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 :

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

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 $w = w_1w_2$ et

$$\begin{array}{l} \forall v_1 \in \Sigma'^*, \forall v_2 \in \Sigma'^*, \forall u \in U, \exists u_1 \in U, \exists u_2 \in U, \\ \quad [v_1wv_2 = h(u) \Longrightarrow (u = u_1u_2 \text{ et } v_1w_1 = h(u_1) \text{ et } w_2v_2 = h(u_2))] \end{array}$$

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

$$\forall v_1 \in \Sigma'^*, \forall v_2 \in \Sigma'^*, [v_1wv_2 \in h(U) \Longrightarrow (v_1w_1 \in h(U) \text{ et } w_2v_2 \in h(U))]$$

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

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

(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 $i_1, \ldots, i_s$ 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

$$h(w) = v_1 u'_0 \varphi(Y_1) u'_1 \varphi(Y_2) u'_2 \ldots \varphi(Y_s) u'_s v_2 \ .$$

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

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

où $\psi \colon \Delta' \to \Sigma^+$ est le morphisme défini par $\psi(Y) = c_{\varphi(Y)}$. À cause des points de synchronisation $d_j|_U h(\psi(Y_j))|_U e_j$, les $s+1$ mots $v_1 u'_0 d_1$, $e_j u'_j d_{j+1}$ (pour $1 \leqslant j \leqslant s-1$) et $e_s u'_s v_2$ sont dans $h(U)$. Nous pouvons donc appliquer h^{-1}, qui est bien défini sur $h(U)$ puisque h est injectif sur U :

$$w = h^{-1}(v_1 u'_0 d_1) \psi(Y_1) h^{-1}(e_1 u'_1 d_2) \ldots \psi(Y_s) h^{-1}(e_s u'_s v_2)$$

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

$$q = h^{-1}(v_1 u'_0 d_1) Y_1 h^{-1}(e_1 u'_1 d_2) \ldots Y_s h^{-1}(e_s u'_s v_2) \ .$$

Si $w' \in h^{-1}([p]_{\Sigma'})$ 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

$$w' = v'_1 h^{-1}(v_1 u'_0 d_1) \psi'(Y_1) h^{-1}(e_1 u'_1 d_2) \ldots \psi'(Y_s) h^{-1}(e_s u'_s v_2) v'_2$$

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

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

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,

$$\begin{aligned}
|q|_\Delta &= s \leqslant r = |p|_\Delta \\
\|q\|_\Sigma &= \left\| h^{-1}(v_1 u'_0 d_1 e_1 u'_1 d_2 e_2 u'_2 \ldots d_s e_s u'_s v_2) \right\|_\Sigma \\
&\leqslant \frac{1}{m} \|v_1 u'_0 d_1 e_1 u'_1 d_2 e_2 u'_2 \ldots d_s e_s u'_s v_2\|_{\Sigma'} \\
&\leqslant \frac{1}{m} \left(\|v_1 v_2\|_{\Sigma'} + \|d_1 e_1 d_2 e_2 \ldots d_s e_s\|_{\Sigma'} + \|u_0 u_1 \ldots u_r\|_{\Sigma'} + \sum_{\substack{1 \leqslant i \leqslant r \\ X_i \notin \Delta'}} \|\varphi(X_i)\|_{\Sigma'} \right) \\
&\leqslant \frac{1}{m} \left(2M + 2sL + \|p\|_{\Sigma'} + (r-s)2L\right)
\end{aligned}$$

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 $Q_j = h^{-j}\left(g^{-1}(p)\right)$, 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

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

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
$$\beta = \max\left(\alpha, \frac{2M + 2L\,|p|_\Delta}{m-1}\right) \ .$$
Prouvons par récurrence sur j que
$$\forall j \in \mathbb{N}, \forall q \in Q_j, |q|_\Delta \leqslant |p|_\Delta \ \text{ et } \ |q|_\Sigma \leqslant \beta \ .$$
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 $q \in Q_{j+1} = h^{-1}(Q_j)$:
$$|q|_\Delta \leqslant |p|_\Delta \ \text{ et } \ |q|_\Sigma \leqslant \frac{2M}{m} + \frac{1}{m}\beta + \frac{2L}{m}\,|p|_\Delta \leqslant \beta \ .$$
Par conséquent,
$$\forall q \in Q, |q|_\Delta \leqslant |p|_\Delta \ \text{ et } \ |q|_\Sigma \leqslant \beta \ ,$$
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
$$\|h(x)\| \geqslant \|y\| + \|z\| = 2 - \frac{n_y + n_z}{2n} \geqslant 2 - \frac{2(n-1)}{2n} = 1 + \frac{1}{n} \ .$$
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 $w_0 \in h^{-j}\left(g^{-1}\left([p]_{\Sigma'}\right)\right)$. 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

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

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

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

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*) :

$$|\mu(w)|,\ |\mu(w)|a,\ |\mu(w)|ab,\ c|\mu(w)|,\ c|\mu(w)|a,$$
$$c|\mu(w)|ab,\ bc|\mu(w)|,\ bc|\mu(w)|a,\ bc|\mu(w)|ab.$$

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 :

$$|a|a,\ bb,\ c|c|,\ |ab|ab,\ bc|bc|,$$
$$c|ac|a,\ bc|abc|a,\ c|abc|ab,\ bc|abbc|ab,$$
$$|A|A|,\ |A|a|A|a,\ |A|ab|A|ab,\ c|A|c|A|,\ c|A|ac|A|a,$$
$$c|A|abc|A|ab,\ cb|A|bc|A|,\ bc|A|abc|A|a,\ bc|A|abbc|A|ab.$$

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) :

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

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 :

$$a|ab|aa,\ abbba,\ ac|bc|a,\ a|abb|aba,\ abc|bbc|a,$$
$$ac|abc|aa,\ abc|abbc|aa,\ ac|abbc|aba,\ abc|abbbc|aba,$$
$$a|A|b|A|a,\ a|A|ab|A|aa,\ a|A|abb|A|aba,\ ac|A|bc|A|a,\ ac|A|abc|A|aa,$$
$$ac|A|abbc|A|aba,\ abc|A|bbc|A|a,\ abc|A|abbc|A|aa,\ abc|A|abbbc|A|aba\,.$$

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 :

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