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 $\alpha^{\text{DOL}}(p)$ et $\alpha^{\text{HDOL}}(p)$: $\alpha^{\text{DOL}}(p)$ 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 $\alpha^{\text{HDOL}}(p)$.

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 $\text{DOL}(\Sigma_k) \subset \text{HDOL}(\Sigma_k, \Sigma_k) \subset \text{HDOL}(\Sigma_k) \subset \mathscr{P}(\Sigma_k^*)$. ■

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, $\text{DOL}(\Sigma_k) \subset \text{DOL}(\Sigma_{k+1})$, $\text{HDOL}(\Sigma_k) \subset \text{HDOL}(\Sigma_{k+1})$ et $\text{HDOL}(\Sigma_k, \Sigma_{k'}) \subset \text{HDOL}(\Sigma_l, \Sigma_{l'})$. ■

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 $w = a_1 a_2 \ldots a_m$, 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 $\Sigma = \{a, b, c\}$:

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

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).

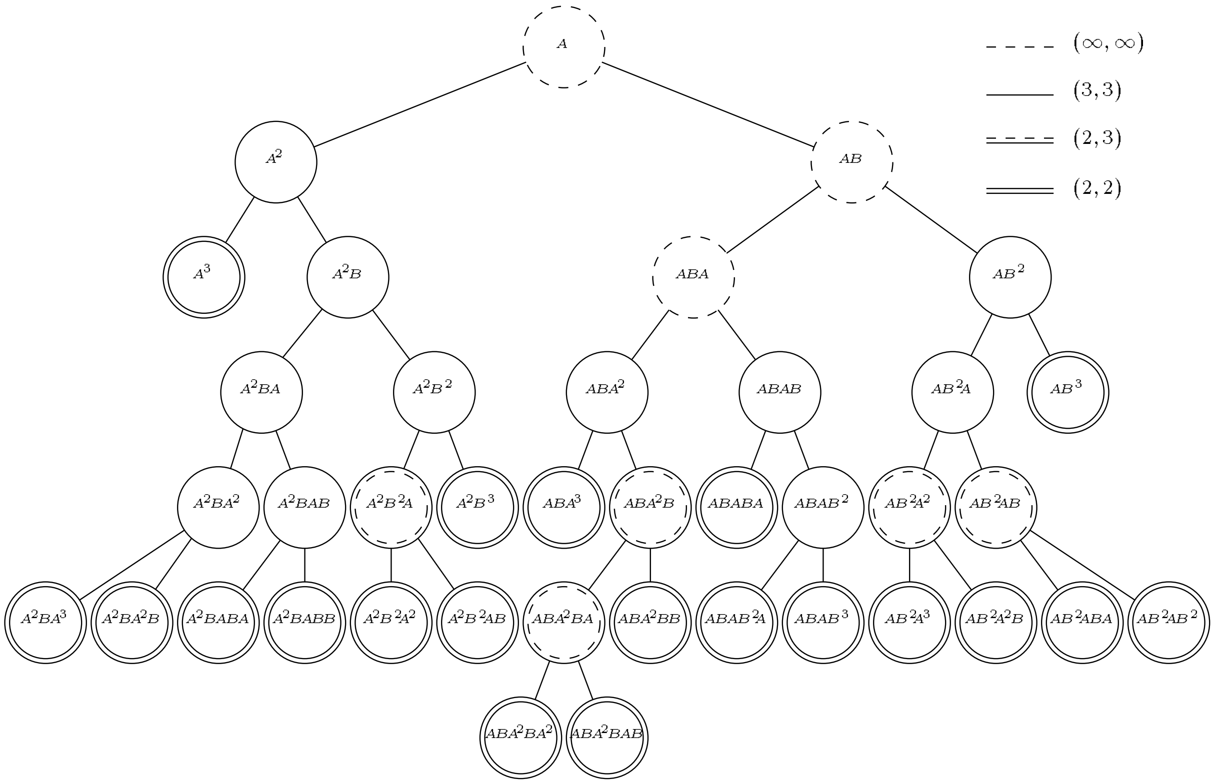

Fig. 3.1 - *Classification des motifs binaires selon* $(\alpha(p), \alpha^{DOL}(p))$