = (ab, aaa))$
ABACBBAAC
2-DOL-évitable $(f = (ab, aa))$
ABACBBAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACBBABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABACBBABAC
2-DOL-évitable $(f = (ab, ba))$
ABACBBABB
2-DOL-évitable *(symétrique de AABAACBAB)*
ABACBBABC
2-DOL-évitable $(f = (aab, bba))$
ABACBBAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAACBABAAC)*
3-DOL-évitable *(contient AA)*
ABACBBACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACAA)*
3-DOL-évitable *(contient AA)*
ABACBBACB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABACBBACBA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABACBBACBAB
2-DOL-évitable *(contient ABACBACAB)*
ABACBBACBC
2-DOL-évitable $(f = (ab, aaa))$
ABACBBACC
2-DOL-évitable *(symétrique de AABCCABCB)*
ABACBBC
2-évitable $(f = (abc, ac, b)$ *et* $g = (aabb, bbaa, baba))$
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABACBBCA
2-DOL-évitable $(f = (aab, abb))$
ABACBBCB
2-évitable *(contient ABACBBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACBBCBA
2-évitable *(contient ABACBBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACBBCBAA
2-DOL-évitable *(symétrique de AABCBBCABA, contient ABACBCAA)*
ABACBBCBAC
2-DOL-évitable $(f = (ab, baa))$
ABACBBCC
2-DOL-évitable *(symétrique de AABBACBC)*
ABACBC
2-DOL-inévitable *(divise ABAABA)*
2-évitabilité inconnue
3-évitable $(f = (ab, ac, dc, db)$ *et*

$g = (ab, bc, aa, cb)$)
3-DOL-évitabilité inconnue
4-DOL-évitable *($f = (ab, ac, dc, db)$)*

ABACBCA
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *($f = (ab, bc, ca)$)*

ABACBCAA
2-DOL-évitable *(symétrique de AABCBACA)*

ABACBCAAC
2-DOL-évitable *($f = (ab, aa)$)*

ABACBCAB
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABACBCA)*

ABACBCABA
2-DOL-évitable *($f = (aab, abb)$)*

ABACBCABB
2-DOL-évitable *(symétrique de AABCACBAB)*

ABACBCABC
2-DOL-évitable *($f = (aab, abb)$)*

ABACBCAC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABACBCA)*

ABACBCACB
2-DOL-évitable *($f = (ab, baa)$)*

ABACBCB
2-évitable *(symétrique de ABABCAC)*
2-DOL-inévitable *(symétrique de ABABCAC)*
3-DOL-évitable *(contient AA)*

ABACBCBA
2-DOL-évitable *($f = (ab, aa)$)*

ABACBCBB
2-évitable *(contient ABACBCB)*
2-DOL-inévitable *(divise AABABCACAA)*
3-DOL-évitable *(contient AA)*

ABACBCBBA
2-évitable *(contient ABACBCB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACBCBBAB
2-DOL-évitable *(symétrique de ABAACACBAB)*

ABACBCBBAC
2-DOL-évitable *($f = (ab, aaa)$)*

ABACBCC
2-évitable *(symétrique de AABACBC)*
2-DOL-inévitable *(symétrique de AABACBC)*
3-DOL-évitable *(contient AA)*

ABACBCCA
2-DOL-évitable *($f = (ab, ba)$)*

ABACBCCB
2-évitable *(contient ABACBCC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACBCCBA
2-DOL-évitable *($f = (aab, abb)$)*

ABACBCCBC
2-DOL-évitable *(symétrique de ABAABACBC)*

ABACC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*

ABACCA
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*

ABACCAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*

ABACCAAB
2-évitable *(contient ABBAA)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*

ABACCAABA
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCAABAA
2-DOL-évitable *(contient ABCCAABA)*

ABACCAABAC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCAABACC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCAABACCA
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCAABACCAA
2-DOL-évitable *(contient ABABA)*

ABACCAB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCABA
2-DOL-inévitable *(symétrique de ABACCABA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCABAA
2-évitable *(symétrique de AABACCABA)*
2-DOL-inévitable *(divise ABACABBACAA)*
3-DOL-évitable *(contient AA)*

ABACCABAAB
2-évitable *(contient ABCCBABB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCABAABA
2-évitable *(contient ABCCBABB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCABAABAC
2-DOL-évitable *(contient ABBAAB)*

ABACCABAABB
2-DOL-évitable *(contient ABACBAABB)*

ABACCABAAC
2-DOL-évitable *(contient ABCCABAC)*

ABACCABAC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*

ABACCABACA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*

ABACCABACC
2-DOL-évitable *(contient ABBABB)*

ABACCABB
2-DOL-évitable *(symétrique de AABCCBAB)*

ABACCABC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCABCA
2-DOL-évitable *($f = (ab, aa)$)*

ABACCAC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*

ABACCACB
2-DOL-évitable *($f = (ab, ba)$)*

ABACCB
2-inévitable *(longueur maximale : 63)*
3-DOL-évitable *(contient AA)*

ABACCBA
2-DOL-inévitable *(divise AABACCABAA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCBAA
2-DOL-inévitable *(divise AABACCABAA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCBAAC
2-DOL-évitable *($f = (aab, abb)$)*

ABACCBAB
2-évitable *($f = (abc, ac, b)$ et $g = (aa, ab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCBABA
2-DOL-évitable *(symétrique de ABABCCABA)*

ABACCBABB
2-évitable *(contient ABACCBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABACCBABBA
2-DOL-évitable *(contient ABACBABBA)*

ABACCBABBC
2-DOL-évitable *($f = (ab, aaa)$)*

ABACCBABC
2-DOL-évitable *($f = (aab, abb)$)*

ABACCBAC
2-évitable *(contient ABBAB)*

2-DOL-inévitable *(divise AABACCABACA)*
3-DOL-évitable *(contient AA)*
ABACCBACA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise AABACCABACA)*
3-DOL-évitable *(contient AA)*
ABACCBACAC
2-DOL-évitable *(symétrique de ABABCAABCB)*
ABACCBACB
2-DOL-évitable $(f = (ab, aaa))$
ABACCBB
2-DOL-inévitable *(divise ABABCCAA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACCBBA
2-DOL-évitable $(f = (ab, ba))$
ABACCBBC
2-DOL-évitable $(f = (abb, aaab))$
ABACCBC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACCBCA
2-évitable $(f = (abc, ac, b)$ *et* $g = (aa, ab, bb))$
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABACCBCAA
2-DOL-évitable *(symétrique de AABCBBACA)*
ABACCBCAC
2-DOL-évitable *(symétrique de ABACAABCB)*
ABACCBCB
2-DOL-évitable *(symétrique de ABABBCAC)*
ABACCBCC
2-DOL-évitable *(symétrique de AABAACBC)*
ABB
2-inévitable *(symétrique de AAB)*
3-DOL-évitable *(contient AA)*
ABBA
2-inévitable *(longueur maximale: 10)*
3-DOL-évitable *(contient AA)*
ABBAA
2-évitable *(symétrique de AABBA)*
2-DOL-inévitable *(symétrique de AABBA)*
3-DOL-évitable *(contient AA)*
ABBAAB
2-DOL-évitable $(f = (aab, bba))$
ABBAAC
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACAAB
2-DOL-évitable $(f = (abb, baa))$
ABBAACAB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACABA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de ABACAABBA)*
3-DOL-évitable *(contient AA)*
ABBAACABAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de AABACAABBA)*
3-DOL-évitable *(contient AA)*
ABBAACABAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBAACABAB
2-DOL-évitable *(contient ABBACABAB)*
ABBAACABAC
2-DOL-évitable *(contient ABBACABC)*
ABBAACABB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACABBA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACABBAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACABBAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBAACABBAAC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBAACABBAC
2-DOL-évitable *(contient ABAACABAC)*
ABBAACAC
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de ABABBCCB)*
3-DOL-évitable *(contient AA)*
ABBAACACB
2-DOL-évitable $(f = (ab, aaa))$
ABBAACB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACBA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACBAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACBAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBAACBAAC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBAACBAACA
2-DOL-évitable *(symétrique de ABAACBAACCA)*
ABBAACBAB
2-DOL-évitable *(symétrique de ABACBBAAB)*
ABBAACBAC
2-DOL-évitable $(f = (aab, abb))$
ABBAACBB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACBBA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACBBAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBAACBBAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBAACBBAB
2-DOL-évitable *(contient ABBAACBAB)*
ABBAACBBAC
2-DOL-évitable *(contient ABBACBBC)*
ABBAACBC
2-DOL-évitable *(symétrique de ABACCBBC)*
ABBAACC
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de AABBCCB)*
3-DOL-évitable *(contient AA)*
ABBAACCA
2-DOL-évitable $(f = (aab, abb))$
ABBAB
2-évitable *(symétrique de ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABA
2-DOL-évitable *(symétrique de ABABBA)*
ABBABB
2-DOL-évitable *(symétrique de AABAAB)*
ABBABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCAA
2-DOL-évitable *(symétrique de AABCACCA)*
ABBABCAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*

ABBABCABA
2-DOL-évitable *(symétrique de ABACBABBA)*
ABBABCABB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCABBA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCABBAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCABBABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABACBAABAC)*
3-DOL-évitable *(contient AA)*
ABBABCABBC
2-DOL-évitable *($f = (ab, baa)$, contient ABBACABC)*
ABBABCABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABACBAC)*
3-DOL-évitable *(contient AA)*
ABBABCAC
2-évitable *(contient ABBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBABCACB
2-DOL-évitable *($f = (abaab, babba)$)*
ABBABCACBC
2-DOL-évitable *(symétrique de ABACABCBBC)*
ABBABCACC
2-DOL-évitable *(symétrique de AABACBCCB)*
ABBABCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBAA
2-DOL-évitable *(contient ABBABCAA)*
ABBABCBAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBABA
2-DOL-évitable *(contient ABBACBABA)*
ABBABCBABB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBABBA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBABBAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBABBABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
ABBABCBABBABCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(symétrique de ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABBABCBABBC
2-DOL-évitable *(contient ABBACBABC)*
ABBABCBABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABACABAC)*
3-DOL-évitable *(contient AA)*
ABBABCBABCC
2-DOL-évitable *(symétrique de AABCBABCBBC)*
ABBABCBB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBBA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBBAA
2-DOL-évitable *(contient ABBABCAA)*
ABBABCBBAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBBABA
2-DOL-évitable *(contient ABBACBABA)*
ABBABCBBABB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise AABAACABAABA)*
3-DOL-évitable *(contient AA)*
ABBABCBBC
2-DOL-évitable *($f = (aab, abb)$)*
ABBABCBC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABACAC)*
3-DOL-évitable *(contient AA)*
ABBABCBCA
2-DOL-évitable *($f = (ab, aaa)$)*
ABBABCC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(symétrique de AABCBBC)*
3-DOL-évitable *(contient AA)*
ABBABCCA
2-DOL-évitable *($f = (ab, ba)$)*
ABBABCCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABACCA)*
3-DOL-évitable *(contient AA)*
ABBABCCBA
2-évitable *(contient ABBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBABCCBAB
2-évitable *(contient ABBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBABCCBABA
2-DOL-évitable *(contient ABBACBABA)*
ABBABCCBABB
2-DOL-évitable *(symétrique de AABACCABAAB)*
ABBABCCBABC
2-DOL-évitable *($f = (ab, aaa)$)*
ABBABCCBB
2-évitable *(contient ABBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBABCCBBA
2-DOL-évitable *($f = (ab, aaa)$)*
ABBAC
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACA
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACAA
2-inévitable *(divise AABCABBA)*
3-DOL-évitable *(contient AA)*
ABBACAAB
2-inévitable *(équivalent à ABBA.AAB)*
3-DOL-évitable *(contient AA)*
ABBACAABA
2-évitable *(contient AABACABBA)*
2-DOL-inévitable *(symétrique de ABAACABBA)*
3-DOL-évitable *(contient AA)*
ABBACAABAB
2-DOL-évitable *(contient ABBACABAB)*
ABBACAABAC
2-DOL-évitable *($f = (abbab, baaba)$)*
ABBACAABB
2-DOL-inévitable *(divise AABBA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBACAABBA
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
ABBACAABBAC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBACAABBACA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBACAABC
2-DOL-évitable *($f = (ab, aa)$)*

ABBACAAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBACAACA)*
3-DOL-évitable *(contient AA)*
ABBACAACA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de ABAABACCA)*
3-DOL-évitable *(contient AA)*
ABBACAACAB
2-DOL-évitable *(contient ABBCACAB)*
ABBACAACB
2-DOL-évitable $(f = (aab, bba))$
ABBACAB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACABA
2-inévitable *(symétrique de ABACABBA)*
3-DOL-évitable *(contient AA)*
ABBACABAA
2-évitable *(symétrique de AABACABBA)*
2-DOL-inévitable *(symétrique de AABACABBA)*
3-DOL-évitable *(contient AA)*
ABBACABAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBACABAB
2-DOL-évitable *(symétrique de ABABCBAAB)*
ABBACABAC
2-évitable *(contient ABBCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
ABBACABACA
2-évitable *(contient ABBCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
ABBACABACAA
2-évitable *(contient ABBCABC)*
2-DOL-inévitable *(symétrique de AABACABACCA)*
3-DOL-évitable *(contient AA)*
ABBACABACAAC
2-DOL-évitable *(contient ABBCABCAC)*
ABBACABACAC
2-DOL-évitable *(contient ABBCABCC)*
ABBACABACB
2-DOL-évitable $(f = (ab, ba))$
ABBACABB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACABBA
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACABBAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABBACABBAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBACABBAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBACABBAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABAABACBAABAC)*
3-DOL-évitable *(contient AA)*
ABBACABBC
2-DOL-évitable $(f = (aab, bba))$
ABBACABC
2-DOL-évitable *(symétrique de ABCACBBC)*
ABBACABCB
2-DOL-évitable *(symétrique de ABACBCAAC)*
ABBACAC
2-évitable *(symétrique de ABABCCB)*
2-DOL-inévitable *(symétrique de ABABCCB)*
3-DOL-évitable *(contient AA)*
ABBACACB
2-évitable *(contient ABBACAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBACACBC
2-DOL-évitable *(symétrique de ABACACBBC)*
ABBACACC
2-DOL-évitable *(symétrique de AABABCCB)*
ABBACB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*

ABBACBA
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACBAA
2-inévitable *(symétrique de AABCABBA)*
3-DOL-évitable *(contient AA)*
ABBACBAAB
2-DOL-évitable $(f = (aab, abbb))$
ABBACBAB
2-inévitable *(divise ABAACBAAB)*
3-DOL-évitable *(contient AA)*
ABBACBABA
2-DOL-évitable *(symétrique de ABABCABBA)*
ABBACBABB
2-inévitable *(divise ABAACBAAB)*
3-DOL-évitable *(contient AA)*
ABBACBABBA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBACBABBAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABBACBABBAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
ABBACBABBACB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBABCBABBABCB)*
3-DOL-évitable *(contient AA)*
ABBACBABBC
2-DOL-évitable $(f = (aab, bba))$
ABBACBABC
2-DOL-évitable $(f = (aab, abb))$
ABBACBAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
ABBACBACA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
ABBACBACAA
2-DOL-évitable *(contient ABBCBCAA)*
ABBACBACC
2-DOL-évitable *(symétrique de AABCABCCB)*
ABBACBB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACBBA
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABBACBBAA
2-DOL-inévitable *(divise ABBAA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBACBBAAB
2-DOL-évitable *(contient ABBACBAAB)*
ABBACBBAAC
2-DOL-évitable *(contient ABBCBBAC)*
ABBACBBAB
2-inévitable *(symétrique de ABAACBAAB)*
3-DOL-évitable *(contient AA)*
ABBACBBABA
2-DOL-évitable *(contient ABBACBABA)*
ABBACBBABB
2-DOL-inévitable *(symétrique de AABAACBAAB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABBACBBC
2-DOL-évitable $(f = (aab, abb))$
ABBACBC
2-évitable *(symétrique de ABACBBC)*
2-DOL-inévitable *(symétrique de ABACBBC)*
3-DOL-évitable *(contient AA)*
ABBACBCA
2-DOL-évitable $(f = (aab, bba))$
ABBACBCB
2-évitable *(contient ABBACBC)*
2-DOL-inévitable *(symétrique de ABABCAAC)*
3-DOL-évitable *(contient AA)*
ABBACBCBA
2-DOL-évitable $(f = (aab, abb))$
ABBACBCBB
2-évitable *(contient ABBACBC)*

2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBACBCBBA
2-D0L-évitable *(contient ABBACCBA)*
ABBACBCC
2-D0L-évitable *(symétrique de AABACBBC)*
ABBACC
2-inévitable *(symétrique de AABCCB)*
3-D0L-évitable *(contient AA)*
ABBACCA
2-D0L-inévitable *(symétrique de ABBACCA)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBACCAA
2-D0L-évitable *(symétrique de AABBACCA)*
ABBACCAB
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBACCABB
2-D0L-évitable *(symétrique de AABCCBAAB)*
ABBACCABC
2-D0L-évitable *($f = (aab, abb)$)*
ABBACCAC
2-évitable *(contient ABBAB)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBACCACB
2-D0L-évitable *($f = (ab, bba)$)*
ABBACCB
2-D0L-inévitable *(divise ABAABCCA)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBACCBA
2-D0L-évitable *($f = (abbb, aaab)$)*
ABBACCBB
2-D0L-évitable *(symétrique de AABBCAAC)*
ABBACCBC
2-D0L-évitable *(symétrique de ABAACBBC)*
ABBC
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCAA
2-inévitable *(divise AABAB)*
3-D0L-évitable *(contient AA)*
ABBCAAB
2-inévitable *(divise AABB)*
3-D0L-évitable *(contient AA)*
ABBCAABA
2-inévitable *(divise AABACABB)*
3-D0L-évitable *(contient AA)*
ABBCAABB
2-inévitable *(divise AABB)*
3-D0L-évitable *(contient AA)*
ABBCAABBA
2-évitable *(contient AABBA)*
2-D0L-inévitable *(divise AABBA)*
3-D0L-évitable *(contient AA)*
ABBCAABBC
2-évitable *(contient ABAAB)*
2-D0L-inévitable *(divise ABBCAABBCAB)*
3-D0L-évitable *(contient AA)*
ABBCAABBCA
2-évitable *(contient ABAAB)*
2-D0L-inévitable *(divise ABBCAABBCAB)*
3-D0L-évitable *(contient AA)*
ABBCAABBCAB
2-évitable *(contient ABAAB)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*
ABBCAABC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-D0L-inévitable *(divise AABBCAABC)*
3-D0L-évitable *(contient AA)*
ABBCAABCA
2-D0L-évitable *($f = (ab, ba)$)*
ABBCAABCB
2-D0L-évitable *(symétrique de ABACCBAAC)*
ABBCAAC
2-D0L-inévitable *(divise ABBCAACA)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCAACA
2-évitable *(contient ABBAB)*
2-D0L-inévitable *(symétrique de ABAABCCA)*
3-D0L-évitable *(contient AA)*
ABBCAACAB
2-D0L-évitable *($f = (ab, ba)$)*
ABBCAACB
2-D0L-évitable *($f = (abbab, baaba)$)*
ABBCAACBA
2-D0L-évitable *($f = (aab, abb)$)*
ABBCAACBB
2-D0L-évitable *(symétrique de AABCCBAAC)*
ABBCAACC
2-D0L-évitable *(symétrique de AABBACCB)*
ABBCAB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCABA
2-inévitable *(divise AABAB)*
3-D0L-évitable *(contient AA)*
ABBCABAA
2-inévitable *(symétrique de AABACBBA)*
3-D0L-évitable *(contient AA)*
ABBCABAB
2-inévitable *(divise ABABB)*
3-D0L-évitable *(contient AA)*
ABBCABABB
2-inévitable *(divise ABABB)*
3-D0L-évitable *(contient AA)*
ABBCABABBC
2-évitable *(contient AABCACAAB)*
2-D0L-inévitable *(divise ABAACBABAAC)*
3-D0L-évitable *(contient AA)*
ABBCABABBCA
2-évitable *(contient AABCACAAB)*
2-D0L-inévitable *(divise ABBCABABBCABB)*
3-D0L-évitable *(contient AA)*
ABBCABABBCAB
2-évitable *(contient AABCACAAB)*
2-D0L-inévitable *(divise ABBCABABBCABB)*
3-D0L-évitable *(contient AA)*
ABBCABABBCABB
2-évitable *(contient AABCACAAB)*
2-D0L-inévitable *(symétrique de AABCAABABCAAB)*
3-D0L-évitable *(contient AA)*
ABBCABABBCB
2-évitable *(contient AABCACAAB)*
2-D0L-inévitable *(symétrique de ABAACACBAAC)*
3-D0L-évitable *(contient AA)*
ABBCABABBCBA
2-D0L-évitable *(contient ABACABCA)*
ABBCABABBCBB
2-D0L-évitable *(contient ABCAABCBB)*
ABBCABABBCBC
2-D0L-évitable *(contient ABAABB)*
ABBCABABC
2-D0L-évitable *($f = (ab, aa)$)*
ABBCABAC
2-D0L-évitable *($f = (ab, ba)$)*
ABBCABB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCABBA
2-inévitable *(divise ABBA)*
3-D0L-évitable *(contient AA)*
ABBCABBAA
2-évitable *(contient ABBAA)*
2-D0L-inévitable *(divise ABBAA)*
3-D0L-évitable *(contient AA)*
ABBCABBAAB
2-D0L-évitable *(contient ABBCBBAAB)*
ABBCABBAB
2-évitable *(contient ABBAB)*
2-D0L-inévitable *(divise ABAABA)*
3-D0L-évitable *(contient AA)*
ABBCABBC
2-évitable *(contient ABCABC)*
2-D0L-inévitable *(cf. A.4)*
3-D0L-évitable *(contient AA)*
ABBCABBCB
2-évitable *(contient ABCABC)*
2-D0L-inévitable *(divise AABBCABBCB)*
3-D0L-évitable *(contient AA)*
ABBCABBCBB
2-D0L-évitable *(symétrique de AABAACBAAC)*
ABBCABBCBC
2-évitable *(contient ABCABC)*

2-D0L-inévitable *(divise ABAACBAACAC)*
3-D0L-évitable *(contient AA)*
ABBCABBCBCA
2-D0L-évitable *($f = (ab, aaa)$)*
ABBCABBCC
2-évitable *(contient ABCABC)*
2-D0L-inévitable *(divise ABAACBAACC)*
3-D0L-évitable *(contient AA)*
ABBCABBCCB
2-évitable *(contient ABCABC)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCABC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$, ou $f = (abc, ac, b)$ et $g = (aabb, bbaa, baba)$)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*
ABBCABCA
2-évitable *(contient ABBCABC)*
2-D0L-inévitable *(divise ABBACABACAA)*
3-D0L-évitable *(contient AA)*
ABBCABCAA
2-évitable *(contient ABBCABC)*
2-D0L-inévitable *(divise ABBACABACAA)*
3-D0L-évitable *(contient AA)*
ABBCABCAAB
2-D0L-évitable *($f = (ab, aa)$)*
ABBCABCAC
2-D0L-évitable *(symétrique de ABACBACCB)*
ABBCABCB
2-évitable *(contient ABBCABC)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCABCBA
2-D0L-évitable *($f = (aab, abb)$)*
ABBCABCBB
2-évitable *(contient ABBCABC)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCABCBBA
2-D0L-évitable *(contient ABCACBBA)*
ABBCABCBBC
2-D0L-évitable *(symétrique de ABBABCABBC)*
ABBCABCBC
2-D0L-évitable *(symétrique de ABABCABBC)*
ABBCABCC
2-D0L-évitable *(symétrique de AABCABBC)*
ABBCAC
2-inévitable *(symétrique de ABACCB)*
3-D0L-évitable *(contient AA)*
ABBCACA
2-évitable *(symétrique de ABABCCA)*
2-D0L-inévitable *(divise AABABCACAA)*
3-D0L-évitable *(contient AA)*
ABBCACAA
2-évitable *(contient ABBCACA)*
2-D0L-inévitable *(divise AABABCACAA)*
3-D0L-évitable *(contient AA)*
ABBCACAB
2-D0L-évitable *($f = (aab, bba)$)*
ABBCACB
2-D0L-évitabilité inconnue
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCACBA
2-D0L-évitable *($f = (ab, aa)$)*
ABBCACBB
2-D0L-évitabilité inconnue
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCACBBA
2-D0L-évitable *($f = (ab, ba)$)*
ABBCACBBC
2-D0L-évitable *(symétrique de ABBACABBC)*
ABBCACBC
2-évitable *(symétrique de ABACABBC)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCACBCA
2-D0L-évitable *($f = (ab, ba)$)*
ABBCACBCB
2-D0L-évitable *(symétrique de ABABCBAAC)*
ABBCACBCC
2-D0L-évitable *(symétrique de AABACABBC)*
ABBCACC
2-D0L-inévitable *(divise ABCCABAA)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCACCA
2-D0L-évitable *(symétrique de ABBABCCA)*
ABBCACCB
2-D0L-évitable *($f = (abbab, baaba)$)*
ABBCACCBA
2-D0L-évitable *($f = (aab, abb)$)*
ABBCACCBB
2-D0L-évitable *(symétrique de AABBCBAAC)*
ABBCACCBC
2-D0L-évitable *(symétrique de ABAACABBC)*
ABBCB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCBA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCBAA
2-inévitable *(divise ABBACBAA)*
3-D0L-évitable *(contient AA)*
ABBCBAAB
2-inévitable *(symétrique de ABBACAAB)*
3-D0L-évitable *(contient AA)*
ABBCBAABB
2-évitable *(contient ABBAA)*
2-D0L-inévitable *(divise ABBAA)*
3-D0L-évitable *(contient AA)*
ABBCBAABBC
2-évitable *(contient ABBAA)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCBAABBCA
2-D0L-évitable *(contient ABCBAABCA)*
ABBCBAABBCB
2-évitable *(contient ABBAA)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCBAABBCBA
2-évitable *(contient ABBAA)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCBAABC
2-D0L-évitable *($f = (aab, bba)$)*
ABBCBAAC
2-D0L-évitable *(symétrique de ABBCACCB)*
ABBCBAACA
2-D0L-évitable *(symétrique de ABAACBCCA)*
ABBCBAACB
2-D0L-évitable *($f = (aabab, baab)$, ou $f = (abbab, baaba)$, ou $f = (abaab, babba)$)*
ABBCBAACBC
2-D0L-évitable *(symétrique de ABACCBABBC)*
ABBCBAB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCBABA
2-inévitable *(divise ABABCBAA)*
3-D0L-évitable *(contient AA)*
ABBCBABAC
2-D0L-évitable *($f = (ab, bba)$)*
ABBCBABB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*
ABBCBABBA
2-évitable *(contient ABAAB)*
2-D0L-inévitable *(divise ABAABA)*
3-D0L-évitable *(contient AA)*
ABBCBABBAB
2-évitable *(contient ABAAB)*
2-D0L-inévitable *(divise ABAABA)*
3-D0L-évitable *(contient AA)*
ABBCBABBC
2-D0L-inévitable *(divise AABCAABABCAAB)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*
ABBCBABBCB
2-évitable *(contient ABCABC)*
2-D0L-inévitable *(divise ABAACAABAACA)*
3-D0L-évitable *(contient AA)*
ABBCBABBCBB
2-évitable *(contient ABCABC)*
2-D0L-inévitable *(divise ABBCABABBCABB)*
3-D0L-évitable *(contient AA)*
ABBCBABBCBC
2-D0L-évitable *(symétrique de ABABBCBABBC)*
ABBCBABBCC
2-D0L-évitable *(symétrique de AABBCBABBC)*
ABBCBABC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*

2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBABCB
2-D0L-évitable *(symétrique de ABACABAAC)*

ABBCBABCC
2-D0L-évitable *(symétrique de AABCBABBC)*

ABBCBAC
2-D0L-inévitable
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBACA
2-D0L-évitable *(symétrique de ABACBCCA)*

ABBCBACB
2-évitable *($f = (abc, ac, b)$ et $g = (aaa, aba, bbb)$)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBACBA
2-D0L-évitable *($f = (aabab, baab)$, ou $f = (abbab, baaba)$, ou $f = (abaab, babba)$)*

ABBCBACBB
2-D0L-évitable *(symétrique de AABCABAAC)*

ABBCBACBBC
2-D0L-évitable *(symétrique de ABBACBABBC)*

ABBCBACBC
2-D0L-évitable *(symétrique de ABACBABBC)*

ABBCBACC
2-D0L-évitable *(symétrique de AABCACCB)*

ABBCBB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*

ABBCBBA
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*

ABBCBBAA
2-inévitable *(symétrique de AABBCBBA)*
3-D0L-évitable *(contient AA)*

ABBCBBAAB
2-D0L-évitable *(symétrique de ABBAACAAB)*

ABBCBBAB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*

ABBCBBABA
2-inévitable *(divise AABABCBAA)*
3-D0L-évitable *(contient AA)*

ABBCBBABB
2-inévitable *(divise AA)*
3-D0L-évitable *(contient AA)*

ABBCBBABBC
2-évitable *($f = (abc, ac, b)$ et $g = (aabbb, baaa, bbabab)$)*
2-D0L-inévitable *(divise ABAACAABAACA)*
3-D0L-évitable *(contient AA)*

ABBCBBABBCB
2-évitable *(contient ABBCBBABBC)*
2-D0L-inévitable *(divise ABAACAABAACA)*
3-D0L-évitable *(contient AA)*

ABBCBBABBCBB
2-D0L-évitable *(contient ABBABB)*

ABBCBBABBCBC
2-D0L-évitable *(contient ABCBBABCBC)*

ABBCBBABC
2-D0L-évitable *($f = (ab, aa)$)*

ABBCBBAC
2-D0L-évitable *($f = (ab, ba)$)*

ABBCBC
2-inévitable *(divise AABAB)*
3-D0L-évitable *(contient AA)*

ABBCBCA
2-D0L-inévitable
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBCAA
2-D0L-évitable *(symétrique de AABCBCCA)*

ABBCBCAB
2-D0L-inévitable
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBCABA
2-évitable *(contient ABABCAC)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBCABAB
2-évitable *(contient ABABCAC)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCBCABABB
2-D0L-évitable *(contient ABCBCAAB)*

ABBCBCABB
2-évitable *(symétrique de AABCACAAB)*
2-D0L-inévitable *(divise AABCAABABCAAB)*
3-D0L-évitable *(contient AA)*

ABBCBCABBA
2-D0L-évitable *(contient ABCCABBA)*

ABBCBCABBC
2-évitable *(contient ABBCBCABB)*
2-D0L-inévitable *(divise AABCAABABCAAB)*
3-D0L-évitable *(contient AA)*

ABBCBCABBCA
2-D0L-évitable *($f = (ab, aaa)$)*

ABBCBCABBCB
2-évitable *(contient ABBCBCABB)*
2-D0L-inévitable *(symétrique de ABAACBABAAC)*
3-D0L-évitable *(contient AA)*

ABBCBCABBCBB
2-D0L-évitable *(contient ABCCABCBB)*

ABBCBCABBCBC
2-D0L-évitable *(contient ABBABB)*

ABBCBCABC
2-D0L-évitable *($f = (ab, aa)$)*

ABBCBCAC
2-D0L-évitable *(symétrique de ABACACCB)*

ABBCC
2-inévitable *(divise AABB)*
3-D0L-évitable *(contient AA)*

ABBCCA
2-inévitable *(longueur maximale: 22)*
3-D0L-évitable *(contient AA)*

ABBCCAA
2-D0L-évitable *(symétrique de AABBCCA)*

ABBCCAB
2-D0L-inévitable
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCCABA
2-D0L-évitable *(symétrique de ABACCBBA)*

ABBCCABB
2-D0L-inévitable *(divise ABCAABBCAAB)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCCABBC
2-évitable *(contient ABBAB)*
2-D0L-inévitable *(divise ABCAABBCAAB)*
3-D0L-évitable *(contient AA)*

ABBCCABBCA
2-D0L-évitable *($f = (ab, ba)$)*

ABBCCABBCB
2-évitable *(contient ABBAB)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCCABBCC
2-D0L-évitable *(contient ABBABB)*

ABBCCABC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-D0L-inévitable *(divise ABBCAABBCAB)*
3-D0L-évitable *(contient AA)*

ABBCCABCA
2-D0L-évitable *($f = (ab, ba)$)*

ABBCCABCB
2-D0L-évitable *(symétrique de ABACBBAAC)*

ABBCCABCC
2-D0L-évitable *(symétrique de AABCAABBC)*

ABBCCAC
2-D0L-inévitable *(symétrique de ABAACCB)*
2-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCCACA
2-D0L-évitable *(symétrique de ABABBCCA)*

ABBCCACB
2-D0L-évitable *($f = (ab, ba)$)*

ABBCCACC
2-D0L-évitable *(symétrique de AABAACCB)*

ABBCCB
2-évitable *(contient AABBA)*
2-D0L-inévitable *(divise AABBA)*
3-D0L-évitable *(contient AA)*

ABBCCBA
2-évitable *(contient AABBA)*
2-D0L-inévitable
3-D0L-évitable *(contient AA)*

ABBCCBAA
2-évitable *(contient AABBA)*
2-D0L-évitabilité inconnue
3-D0L-évitable *(contient AA)*

ABBCCBAB
- 2-évitable *(contient AABBA)*
- 2-DOL-inévitable *(symétrique de ABACCAAB)*
- 3-DOL-évitable *(contient AA)*

ABBCCBABA
- 2-DOL-évitable *(symétrique de ABABCCBBA)*

ABBCCBABB
- 2-évitable *(contient AABBA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABBCCBABBA
- 2-DOL-évitable *(symétrique de ABBABCCBBA)*

ABBCCBABBC
- 2-évitable *(contient AABBA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABBCCBABBCB
- 2-évitable *(contient AABBA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABBCCBABBCBB
- 2-DOL-évitable *(contient ABBCABCC)*

ABBCCBABBCC
- 2-évitable *(contient AABBA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABBCCBABBCCB
- 2-évitable *(contient AABBA)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABBCCBABC
- 2-DOL-évitable *(*$f = (aab, abb)$*)*

ABBCCBAC
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABC
- inévitable

ABCA
- inévitable

ABCAA
- 2-inévitable *(divise AA)*
- 3-DOL-évitable *(contient AA)*

ABCAAB
- 2-inévitable *(divise AA)*
- 3-DOL-évitable *(contient AA)*

ABCAABA
- 2-inévitable *(divise AA)*
- 3-DOL-évitable *(contient AA)*

ABCAABAA
- 2-inévitable *(divise AA)*
- 3-DOL-évitable *(contient AA)*

ABCAABAB
- 2-inévitable *(divise AABAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABABC
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aabb, bbaa, baba)$*)*
- 2-DOL-inévitable *(divise AABCAABABCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABABCA
- 2-évitable *(contient ABCAABABC)*
- 2-DOL-inévitable *(divise AABCAABABCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABABCAA
- 2-évitable *(contient ABCAABABC)*
- 2-DOL-inévitable *(divise AABCAABABCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABABCAAB
- 2-évitable *(contient ABCAABABC)*
- 2-DOL-inévitable *(divise AABCAABABCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABABCAABA
- 2-DOL-évitable *(contient ABABA)*

ABCAABAC
- 2-évitable *(*$f = (abc, ac, b)$ *et* $g = (aaab, abab, bb)$*)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABCAABACA
- 2-DOL-évitable *(symétrique de ABACAABCA)*

ABCAABACC
- 2-évitable *(contient ABCAABAC)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABCAABACCA
- 2-DOL-évitable *(contient ABCACBBC)*

ABCAABB
- 2-inévitable *(divise AABB)*
- 3-DOL-évitable *(contient AA)*

ABCAABBA
- 2-évitable *(contient AABBA)*
- 2-DOL-inévitable *(divise AABBA)*
- 3-DOL-évitable *(contient AA)*

ABCAABBC
- 2-évitable *(symétrique de ABBCCABC)*
- 2-DOL-inévitable *(divise ABCAABBCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABBCA
- 2-évitable *(contient ABCAABBC)*
- 2-DOL-inévitable *(divise ABCAABBCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABBCAA
- 2-évitable *(contient ABCAABBC)*
- 2-DOL-inévitable *(divise ABCAABBCAAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABBCAAB
- 2-évitable *(contient ABCAABBC)*
- 2-DOL-inévitable *(symétrique de ABBCAABBCAB)*
- 3-DOL-évitable *(contient AA)*

ABCAABBCAABA
- 2-DOL-évitable *(symétrique de ABAACBBAACBA)*

ABCAABBCAB
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABCAABC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABCAABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABCAABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCAA
- 2-DOL-évitable *(contient ABABA)*

ABCAABCAB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABACABBACABA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCAC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABCAABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCACA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABCAABCACAA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCACAA
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(symétrique de AABABCAABCA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCB
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABACABBACAA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCBA
- 2-DOL-évitable *(*$f = (ab, ba)$*)*

ABCAABCBB
- 2-DOL-évitable *(symétrique de AABACCBAC)*

ABCAABCBC
- 2-DOL-évitable *(contient ABAABB)*

ABCAABCC
- 2-évitable *(contient ABAAB)*
- 2-DOL-inévitable *(divise ABACABBACAA)*
- 3-DOL-évitable *(contient AA)*

ABCAABCCA
- 2-DOL-évitable *(symétrique de ABBCAABCA)*

ABCAAC
- 2-inévitable *(divise ABBA)*
- 3-DOL-évitable *(contient AA)*

ABCAACA
- 2-évitable *(contient ABBAB)*
- 2-DOL-inévitable *(divise ABAABA)*
- 3-DOL-évitable *(contient AA)*

ABCAACAB
- 2-évitable *(contient ABBAB)*
- 2-DOL-inévitable *(divise ABACABAABACABA)*
- 3-DOL-évitable *(contient AA)*

ABCAACABA
- 2-DOL-évitable *(symétrique de ABACAACBA)*

ABCAACABB
- 2-évitable *(contient ABBAB)*
- 2-DOL-évitabilité inconnue
- 3-DOL-évitable *(contient AA)*

ABCAACABBA
- 2-DOL-évitable *(contient ABCACBBA)*

ABCAACABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCAACABCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCAACABCAA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCAACABCAAB
2-DOL-évitable *(contient ABCCBABCA)*
ABCAACABCAAC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCAACABCAACA
2-DOL-évitable *(contient ABABA)*
ABCAACABCAB
2-DOL-évitable *(contient ABCACBCB)*
ABCAACABCAC
2-DOL-évitable *(contient ABAACABAC)*
ABCAACABCC
2-DOL-évitable *(contient ABAACABC)*
ABCAACB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCAACBA
2-évitable *($f = (abc, ac, b)$ et $g = (aabbb, baaa, bbabab)$)*
2-DOL-inévitable *(symétrique de ABCAACBA)*
3-DOL-évitable *(contient AA)*
ABCAACBAC
2-DOL-évitable *($f = (aab, abb)$)*
ABCAACBB
2-DOL-inévitable *(symétrique de AABCCBAC)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCAACBBA
2-DOL-évitable *(symétrique de ABBCAACBA)*
ABCAACBBC
2-DOL-évitable *(symétrique de ABBACCABC)*
ABCAACBC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCAACBCA
2-DOL-évitable *($f = (aab, abb)$)*
ABCAACBCC
2-DOL-évitable *(symétrique de AABACCABC)*
ABCAACC
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABCAACCB
2-DOL-évitable *(symétrique de ABBCCBAC)*
ABCAB
inévitable
ABCABA
inévitable
ABCABAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCABAAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCABAABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCABAABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABAABCA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABAABCAA
2-DOL-évitable *(contient ABCBAABCA)*
ABCABAABCAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABAABCABA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABAABCABAA
2-DOL-évitable *(contient ABABA)*
ABCABAAC
2-évitable *(symétrique de ABBCBACB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABAB
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABCABABB
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABCABABBC
2-DOL-évitable *(symétrique de ABBCBCABC)*
ABCABABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABCABABCA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise AABCAABABCAAB)*
3-DOL-évitable *(contient AA)*
ABCABABCAA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACBABACBB)*
3-DOL-évitable *(contient AA)*
ABCABABCAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise AABCAABABCAAB)*
3-DOL-évitable *(contient AA)*
ABCABABCABA
2-DOL-évitable *(contient ABABA)*
ABCABABCABB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*
ABCABABCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABABCBA
2-DOL-évitable *(contient ABACABCA)*
ABCABABCBB
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABABCC
2-DOL-évitable *(contient ABAABB)*
ABCABAC
2-DOL-inévitable *(divise ABCABCBA)*
2-évitabilité inconnue
3-DOL-évitable *($f = (ab, bc, ca)$)*
ABCABACA
2-DOL-évitable *(symétrique de ABACABCA)*
ABCABACB
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABCABAC)*
ABCABACBA
2-DOL-évitable *($f = (aab, abb)$)*
ABCABACBB
2-évitable *(symétrique de AABCACBAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient ABCABAC)*
ABCABACBC
2-DOL-évitable *(symétrique de ABACBCABC)*
ABCABACC
2-DOL-évitable *(symétrique de AABCBACB)*
ABCABB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCABBA
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABCABBAA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABCABBAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCABBABC
2-évitable *(contient ABBAB)*

2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABBABCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABBABCAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCABBABCABB
2-DOL-évitable *(contient ABBABB)*
ABCABBABCC
2-DOL-évitable *(symétrique de AABCBBCABC)*
ABCABBAC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABBACA
2-DOL-évitable *(symétrique de ABACCABCA)*
ABCABBACB
2-DOL-évitable *(symétrique de ABCAACBAC)*
ABCABBACC
2-DOL-évitable *(symétrique de AABCCBACB)*
ABCABBC
2-évitable *(symétrique de ABBCABC)*
2-DOL-inévitable *(symétrique de ABBCABC)*
3-DOL-évitable *(contient AA)*
ABCABBCA
2-évitable *(contient ABCABBC)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABCABBCAA
2-évitable *(contient ABCABBC)*
2-DOL-inévitable *(divise ABACABBACAA)*
3-DOL-évitable *(contient AA)*
ABCABBCAB
2-évitable *(contient ABCABBC)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABCABBCABA
2-évitable *(contient ABCABBC)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABCABBCABAB
2-DOL-évitable *(symétrique de ABABCAABCAB)*
ABCABBCABC
2-DOL-évitable *($f = (ab, aa)$)*
ABCABBCAC
2-DOL-évitable *(symétrique de ABACCBACB)*
ABCABBCB
2-évitable *(contient ABCABBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABBCBA
2-évitable *(contient ABCABBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABBCBAA
2-DOL-évitable *(symétrique de AABCBBACBA, contient ABACBCAA)*
ABCABBCBB
2-évitable *(contient ABCABBC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABBCBC
2-DOL-évitable *(symétrique de ABABBCABC)*
ABCABBCC
2-évitable *(contient ABCABBC)*
2-DOL-inévitable *(divise ABCABCCAA)*
3-DOL-évitable *(contient AA)*
ABCABBCCA
2-DOL-évitable *(symétrique de ABBCCABCA)*
ABCABBCCB
2-DOL-évitable *(symétrique de ABBAACBAC)*
ABCABC
2-évitable *($f = (abc, ac, b)$ et $g = (abab, aabb, baaa)$)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCABCA
2-DOL-évitable *(contient ABABA)*
ABCABCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABACBAC)*
3-DOL-évitable *(contient AA)*
ABCABCBA
2-évitable *(contient ABCABC)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABCABCBAC
2-DOL-évitable *($f = (aab, abb)$)*
ABCABCBB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCABCBBA
2-DOL-évitable *(symétrique de ABBCBACBA)*
ABCABCBBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCABCBBCA
2-DOL-évitable *($f = (aab, bba)$)*
ABCABCBBCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABAABACBAC)*
3-DOL-évitable *(contient AA)*
ABCABCBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*
ABCABCBCA
2-DOL-évitable *(contient ABABBA)*
ABCABCBCC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*
ABCABCC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*
ABCABCCA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*
ABCABCCAA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de AABBCABCA)*
3-DOL-évitable *(contient AA)*
ABCABCCAB
2-DOL-évitable *(contient ABABBA)*
ABCABCCAC
2-DOL-évitable *(symétrique de ABAACBACB)*
ABCABCCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise AABACABACCA)*
3-DOL-évitable *(contient AA)*
ABCABCCBA
2-DOL-évitable *($f = (ab, ba)$)*
ABCABCCBB
2-évitable *(contient ABCABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCABCCBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCBABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCABCCBCA
2-DOL-évitable *($f = (aab, bba)$)*
ABCAC
inévitable
ABCACA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABCACAA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABCACAAB
2-DOL-inévitable *(symétrique de ABBCBCAB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACAABA
2-évitable *(contient ABCBCCAC)*
2-DOL-inévitable *(divise ABBCABABBCB)*
3-DOL-évitable *(contient AA)*
ABCACAABAB
2-évitable *(contient ABCBCCAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACAABB
2-DOL-évitable *(symétrique de AABBCBCAB)*
ABCACAABC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*

ABCACAABCA
2-évitable *(contient ABCACAABC)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*
ABCACAABCAA
2-évitable *(contient ABCACAABC)*
2-DOL-inévitable *(divise ABBCABABBCABB)*
3-DOL-évitable *(contient AA)*
ABCACAABCAAB
2-DOL-évitable *(contient ABBABA)*
ABCACAABCAB
2-DOL-évitable *(contient ABCBCABA)*
ABCACAB
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACABA
2-DOL-inévitable *(symétrique de ABACACBA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACABAA
2-évitable *(symétrique de AABACACBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACABAAC
2-DOL-évitable *(contient ABCBACCB)*
ABCACABAB
2-DOL-évitable *(symétrique de ABABCBCAB)*
ABCACABAC
2-DOL-évitable *(f = (ab, aa))*
ABCACABB
2-DOL-évitable *(symétrique de AABCBCAB)*
ABCACABC
2-évitable *(f = (abc, ac, b) et g = (aaa, aba, bbb))*
2-DOL-inévitable *(symétrique de ABCACABC)*
3-DOL-évitable *(contient AA)*
ABCACABCA
2-évitable *(contient ABCACABC)*
2-DOL-inévitable *(divise AABCAABABCAAB)*
3-DOL-évitable *(contient AA)*
ABCACABCAA
2-évitable *(contient ABCACABC)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCACABCAC
2-évitable *(contient ABCACABC)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCACABCACA
2-DOL-évitable *(contient ABABA)*
ABCACABCC
2-évitable *(contient ABCACABC)*
2-DOL-inévitable *(divise ABACBABACBB)*
3-DOL-évitable *(contient AA)*
ABCACABCCA
2-DOL-évitable *(contient ABACABCA)*
ABCACB
2-DOL-inévitable *(divise ABCABCBA)*
2-évitabilité inconnue
3-évitable *(f = (ab, ac, dc, db) et g = (ab, ac, bb, cc), ou f = (ab, ac, dc, db) et g = (ab, ac, cc, bb))*
3-DOL-évitabilité inconnue
4-DOL-évitable *(f = (ab, ac, dc, db))*
ABCACBA
2-DOL-inévitable *(symétrique de ABCACBA)*
2-évitabilité inconnue
3-DOL-évitable *(f = (ab, bc, ca))*
ABCACBAA
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABCACBA)*
ABCACBAB
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABCACBA)*
ABCACBABB
2-DOL-évitable *(symétrique de AABACBCAB)*
ABCACBABC
2-DOL-évitable *(f = (ab, bba))*
ABCACBAC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABCACBA)*
ABCACBACA
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient ABCACBA)*
ABCACBACB
2-DOL-évitable *(symétrique de ABCABCBAC)*
ABCACBACC
2-DOL-évitable *(symétrique de AABCABACB)*
ABCACBB
2-DOL-inévitable *(symétrique de AABCBAC)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACBBA
2-DOL-évitable *(symétrique de ABBCACBA)*
ABCACBBC
2-DOL-évitable *(f = (aaba, bbab))*
ABCACBBCA
2-DOL-évitable *(f = (ab, baa))*
ABCACBBCB
2-DOL-évitable *(symétrique de ABAABCBAC)*
ABCACBBCC
2-DOL-évitable *(symétrique de AABBACABC)*
ABCACBC
2-DOL-inévitable *(symétrique de ABACABC)*
2-évitabilité inconnue
3-évitable *(symétrique de ABACABC)*
3-DOL-évitabilité inconnue
4-DOL-évitable *(symétrique de ABACABC)*
ABCACBCA
2-DOL-inévitable *(divise ABACAACABACAAC)*
2-évitabilité inconnue
3-DOL-évitable *(f = (ab, bc, ca))*
ABCACBCAA
2-évitable *(contient ABCBABCC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient ABCACBCA)*
ABCACBCAAC
2-DOL-évitable *(symétrique de ABBACABACB)*
ABCACBCAB
2-DOL-évitable *(f = (aab, abb))*
ABCACBCAC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise AABCACABCAC)*
3-DOL-évitable *(contient ABCACBCA)*
ABCACBCACA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABABCBABCA)*
3-DOL-évitable *(contient ABCACBCA)*
ABCACBCACC
2-DOL-évitable *(symétrique de AABACABACB)*
ABCACBCB
2-DOL-évitable *(symétrique de ABABCBAC)*
ABCACBCC
2-DOL-évitable *(symétrique de AABACABC)*
ABCACC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCACCA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCACCAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable
3-DOL-évitable *(contient AA)*
ABCACCABB
2-DOL-évitable *(symétrique de AABCCBCAB)*
ABCACCABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCACCABCA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCACCABCAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCACCABCACC
2-DOL-évitable *(contient ABBABB)*
ABCACCABCB
2-DOL-évitable *(symétrique de ABACBBCBAC)*
ABCACCAC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCACCACB
2-DOL-évitable *(f = (ab, bba))*
ABCACCB
2-DOL-inévitable *(symétrique de ABBCBAC)*

2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCACCBA
2-DOL-évitable $(f = (ab, aa))$
ABCACCBB
2-DOL-évitable *(symétrique de AABBCBAC)*
ABCACCBC
2-DOL-évitable *(symétrique de ABAACABC)*
ABCB
inévitable
ABCBA
inévitable
ABCBAA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCBAAB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*
ABCBAABA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCBAABB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*
ABCBAABBC
2-DOL-évitable *(symétrique de ABBCCBABC)*
ABCBAABC
2-DOL-inévitable *(divise ABACABBACABA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBAABCA
2-DOL-évitable *(symétrique de ABCAACBCA)*
ABCBAABCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABCBAABCBA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABCBAABCBAA
2-DOL-évitable *(contient ABABA)*
ABCBAABCBAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*
ABCBAABCBB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABBACAA)*
3-DOL-évitable *(contient AA)*
ABCBAABCBBC
2-DOL-évitable *(symétrique de ABBABCCBABC)*
ABCBAAC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBAACA
2-évitable *(symétrique de ABAACBCA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBAACAB
2-DOL-évitable $(f = (ab, ba))$
ABCBAACB
2-DOL-évitable $(f = (abbab, baaba))$
ABCBAACBA
2-DOL-évitable *(symétrique de ABCAABCBA)*
ABCBAACBB
2-DOL-évitable *(symétrique de AABCCABAC)*
ABCBAACBC
2-DOL-évitable *(symétrique de ABACCBABC)*
ABCBAACC
2-DOL-évitable *(symétrique de AABBCACB)*
ABCBAB
inévitable
ABCBABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABCBABAA
2-inévitable *(divise ABABB)*
3-DOL-évitable *(contient AA)*
ABCBABAC
2-évitable *(symétrique de ABCBCACB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBABACA
2-DOL-évitable *(symétrique de ABACACBCA)*
ABCBABACB
2-DOL-évitable *(symétrique de ABCACABAC)*
ABCBABACC
2-DOL-évitable *(symétrique de AABCBCACB)*
ABCBABB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCBABBA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCBABBAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCBABBABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBABBABCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBABBABCBA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBABBABCBAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBABBABCBABB
2-DOL-évitable *(contient ABBABB)*
ABCBABBABCC
2-DOL-évitable *(contient ABCACABB)*
ABCBABBAC
2-DOL-évitable $(f = (ab, aa))$
ABCBABBC
2-évitable *(symétrique de ABBCBABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBABBCA
2-DOL-évitable $(f = (aab, abb))$
ABCBABBCB
2-DOL-évitable *(symétrique de ABAACABAC)*
ABCBABBCC
2-DOL-évitable *(symétrique de AABBCBABC)*
ABCBABC
2-évitabilité inconnue
3-DOL-inévitable *(cf. A.4)*
3-évitabilité inconnue
4-DOL-évitable *(cf. [18])*
ABCBABCA
2-DOL-inévitable *(divise ABABCBABCA)*
2-évitabilité inconnue
3-DOL-évitable *(symétrique de ABCACBCA)*
ABCBABCAA
2-évitable *(symétrique de AABCACBCA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient ABCBABCA)*
ABCBABCAC
2-DOL-évitable *(symétrique de ABACBCACB)*
ABCBABCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABACABAC)*
3-DOL-évitable *(contient AA)*
ABCBABCBA
2-DOL-évitable *(contient ABABA)*
ABCBABCBB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABBACABACAA)*
3-DOL-évitable *(contient AA)*
ABCBABCBBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCBABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCBABCBBCB
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABAABACABAC)*
3-DOL-évitable *(contient AA)*
ABCBABCBBCBA
2-DOL-évitable *(contient ABABBA)*
ABCBABCBC
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*

ABCBABCBCC
2-DOL-évitable *(symétrique de AABABCBABC)*
ABCBABCC
2-évitable *(symétrique de AABCBABC)*
2-DOL-inévitable *(divise AABACABACCA)*
3-DOL-évitable *(contient AA)*
ABCBABCCA
2-DOL-évitable *(symétrique de ABBCACBCA)*
ABCBABCCB
2-évitable *(contient ABCBABCC)*
2-DOL-inévitable *(divise AABACABACCA)*
3-DOL-évitable *(contient AA)*
ABCBABCCBA
2-DOL-évitable $(f = (ab, aaa))$
ABCBABCCBB
2-DOL-évitable *(symétrique de AABBACABAC)*
ABCBAC
2-DOL-inévitable *(divise AABCBAC)*
2-évitabilité inconnue
3-évitable *(symétrique de ABCACB)*
3-DOL-évitabilité inconnue
4-DOL-évitable *(symétrique de ABCACB)*
ABCBACA
2-DOL-inévitable *(symétrique de ABACBCA)*
2-évitabilité inconnue
3-DOL-évitable *(symétrique de ABACBCA)*
ABCBACAA
2-évitable *(contient ABACBCC)*
2-DOL-inévitable *(symétrique de AABACBCA)*
3-DOL-évitable *(contient ABCBACA)*
ABCBACAAC
2-DOL-évitable *(symétrique de ABBABCACB)*
ABCBACAB
2-DOL-inévitable *(symétrique de ABCBACAB)*
2-évitabilité inconnue
3-DOL-évitable *(contient ABCBACA)*
ABCBACABA
2-DOL-évitable *(symétrique de ABACABCBA)*
ABCBACABB
2-DOL-évitable *(symétrique de AABCBACAB)*
ABCBACABC
2-DOL-évitable *(symétrique de ABCACBABC)*
ABCBACAC
2-DOL-évitable *(symétrique de ABABCACB)*
ABCBACB
2-DOL-inévitable *(divise ABCBACBB)*
2-évitabilité inconnue
3-DOL-évitable *(symétrique de ABCABAC)*
ABCBACBA
2-évitable *(contient ABCABC)*
2-DOL-inévitable *(symétrique de ABCABCBA)*
3-DOL-évitable *(contient ABCBACB)*
ABCBACBAA
2-DOL-évitable *(symétrique de AABCABCBA)*
ABCBACBAB
2-DOL-évitable *(symétrique de ABACBACAB)*
ABCBACBB
2-évitable *(symétrique de AABCABAC)*
2-DOL-inévitable *(symétrique de AABCABAC)*
3-DOL-évitable *(contient ABCBACB)*
ABCBACBBA
2-DOL-évitable *(symétrique de ABBCABCBA)*
ABCBACBBC
2-DOL-évitable *(symétrique de ABBACBABC)*
ABCBACBC
2-DOL-inévitable *(symétrique de ABACBABC)*
2-évitabilité inconnue
3-DOL-évitable *(contient ABCBACB)*
ABCBACBCA
2-DOL-évitable $(f = (aab, abb))$
ABCBACBCB
2-DOL-évitable *(symétrique de ABABCABAC)*
ABCBACBCC
2-DOL-évitable *(symétrique de AABACBABC)*
ABCBACC
2-DOL-inévitable *(symétrique de AABCACB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBACCA
2-DOL-évitable *(symétrique de ABBACBCA)*
ABCBACCB
2-DOL-évitable *(symétrique de ABBCABAC)*
ABCBB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCBBA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCBBAA
2-inévitable *(divise AABBCBBA)*
3-DOL-évitable *(contient AA)*
ABCBBAAB
2-évitable *(contient AABBA)*
2-DOL-inévitable *(divise AABBA)*
3-DOL-évitable *(contient AA)*
ABCBBAABC
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBAABCA
2-DOL-évitable *(contient ABCCAABA)*
ABCBBAABCB
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBAABCBA
2-DOL-évitable *(contient ABCBAABCA)*
ABCBBAABCBB
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBAABCBBA
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBAABCBBAA
2-DOL-évitable *(contient ABABA)*
ABCBBAABCBBAB
2-évitable *(contient AABBA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBAAC
2-DOL-évitable *(symétrique de ABBCCACB)*
ABCBBAB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCBBABA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABCBBABAC
2-DOL-évitable $(f = (ab, aa))$
ABCBBABB
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*
ABCBBABBC
2-DOL-évitable *(symétrique de ABBCBBABC)*
ABCBBABC
2-DOL-inévitable *(divise ABACABAABACABA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBABCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBBABCBB
2-DOL-évitable *(contient ABBABB)*
ABCBBABCBC
2-DOL-évitable *(symétrique de ABABCBBABC)*
ABCBBABCC
2-évitable *(symétrique de AABCBBABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBABCCB
2-DOL-évitable *(symétrique de ABBACAABAC)*
ABCBBABCCBB
2-DOL-évitable *(symétrique de AABBACAABAC)*
ABCBBAC
2-DOL-inévitable
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBACA
2-évitable *(symétrique de ABACCBCA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBACAA
2-DOL-évitable *(symétrique de AABACCBCA)*
ABCBBACAB
2-DOL-évitable *(symétrique de ABCBAACAB)*
ABCBBACB
2-évitable *(symétrique de ABCAABAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBACBA
2-évitable *(contient ABCBBACB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBBACBAB
2-DOL-évitable *(symétrique de ABACBAACAB)*
ABCBBACBB
2-DOL-évitable *(symétrique de AABCAABAC)*
ABCBBACBC
2-DOL-évitable *(symétrique de ABACBBABC)*
ABCBBACC
2-DOL-évitable *(symétrique de AABCCACB)*
ABCBBC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCBBCA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABCACCAB)*
3-DOL-évitable *(contient AA)*
ABCBBCAA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de AABCCBCA)*
3-DOL-évitable *(contient AA)*
ABCBBCAAB
2-DOL-évitable *(symétrique de ABBCAACAB)*
ABCBBCAAC
2-DOL-évitable *(symétrique de ABBACCACB)*
ABCBBCAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCBBCABA
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBCABAB
2-DOL-évitable *(contient ABCBACAC)*
ABCBBCABB
2-DOL-évitable *(symétrique de AABCAACAB)*
ABCBBCABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCBBCABCA
2-DOL-évitable *(symétrique de ABCABCCBCA)*
ABCBBCABCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCBBCABCBB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCBBCABCBBA
2-DOL-évitable *(contient ABCBABCCA)*
ABCBBCABCBBC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*
ABCBBCABCBBCA
2-DOL-évitable *(contient ABABA)*
ABCBBCABCBBCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de ABAABACBAABAC)*
3-DOL-évitable *(contient AA)*
ABCBBCABCBBCBB
2-DOL-évitable *(contient ABBCABBCBB)*
ABCBBCAC
2-DOL-évitable *(symétrique de ABACCACB)*
ABCBBCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCBBCBA
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(symétrique de ABCBBCBA)*
3-DOL-évitable *(contient AA)*
ABCBBCBAA
2-évitable *(contient ABAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBBCBAAB
2-DOL-évitable *(contient ABCBCAAB)*
ABCBBCBAB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBBCBABA
2-DOL-évitable *(contient ABCBCABA)*
ABCBBCBABB
2-DOL-évitable *(symétrique de AABACAACAB)*
ABCBBCBABC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBBCBABCA
2-DOL-évitable *(contient ABCBCABA)*
ABCBBCBABCB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCBBCBABCBA
2-DOL-évitable *(contient ABBABA)*
ABCBBCBABCBB
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCBBCBABCBBC
2-évitable *(contient ABAAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCBBCBABCBBCB
2-DOL-évitable *(contient ABBABB)*
ABCBBCBABCBBCC
2-DOL-évitable *(contient ABBCBABBCC)*
ABCBBCBAC
2-DOL-évitable *(symétrique de ABCACCACB)*
ABCBC
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*
ABCBCA
2-inévitable *(longueur maximale: 43)*
3-DOL-évitable *(contient AA)*
ABCBCAA
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBCAAB
2-DOL-évitable *(symétrique de ABBCACAB)*
ABCBCAAC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBCAACA
2-DOL-évitable *(symétrique de ABAABCBCA)*
ABCBCAACC
2-DOL-évitable *(symétrique de AABBACACB)*
ABCBCAB
2-DOL-inévitable *(symétrique de ABCACAB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBCABA
2-DOL-évitable *(symétrique de ABACBCBA)*
ABCBCABB
2-évitable *(symétrique de AABCACAB)*
2-DOL-inévitable *(symétrique de AABCACAB)*
3-DOL-évitable *(contient AA)*
ABCBCABBA
2-DOL-évitable *(symétrique de ABBACBCBA)*
ABCBCABBC
2-DOL-évitable *(symétrique de ABBCABABC)*
ABCBCABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(symétrique de ABCABABC)*
3-DOL-évitable *(contient AA)*
ABCBCABCA
2-DOL-évitable *(contient ABBABA)*
ABCBCABCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCBCABCBB
2-DOL-évitable *(symétrique de AABACBABAC)*
ABCBCABCBC
2-DOL-évitable *(contient ABBABB)*
ABCBCABCC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCBCAC
2-DOL-inévitable *(symétrique de ABACACB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCBCACA
2-évitable *(symétrique de ABABCBCA)*
2-DOL-inévitable *(symétrique de ABABCBCA)*
3-DOL-évitable *(contient AA)*

ABCBCACAB
2-DOL-évitable *($f = (ab, baa)$)*

ABCBCACB
2-évitable *($f = (abc, ac, b)$ et $g = (abbb, bbaa, aba)$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBCACBB
2-DOL-évitable *(symétrique de $AABCBABAC$)*

ABCBCACBC
2-DOL-évitable *(symétrique de $ABACABABC$)*

ABCBCACC
2-DOL-évitabilité inconnue
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBCACCA
2-DOL-évitable *(symétrique de $ABBABCBCA$)*

ABCBCACCB
2-DOL-évitable *(symétrique de $ABBCBABAC$)*

ABCBCC
2-inévitable *(divise $ABABB$)*
3-DOL-évitable *(contient AA)*

ABCBCCA
2-DOL-inévitable *(symétrique de $ABBCBCA$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBCCAA
2-évitable *(symétrique de $AABBCBCA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBCCAAC
2-DOL-évitable *(symétrique de $ABBAACACB$)*

ABCBCCAB
2-DOL-inévitable *(divise $ABBCABABBCABB$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBCCABC
2-évitable *(symétrique de $ABCAABABC$)*
2-DOL-inévitable *(divise $ABBCABABBCABB$)*
3-DOL-évitable *(contient AA)*

ABCBCCABCB
2-DOL-évitable *(symétrique de $ABACBBABAC$)*

ABCBCCABCC
2-évitable *(contient $ABCBCCABC$)*
2-DOL-inévitable *(divise $ABBCABABBCABB$)*
3-DOL-évitable *(contient AA)*

ABCBCCABCCA
2-DOL-évitable *(symétrique de $ABBCABBCBCA$)*

ABCBCCABCCC
2-DOL-évitable *(contient $ABBCABCC$)*

ABCBCCAC
2-évitable *(symétrique de $ABAACACB$)*
2-DOL-inévitable *(divise $ABBCABABBCB$)*
3-DOL-évitable *(contient AA)*

ABCBCCACA
2-évitable *(contient $ABCBCCAC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCBCCACAB
2-DOL-évitable *($f = (ab, aaa)$)*

ABCBCCACB
2-DOL-évitable *(symétrique de $ABCBBABAC$)*

ABCBCCACC
2-DOL-évitable *(symétrique de $AABAACACB$)*

ABCC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*

ABCCA
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*

ABCCAA
2-inévitable *(divise $AABB$)*
3-DOL-évitable *(contient AA)*

ABCCAAB
2-DOL-inévitable *(symétrique de $ABBCCAB$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCAABA
2-DOL-évitable *(symétrique de $ABAACCBA$)*

ABCCAABC
2-évitable *($f = (abc, ac, b)$ et $g = (aaab, abab, bb)$)*
2-DOL-inévitable *(divise $ABBCAABBCAB$)*
3-DOL-évitable *(contient AA)*

ABCCAABCA
2-évitable *(contient $ABCCAABC$)*
2-DOL-inévitable *(divise $ABBCAABBCAB$)*
3-DOL-évitable *(contient AA)*

ABCCAABCAA
2-DOL-évitable *(contient $ABBCABCC$)*

ABCCAABCC
2-évitable *(contient $ABCCAABC$)*
2-DOL-inévitable *(divise $ABCAABBCAAB$)*
3-DOL-évitable *(contient AA)*

ABCCAABCCA
2-évitable *(contient $ABCCAABC$)*
2-DOL-inévitable *(divise $ABCAABBCAAB$)*
3-DOL-évitable *(contient AA)*

ABCCAABCCAA
2-DOL-évitable *(contient $ABABA$)*

ABCCAABCCAC
2-évitable *(contient $ABCCAABC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCAABCCACC
2-DOL-évitable *(contient $ABBCABBCBB$)*

ABCCAAC
2-évitable *(contient $AABBA$)*
2-DOL-inévitable *(divise $AABBA$)*
3-DOL-évitable *(contient AA)*

ABCCAACB
2-DOL-évitable *($f = (aab, abb)$)*

ABCCAB
2-inévitable *(longueur maximale : 107)*
3-DOL-évitable *(contient AA)*

ABCCABA
2-DOL-inévitable *(divise $ABACABBACAA$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCABAA
2-DOL-inévitable *(symétrique de $AABACCBA$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCABAAC
2-DOL-évitable *(symétrique de $ABBCBAACB$)*

ABCCABAB
2-évitable *(contient $ABBAA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCABABB
2-évitable *(contient $ABBAA$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCABABC
2-DOL-évitable *(contient $ABBAAB$)*

ABCCABAC
2-DOL-évitable *(symétrique de $ABCBAACB$)*

ABCCABACA
2-DOL-évitable *(symétrique de $ABACABBCA$)*

ABCCABACC
2-DOL-évitable *(symétrique de $AABCBAACB$)*

ABCCABB
2-DOL-inévitable *(symétrique de $AABCCAB$)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCABBA
2-DOL-évitable *(symétrique de $ABBACCBA$)*

ABCCABBC
2-évitable *(symétrique de $ABBCAABC$)*
2-DOL-inévitable *(divise $ABCCABBCC$)*
3-DOL-évitable *(contient AA)*

ABCCABBCA
2-évitable *(contient $ABCCABBC$)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCABBCB
2-DOL-évitable *(symétrique de $ABAACBBAC$)*

ABCCABBCC
2-évitable *(contient $ABCCABBC$)*
2-DOL-inévitable *(symétrique de $AABBCAABC$)*
3-DOL-évitable *(contient AA)*

ABCCABBCCA
2-DOL-évitable *(symétrique de $ABBCCABBCA$)*

ABCCABBCCB
2-DOL-évitable *(contient $ABBACBBC$)*

ABCCABC
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABCAABCACAA$)*
3-DOL-évitable *(contient AA)*

ABCCABCA
2-évitable *(contient $ABBAB$)*
2-DOL-inévitable *(divise $ABACABBACABA$)*
3-DOL-évitable *(contient AA)*

ABCCABCAB
2-DOL-évitable *(contient $ABBABA$)*

ABCCABCAC
2-DOL-évitable *(symétrique de $ABACBAACB$)*

ABCCABCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*

ABCCABCBB
2-DOL-évitable *(symétrique de AABACBBAC)*

ABCCABCBC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*

ABCCABCBCC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABCAABCACAA)*
3-DOL-évitable *(contient AA)*

ABCCABCBCCA
2-DOL-évitable *(symétrique de ABBCBCABBCA)*

ABCCABCBCCB
2-DOL-évitable *(contient ABBABCBBC)*

ABCCABCC
2-DOL-évitable *(contient ABBABB)*

ABCCAC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*

ABCCACA
2-inévitable *(divise AABAB)*
3-DOL-évitable *(contient AA)*

ABCCACAB
2-DOL-inévitable *(divise AABCAABABCAAB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCACABC
2-évitable *(symétrique de ABCACAABC)*
2-DOL-inévitable *(divise AABCAABABCAAB)*
3-DOL-évitable *(contient AA)*

ABCCACABCB
2-DOL-évitable *(symétrique de ABACBCBBAC)*

ABCCACABCC
2-évitable *(contient ABCCACABC)*
2-DOL-inévitable *(divise AABCAABABCAAB)*
3-DOL-évitable *(contient AA)*

ABCCACABCCA
2-évitable *(contient ABCCACABC)*
2-DOL-inévitable *(divise AABCAABABCAAB)*
3-DOL-évitable *(contient AA)*

ABCCACABCCAC
2-DOL-évitable *(symétrique de ABAACBABAACB)*

ABCCACB
2-DOL-inévitable *(symétrique de ABCBBAC)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCACBA
2-DOL-évitable *(symétrique de ABCACCBA)*

ABCCACBB
2-évitable *(symétrique de AABCBBAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCACBBA
2-DOL-évitable *(symétrique de ABBCACCBA)*

ABCCACBBC
2-DOL-évitable *(symétrique de ABBACAABC)*

ABCCACBC
2-évitable *(symétrique de ABACAABC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCACBCA
2-DOL-évitable *(symétrique de ABCBABBCA)*

ABCCACBCB
2-DOL-évitable *(symétrique de ABABCBBAC)*

ABCCACBCC
2-DOL-évitable *(symétrique de AABACAABC)*

ABCCACC
2-inévitable *(divise AA)*
3-DOL-évitable *(contient AA)*

ABCCACCB
2-DOL-évitable *(symétrique de ABBCBBAC)*

ABCCB
2-inévitable *(divise ABBA)*
3-DOL-évitable *(contient AA)*

ABCCBA
2-inévitable *(longueur maximale: 21)*
3-DOL-évitable *(contient AA)*

ABCCBAA
2-DOL-inévitable *(symétrique de AABCCBA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBAAB
2-évitable *(symétrique de ABBACCAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBAABA
2-évitable *(contient ABCCBAAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBAABC
2-DOL-évitable *($f = (aab, abb)$)*

ABCCBAAC
2-DOL-évitable *(symétrique de ABBCAACB)*

ABCCBAB
2-DOL-inévitable *(symétrique de ABACCAB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBABA
2-DOL-évitable *(symétrique de ABABCCBA)*

ABCCBABB
2-évitable *(symétrique de AABACCAB)*
2-DOL-inévitable *(divise ABACABBACAA)*
3-DOL-évitable *(contient AA)*

ABCCBABBA
2-évitable *(contient ABCCBABB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBABBAB
2-évitable *(contient ABCCBABB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBABBABC
2-DOL-évitable *(contient ABBCACAB)*

ABCCBABBC
2-DOL-évitable *(symétrique de ABBCBAABC)*

ABCCBABC
2-DOL-inévitable *(divise ABACABBACABA)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBABCA
2-DOL-évitable *(symétrique de ABCACBBCA)*

ABCCBABCB
2-évitable *(contient ABBCABC)*
2-DOL-inévitable *(divise ABACABBACABA)*
3-DOL-évitable *(contient AA)*

ABCCBABCBA
2-DOL-évitable *(symétrique de ABCBABCCBA)*

ABCCBABCBB
2-DOL-évitable *(contient ABBCABCC)*

ABCCBABCC
2-évitable *(symétrique de AABCBAABC)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*

ABCCBABCCB
2-évitable *(contient ABCCBABCC)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*

ABCCBABCCBA
2-DOL-évitable *(contient ABABA)*

ABCCBABCCBB
2-évitable *(contient ABCCBABCC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBABCCBC
2-évitable *(contient ABCCBABCC)*
2-DOL-inévitable *(divise ABCBBCABCBBCB)*
3-DOL-évitable *(contient AA)*

ABCCBABCCBCC
2-DOL-évitable *(contient ABBCABBCBB)*

ABCCBAC
2-DOL-inévitable *(symétrique de ABCAACB)*
2-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBACA
2-DOL-évitable *(symétrique de ABACBBCA)*

ABCCBACB
2-évitable *(symétrique de ABCABBAC)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*

ABCCBACBA
2-DOL-évitable *(symétrique de ABCABCCBA)*

ABCCBACBB
2-DOL-évitable *(symétrique de AABCABBAC)*

ABCCBACBC
2-DOL-évitable *(symétrique de ABACBAABC)*

ABCCBACC
2-DOL-évitable *(symétrique de AABCAACB)*

ABCCBB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(divise ABBAA)*
3-DOL-évitable *(contient AA)*

ABCCBBA
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de ABBCCBA)*
3-DOL-évitable *(contient AA)*
ABCCBBAA
2-DOL-évitable *(symétrique de AABBCCBA)*
ABCCBBAB
2-évitable *(contient ABBAA)*
2-DOL-inévitable *(symétrique de ABAACCAB)*
3-DOL-évitable *(contient AA)*
ABCCBBABA
2-DOL-évitable *(symétrique de ABABBCCBA)*
ABCCBBABB
2-DOL-évitable *(symétrique de AABAACCAB)*
ABCCBBABC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCCBBABCB
2-DOL-évitable *(symétrique de ABACAABBAC)*
ABCCBBABCC
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCCBBABCCB
2-évitable *(contient ABBAA)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCCBBABCCBB
2-DOL-évitable *(contient ABBABB)*
ABCCBBAC
2-DOL-évitable *(symétrique de ABCCAACB)*
ABCCBC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABAABA)*
3-DOL-évitable *(contient AA)*
ABCCBCA
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABCCBCAC)*
3-DOL-évitable *(contient AA)*
ABCCBCAA
2-évitable *(contient ABBAB)*
2-DOL-évitabilité inconnue
3-DOL-évitable *(contient AA)*
ABCCBCAAC
2-DOL-évitable *(symétrique de ABBACAACB)*
ABCCBCAB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(symétrique de ABCACCAB)*
3-DOL-évitable *(contient AA)*
ABCCBCABA
2-DOL-évitable *(symétrique de ABACBCCBA)*
ABCCBCABC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACABAABACABA)*
3-DOL-évitable *(contient AA)*
ABCCBCABCA
2-DOL-évitable *(symétrique de ABCABCBBCA)*
ABCCBCABCC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCCBCABCCB
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(divise ABACAACABACAAC)*
3-DOL-évitable *(contient AA)*
ABCCBCABCCBB
2-DOL-évitable *(contient ABBCBABBCC)*
ABCCBCABCCBC
2-DOL-évitable *(symétrique de ABAABCABAABC)*
ABCCBCAC
2-évitable *(contient ABBAB)*
2-DOL-inévitable *(symétrique de ABACAACB)*
3-DOL-évitable *(contient AA)*
ABCCBCACA
2-DOL-évitable *(symétrique de ABABCBBCA)*
ABCCBCACB
2-DOL-évitable *(symétrique de ABCBABBAC)*
ABCCBCACC
2-DOL-évitable *(symétrique de AABACAACB)*

Bibliographie

[1] S. I. Adian, *The Burnside Problem and Identities in Groups*, Springer-Verlag, 1979.

[2] J.-P. Allouche et J. Shallit, The ring of k-regular sequences, *Theoret. Comput. Sci.* **98** (1992), 163–197.

[3] S. E. Aršon, Démonstration de l'existence des suites asymétriques infinies, *Mat. Sb.* **2(44)** (1937), 769–777. En russe, résumé en français.

[4] K. A. Baker, Some problems on avoidability. Exposé au LITP, 1992.

[5] K. A. Baker, G. F. McNulty et W. Taylor, Growth problems for avoidable words, *Theoret. Comput. Sci.* **69** (1989), 319–345.

[6] D. R. Bean, A. Ehrenfeucht et G. F. McNulty, Avoidable patterns in strings of symbols, *Pacific J. of Math.* **85** (1979), 261–294.

[7] J. Berstel, Sur les mots sans carré définis par un morphisme, in *ICALP '79*, H. A. Maurer (éd.), Graz, 1979, pp. 16–25, *Lect. Notes Comp. Sci.* n° 71, Springer-Verlag.

[8] J. Berstel, Some recent results on square-free words, in *STACS '84*, M. Fontet et K. Mehlhorn (éd.), pp. 14–25, *Lect. Notes Comp. Sci.* n° 166, Springer-Verlag, 1984.

[9] J. Berstel et D. Perrin, *Theory of Codes*, Academic Press, 1985.

[10] F.-J. Brandenburg, Uniformely growing k-th power-free homomorphisms, *Theoret. Comput. Sci.* **23** (1983), 69–82.

[11] S. Brlek, Enumeration of factors in the Thue-Morse word, *Discr. Appl. Math.* **24** (1989), 83–96.

[12] A. Carpi, On representation systems, Preprint n° I.C.-R.I. 104/93, Istituto di Cibernetica, Arco Felice (NA), Italia, 1993.

[13] A. Carpi, Overlap-free words and finite automata, *Theoret. Comput. Sci.* **115** (1993), 243–260.

[14] A. Carpi, Repetitions in the Kolakovski sequence, *Bull. EATCS* **50** (1993), 194–196.

[15] J. Cassaigne, Unavoidable binary patterns, *Acta Inf.* **30** (1993), 385–395.

[16] M. Crochemore, Linear searching for a square in a word, *Bull. EATCS* **24** (oct. 1984), 66–72.

[17] J. D. Currie, Open problems in pattern avoidance, *Amer. Math. Monthly* **100** (1993), 790–793.

[18] J. D. Currie et A. Bendor-Samuel, Words without near-repetitions, *Canad. Math. Bull.* **35**,2 (1992), 161–166.

[19] A. de Luca et S. Varricchio, Some combinatorial properties of the Thue-Morse sequence and a problem in semigroups, *Theoret. Comput. Sci.* **63** (1989), 333–348.

[20] R. Dean, A sequence without repeats on x, x^{-1}, y, y^{-1}, *Amer. Math. Monthly* **72** (1965), 383–385.

[21] F. Dejean, Sur un théorème de Thue, *J. Combin. Th. A* **13** (1972), 90–99.

[22] F. M. Dekking, On repetitions of blocks in binary sequences, *J. Combin. Th. A* **20** (1976), 292–299.

[23] F. M. Dekking, Strongly non-repetitive sequences and progression-free sets, *J. Combin. Th. A* **27** (1979), 181–185.

[24] P. Dumas, *Récurrences mahlériennes, suites automatiques, études asymptotiques*, Thèse de doctorat, Université Bordeaux I, 1993.

[25] A. Ehrenfeucht et G. Rozenberg, On the structure of polynomially bounded DOL systems, *Fund. Inform.* **2** (1979), 187–197.

[26] A. Ehrenfeucht et G. Rozenberg, Repetition of subwords in DOL languages, *Inform. and Control* **59** (1983), 13–35.

[27] R. C. Entringer, D. E. Jackson et J. A. Schatz, On nonrepetitive sequences, *J. Combin. Th. A* **16** (1974), 159–164.

[28] P. Erdös, Some unsolved problems, *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **6** (1961), 221–254.

[29] A. A. EVDOKIMOV, Strongly asymmetric sequences generated by a finite number of symbols, *Soviet Math. Dokl.* **9** (1968), 536–539. Traduction anglaise.

[30] P. GORALČIK, Un résultat de Mme Mel'ničuk. Exposé au LITP, 1992.

[31] P. GORALČIK ET T. VANIČEK, Binary patterns in binary words, *Inter. J. Algebra Comput.* **1**,3 (1991), 387–391.

[32] D. HAWKINS ET W. MIENTKA, On sequences which contain no repetitions, *Math. Student* **24** (1956), 185–187.

[33] S. ISTRAIL, On irreducible languages and nonrational numbers, *Bull. Math. Soc. Sci. Math. R. S. Roumanie* **21** (1977), 301–308.

[34] V. KERÄNEN, Abelian squares are avoidable on 4 letters, in *ICALP '92*, W. Kuich (éd.), Wien, 1992, pp. 41–52, *Lect. Notes Comp. Sci.* n° 623, Springer-Verlag.

[35] A. J. KFOURY, A linear time algorithm to decide whether a binary word contains an overlap, *Theoret. Inform. Appl.* **22** (1988), 135–145.

[36] Y. KOBAYASHI, Enumeration of irreducible binary words, *Discr. Appl. Math.* **20** (1988), 221–232.

[37] J. LEECH, A problem on strings of beads, *Math. Gazette* **41** (1957), 277–278.

[38] A. LEPISTÖ, Repetitions in Kolakovski sequence. Preprint, 1993.

[39] M. LOTHAIRE, *Combinatorics on Words*, *Encyclopedia of Mathematics and its Applications* vol. 17, Addison-Wesley, Reading, Mass., 1983.

[40] F. MIGNOSI ET P. SÉÉBOLD, If a DOL language is k-power-free then it is circular, in *ICALP '93*, Lund, 1993, *Lect. Notes Comp. Sci.* n° 700, Springer-Verlag.

[41] M. MORSE ET G. A. HEDLUND, Unending chess, symbolic dynamics, and a problem in semigroups, *Duke Math. J.* **11** (1944), 1–7.

[42] B. MOSSÉ, Puissances de mots et reconnaissabilité des points fixes d'une substitution, *Theoret. Comput. Sci.* **99** (1992), 327–334.

[43] J. MOULIN-OLLAGNIER, Preuve de la conjecture de Dejean pour des alphabets à 5, 6, 7, 8, 9 lettres, Prépublication du département de mathématique et informatique n° 89-4, Université Paris-Nord et GRECO de Calcul Formel, 1989.

[44] J. NÉRAUD, Matching a two variable pattern, rapport LITP n° 92-95, 1992.

[45] J. NÉRAUD, Equations in words : an algorithmic contribution, *Bulletin de la Société Mathématique de Belgique*, 1993. À paraître.

[46] J. NÉRAUD, New algorithms for detecting morphic images of a word, in *MFCS' 93*, *Lect. Notes Comp. Sci.* n° 711, Springer-Verlag, 1993.

[47] S. NILGENS, *3-vermeidbare Muster über dem ternären Alphabet*, Diplomarbeit, Fachbereich Informatik, Universität Frankfurt, fév. 1991.

[48] J.-J. PANSIOT, À propos d'une conjecture de F. Dejean sur les répétitions dans les mots, *Discr. Appl. Math.* **7** (1984), 297–311.

[49] P. A. B. PLEASANTS, Non-repetitive sequences, *Proc. Cambridge Phil. Soc* **68** (1970), 267–274.

[50] A. RESTIVO ET S. SALEMI, Overlap-free words on two symbols, in *Automata on Infinite Words*, M. Nivat et D. Perrin (éd.), pp. 198–206, *Lect. Notes Comp. Sci.* n° 192, Springer-Verlag, 1985.

[51] L. ROSAZ, *Unavoidable sets of words*, thèse de doctorat, Université Paris 7, 1993.

[52] P. ROTH, l-occurences of avoidable patterns, in *STACS '91*, C. Choffrut et M. Jantzen (éd.), Hamburg, 1991, *Lect. Notes Comp. Sci.* n° 480, Springer-Verlag.

[53] P. ROTH, Every binary pattern of length six is avoidable on the two-letter alphabet, *Acta Inf.* **29** (1992), 95–106.

[54] G. ROZENBERG ET A. SALOMAA, *The Mathematical Theory of L-Systems*, Academic Press, 1980.

[55] A. SALOMAA, *Jewels of Formal Language Theory*, Computer Science Press, Washington, D.C., 1981.

[56] U. SCHMIDT, Avoidable patterns on two letters, *Theoret. Comput. Sci.* **63** (1989), 1–17.

[57] R. O. SHELTON, Aperiodic words on three symbols I, *J. Reine Angew. Math.* **321** (1981), 195–209.

[58] R. O. SHELTON ET R. P. SONI, Aperiodic words on three symbols II, *J. Reine Angew. Math.* **327** (1981), 1–11.

[59] R. O. Shelton et R. P. Soni, Aperiodic words on three symbols III, *J. Reine Angew. Math.* **330** (1982), 44–52.

[60] H. J. Shyr, A strongly primitive word of arbitrary length and its applications, *Inter. J. Comput. Math., Section A* **6** (1977), 165–170.

[61] A. Thue, Über unendliche Zeichenreihen, *Norske Vid. Selsk. Skr., I. Mat. Nat. Kl., Christiana* **7** (1906), 1–22.

[62] A. Thue, Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen, *Norske Vid. Selsk. Skr., I. Mat. Nat. Kl., Christiana* **1** (1912), 1–67.

[63] T. Vaniček, Mots inévitables, Thèse, Faculté de mathématiques et de physique, Université Charles, 1989. En tchèque.

[64] T. Zech, Wiederholungsfreie Folgen, *Z. angew. Math. Mech.* **38** (1958), 206–209.

[65] A. I. Zimin, Blocking sets of terms, *Math. USSR Sbornik* **47**,2 (1984), 353–364. Traduction anglaise.

Notations

Index

Tables et figures

Table des matières