

# Figures composées de pixels et monoïde inversif\*

Michel Latteux      Denis Robilliard      David Simplot

## Résumé

Une figure pointée est un ensemble fini de pixels connexes muni d'un point de départ et d'un point d'arrivée permettant de définir un produit de concaténation. Nous montrons que l'ensemble des figures pointées est un monoïde inversif finiment engendré. Nous cherchons à décrire ce monoïde à l'aide d'un monoïde libre comme le préconise Sakarovitch [8]. Ainsi, un mot sur l'alphabet  $\Pi$  peut représenter une figure pointée dans une sémantique proche de celle décrite par Maurer, Rozenberg et Welzl [5] qui décrivent des figures composées de segments. Cette dernière représentation a été largement étudiée et nous étendons, à tout monoïde inversif vérifiant une certaine condition, un résultat de Séébold et Slowinski [10] qui présentent un système de réécriture permettant d'obtenir à partir d'un mot tous les mots qui décrivent la même figure.

## 1 Introduction

Nous nous proposons d'introduire dans cet article une approche théorique de l'étude d'objets utilisés couramment en informatique graphique : les images composées de pixels. Pour notre part, nous nous limitons aux "images" en noir et blanc, mais nous considérons notre grille de pixels comme étant potentiellement infinie. Ainsi, une image peut être vue comme un ensemble fini de pixels noirs, les autres étant considérés comme blancs.

---

\*Ce travail a été supporté en partie par "PRC/GDR Mathématiques et Informatique" et ESPRIT Basic Research Action 6317 ASMICS 2

Received by the editors May 95.

Communicated by M. Boffa.

1991 *Mathematics Subject Classification* : 68Q45.

*Key words and phrases* : Langages de figures, monoïde inversif, système de réécriture.

Tout d’abord, nous décrivons les structures graphiques que sont les *dessins* et les *figures*. Les dessins sont des ensembles finis de pixels “ancrés” dans le plan  $\mathbb{Z}^2$  tandis que les figures sont des classes d’équivalence de dessins modulo une translation. En étudiant les figures, nous nous intéressons surtout à leur forme plutôt qu’à l’endroit où elles se situent dans le plan. Nous nous limitons à l’étude des figures possédant une certaine connexité qui correspond à la 8-connexité usuelle.

Enfin, nous munissons les figures d’un point de départ et d’un point d’arrivée afin de pouvoir les concaténer. Nous parlons alors de *figures pointées*. L’ensemble des figures pointées muni d’un produit de concaténation est un monoïde inversif finiment engendré. Ce résultat nous pousse à nous interroger sur la représentation possible de figures à l’aide de mots. Il est en effet plus pratique de se ramener aux monoïdes libres sur lesquels nous disposons de nombreux outils. Ce type d’approche nous incite à chercher un système générateur du monoïde des figures [8].

Nous montrons que les mots sur l’alphabet  $\Pi = \{u, r, d, l\}$  peuvent servir à représenter toutes les figures en interprétant les lettres des mots comme des ordres donnés à un traceur :  $u$  (de l’anglais up) se déplacer vers le haut,  $r$  (right) à droite,  $d$  (down) vers le bas et  $l$  (left) à gauche. De plus, à chaque déplacement unitaire, on allume le pixel se trouvant à droite du mouvement.

Ceci n’est pas sans rappeler la modélisation proposée par H.A. Maurer, G. Rozenberg et E. Welzl dans [5] où l’on utilise une représentation similaire pour décrire des figures constituées de segments. Cette représentation a été largement étudiée et, notamment, P. Séebold et K. Slowinski donnent dans [10] un système de réécriture qui peut générer tous les mots décrivant une figure à partir d’un mot la décrivant. En fait, ce système de réécriture “engendre” la congruence qui met en relation les mots qui décrivent la même figure.

Dans la recherche d’un tel système de réécriture pour notre représentation des figures composées de pixels, nous montrons qu’un système de réécriture générique proche de celui de [10] peut être utilisé pour les systèmes générateurs de monoïdes inversifs vérifiant certaines conditions. Et plus particulièrement, nous montrons qu’il est adapté au système générateur du monoïde des figures que nous utilisons.

## 2 Préliminaires et notations

Nous supposons le lecteur familier avec la théorie des langages formels et nous ne précisons que quelques notations. Si celles-ci lui semblent toutefois incomplètes, il peut se reporter aux ouvrages de J. Berstel [1] ou de S. Eilenberg [3].

L’ensemble des entiers naturels est noté  $\mathbb{N}$ , l’ensemble des entiers relatifs  $\mathbb{Z}$  et l’ensemble des réels est noté  $\mathbb{R}$ .

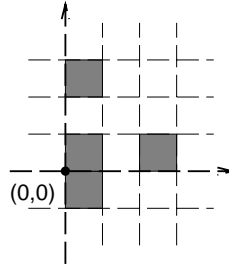
Un *alphabet* est un ensemble (éventuellement infini) et est noté avec une lettre grecque majuscule ( $\Sigma, \Pi, \Psi\dots$ ). Les éléments d’un alphabet sont des *lettres* et sont notés avec des lettres minuscules (on notera par exemple  $a \in \Sigma\dots$ ). Une chaîne finie sur un alphabet est un *mot* et est notée avec une lettre grecque minuscule ( $\alpha, \beta, \omega\dots$ ). L’ensemble  $\Sigma^*$  représente le monoïde libre engendré par l’alphabet  $\Sigma$  et on notera  $\varepsilon$  le mot vide. Pour un mot  $\omega \in \Sigma^*$ ,  $|\omega|$  dénote sa longueur.

Un monoïde  $M$  est *inversif*, si pour tout  $x \in M$ , il existe un et un seul élément noté  $x^{-1}$  tel que  $x.x^{-1}.x = x$  et  $x^{-1}.x.x^{-1} = x^{-1}$  que l’on appelle inverse de  $x$ . Pour

---

La figure ci-dessous illustre le dessin  $p \in \mathcal{D}$  avec :

$$p = \{\text{pix}(0, -1), \text{pix}(0, 0), \text{pix}(0, 2), \text{pix}(2, 0)\}$$




---

Figure 1: Exemple de dessin

plus de renseignements, le lecteur pourra se reporter à l'ouvrage de M. Petrich [6].

Soit  $M$  un monoïde, un *système générateur* de  $M$  est un couple  $(\Sigma, h)$  où  $\Sigma$  est un alphabet et  $h$  un homomorphisme de  $\Sigma^*$  dans  $M$  tels que  $h(\Sigma^*) = M$ . Pour plus de précisions, nous invitons le lecteur à se reporter à l'article de J. Sakarovitch [8]. La *congruence associée à un système générateur*  $(\Sigma, h)$  est la congruence  $\alpha$  définie par :  $\forall \omega, \omega' \in \Sigma^*, \omega \alpha \omega'$  ssi  $h(\omega) = h(\omega')$ .

### 3 Des dessins

Les objets graphiques que nous nous proposons d'étudier sont des ensembles finis de pixels, un pixel étant un carré unitaire  $[i, i + 1] \times [j, j + 1]$  de  $\mathbb{R}^2$  avec  $(i, j) \in \mathbb{Z}^2$ . L'ensemble des pixels est noté  $P$ . Il va de soi qu'un pixel peut être désigné sans ambiguïté par son coin inférieur gauche. On dénote  $\text{pix}(i, j)$  avec  $(i, j) \in \mathbb{Z}^2$  le pixel  $[i, i + 1] \times [j, j + 1]$ .

Un *dessin* est défini comme étant un ensemble fini de pixels (voir figure 1). On note  $\mathcal{D}$  l'ensemble des dessins.

Un *dessin pointé* est un dessin muni de deux points de  $\mathbb{Z}^2$  distingués appelés respectivement *point de départ* et *point d'arrivée*. Ceci nous servira notamment à concaténer deux dessins comme nous le verrons plus loin. Un dessin pointé  $q$  est un triplet  $(p, d, a)$  où  $p$  est un dessin,  $d$  et  $a$  étant deux points de  $\mathbb{Z}^2$ . On dit que  $p$  est la *base* du dessin pointé et est notée  $\text{base}(q)$ . On dit que  $d$  est le *point de départ* du dessin pointé et est noté  $\text{dép}(q)$ . Enfin,  $a$  est appelé *point d'arrivée* du dessin pointé et est noté  $\text{arr}(q)$  (voir figure 2). L'ensemble des dessins pointés est noté  $\mathcal{D}_p$ .

Par abus de langage, on dira qu'un pixel appartient à un dessin pointé alors qu'il faudrait dire que le pixel appartient à la base du dessin pointé.

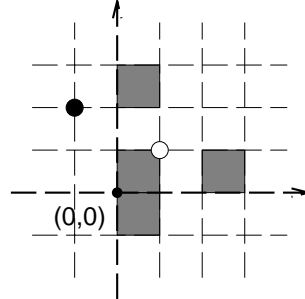
Le *poids* d'un dessin  $p$  est noté  $\|p\|$  et correspond à son cardinal, c'est-à-dire au nombre de pixels qu'il contient. Le poids d'un dessin pointé  $q$  est le poids de sa base. Soient deux dessins (éventuellement pointés)  $p$  et  $q$ , on dit que  $p$  est plus petit que  $q$  si  $\|p\| \leq \|q\|$ .

Les points de départ et d'arrivée nous permettent de concaténer deux dessins pointés. La concaténation de deux dessins pointés  $p$  et  $q$  n'est définie que si  $\text{arr}(p) = \text{dép}(q)$ , et dans ce cas elle correspond à l'union des bases. Un exemple

---

La figure ci-dessous montre le dessin pointé  $q \in \mathcal{D}_p$  avec :

$$q = (p, (1, 1), (-1, 2))$$



Par convention, nous représentons le point de départ par un rond blanc et le point d'arrivée par un rond noir.

---

Figure 2: Exemple de dessin pointé

de concaténation définie est donnée figure 3.

**Définition 3.1** Soient  $p, q \in \mathcal{D}_p$ , la concaténation de  $p$  avec  $q$  notée  $p.q$  n'est définie que si le point d'arrivée de  $p$  correspond avec le point de départ de  $q$  : si  $\text{arr}(p) = \text{dép}(q)$  alors  $p.q = (\text{base}(p) \cup \text{base}(q), \text{dép}(p), \text{arr}(q))$ .

La *connexité* des dessins est une propriété importante. La définition formelle de la connexité est l'occasion pour nous d'introduire la notion d'armature et de voisinage qui, par ailleurs, sont des éléments souvent utiles dans les démonstrations.

L'*armature* d'un dessin  $p$  est l'ensemble des points de la grille constituée par  $\mathbb{Z}^2$  qui appartiennent à un pixel du dessin. L'armature d'un dessin pointé  $q$  est l'armature de sa base.

**Définition 3.2** Soit  $p \in \mathcal{D}$ , l'armature de  $p$  est notée  $\text{Arm}(p)$  et est définie par :

$$\text{Arm}(p) = \bigcup_{\text{pix}(i,j) \in p} \{(i, j), (i + 1, j), (i, j + 1), (i + 1, j + 1)\}$$

Soit  $q \in \mathcal{D}_p$ , l'armature de  $q$  est définie par :

$$\text{Arm}(q) = \text{Arm}(\text{base}(q))$$

Le *voisinage* d'un pixel  $\text{pix}(i, j) \in P$  est désigné par  $\text{Voi}(\text{pix}(i, j))$  et indique l'ensemble des 8 pixels qui l'entourent.

**Définition 3.3** Le voisinage d'un pixel est défini comme suit :

$$\text{Voi}(\text{pix}(i, j)) = \left\{ \text{pix}(k, l) \in P \mid \begin{array}{l} i - 1 \leq k \leq i + 1 \\ j - 1 \leq l \leq j + 1 \end{array} \quad (k, l) \neq (i, j) \right\}$$

On peut maintenant définir formellement la notion de connexité.

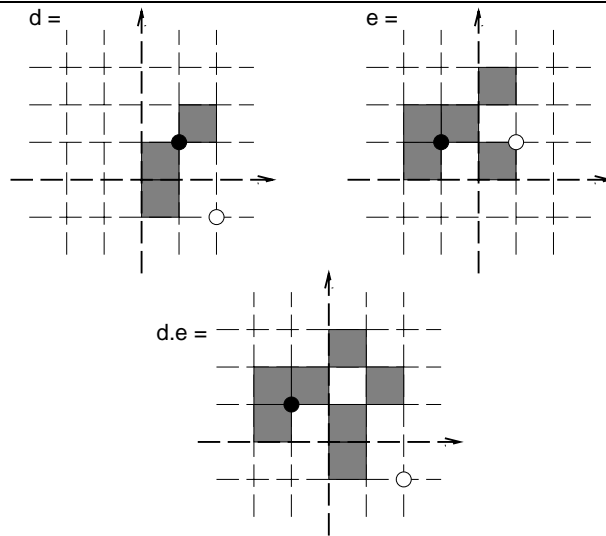


Figure 3: Exemple de concaténation définie

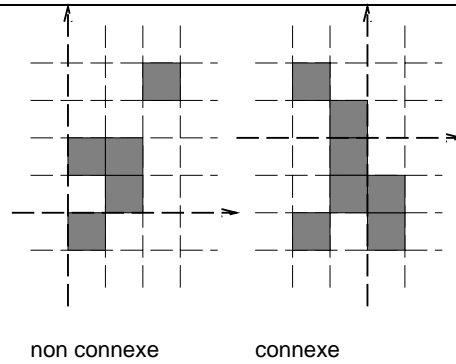


Figure 4: Exemples de connexité

**Définition 3.4** On dit qu'un dessin  $p \in \mathcal{D}$  est *connexe*, si et seulement si :

$$\forall a, b \in p \quad \exists c_1, c_2, \dots, c_n \in p \quad \text{tels que :} \quad \begin{cases} c_1 = a \\ c_n = b \\ \forall 1 \leq i < n \quad c_{i+1} \in \text{Voi}(c_i) \end{cases}$$

Un dessin est connexe si de tout pixel du dessin on peut atteindre tous les pixels du dessin en se déplaçant uniquement d'un pixel aux pixels qui lui sont voisins, c'est-à-dire qui le touchent au moins par un coin (voir figure 4).

On définit aussi la connexité pour les dessins pointés en ajoutant une condition sur la position des points de départ et d'arrivée.

**Définition 3.5**

1. Soit  $p \in \mathcal{D}_p$  un dessin pointé non vide ( $\|p\| \neq 0$ ), on dit que  $p$  est connexe si et seulement si :
  - (a)  $\text{base}(p)$  est connexe ;
  - (b)  $\text{dép}(p) \in \text{Arm}(p)$  ;

(c)  $\text{arr}(p) \in \text{Arm}(p)$ .

2. Soit  $q \in \mathcal{D}_p$  un dessin pointé vide ( $\|q\| = 0$ ), on dit que  $q$  est connexe si et seulement si  $\text{dép}(q) = \text{arr}(q)$ .

L'ensemble des dessins connexes est noté  $\mathcal{D}^c$  tandis que l'ensemble des dessins pointés connexes est noté  $\mathcal{D}_p^c$ .

Nous venons de définir des objets graphiques que nous appelons dessins. Mais plus que l'endroit où ils sont situés dans le plan, c'est leur forme qui nous intéresse.

## 4 Des figures

On définit une *figure* comme une classe d'équivalence de dessins modulo une translation. Plus formellement, nous donnons tout d'abord la définition d'une *translation* sur les pixels et sur les points de  $\mathbb{Z}^2$ . Soient  $a = (x, y) \in \mathbb{Z}^2$  un point et  $(i, j) \in \mathbb{Z}^2$ , le translaté du point  $a$  de  $(i, j)$  est  $\tau_{i,j}(a) = (x + i, y + j)$ . Soient  $b = \text{pix}(x, y) \in \mathbb{P}$  un pixel et  $(i, j) \in \mathbb{Z}^2$ , le translaté du pixel  $b$  de  $(i, j)$  est  $\tau_{i,j}(b) = \text{pix}(x + i, y + j)$ .

Cette notion de translation s'étend naturellement à un ensemble de points ou de pixels. Ce qui nous permet de définir la relation d'équivalence  $\sim$  sur  $\mathcal{D}$  :

$$\forall p, q \in \mathcal{D} \quad p \sim q \Leftrightarrow \exists i, j \in \mathbb{Z} \quad \tau_{i,j}(p) = q$$

De même, on définit  $\simeq$  sur  $\mathcal{D}_p$  :

$$\forall p, q \in \mathcal{D}_p \quad p \simeq q \Leftrightarrow \exists i, j \in \mathbb{Z} \text{ tels que } \begin{cases} \tau_{i,j}(\text{base}(p)) & = \text{base}(q) \\ \tau_{i,j}(\text{dép}(p)) & = \text{dép}(q) \\ \tau_{i,j}(\text{arr}(p)) & = \text{arr}(q) \end{cases}$$

Une *figure* est une classe d'équivalence de  $\sim$  tandis qu'une *figure pointée* est une classe d'équivalence de  $\simeq$ . Nous notons  $\mathcal{F}$  l'ensemble des figures et  $\mathcal{F}_p$  l'ensemble des figures pointées. Soit  $p \in \mathcal{D}$  et  $q \in \mathcal{D}_p$ , on désigne par  $[p]_{\sim}$  figure contenant  $p$  et  $[q]_{\simeq}$  la figure pointée contenant  $q$ .

Comme nous le verrons par la suite, il est plus aisé de raisonner sur un représentant d'une figure que sur la figure elle-même. Aussi lorsqu'il est clair qu'une propriété est conservée par translation, nous utiliserons cette approche. Par exemple, la connexité est une propriété qui est conservée par translation. Aussi, pour montrer qu'une figure est connexe, on considère l'un de ses représentants. De même, pour illustrer une figure par un schéma, nous donnons, en fait, un de ses représentants, et nous omettons les axes.

On note  $\mathcal{F}^c$  l'ensemble des figures connexes et  $\mathcal{F}_p^c$  l'ensemble des figures pointées connexes.

On peut aisément étendre la notion de concaténation de dessins aux figures.

**Définition 4.1** Soient  $f, g \in \mathcal{F}_p$  deux figures pointées, on définit la concaténation de  $f$  avec  $g$  notée  $f.g$  :

$$\forall f, g \in \mathcal{F}_p \quad f.g = [p.q]_{\simeq} \quad p \in f, q \in g, p.q \text{ défini}$$

On voit ainsi que, contrairement au cas des dessins, la concaténation de deux figures est toujours définie. En effet, on peut toujours trouver  $p$  et  $q$  tels que  $p.q$  est défini. Ceci vient du fait que si l'on se fixe un point  $a \in \mathbb{Z}^2$  il existe toujours, pour une figure pointée  $f$ , un représentant  $p$  tel que  $\text{arr}(p) = a$  et un représentant  $q$  tel que  $\text{dép}(q) = a$ . On vérifie ainsi clairement que la concaténation de deux figures est bien unique.

**Proposition 4.2** *L'ensemble des figures pointées muni de la concaténation est un monoïde.*

*Preuve.* On voit facilement que la concaténation est associative. De plus, elle possède bien un élément neutre que nous notons  $f_\varepsilon = [(\emptyset, a, a)]_\simeq$  avec  $a \in \mathbb{Z}^2$ . Nous appelons  $f_\varepsilon$  “la figure pointée vide” ou plus simplement la “figure vide” lorsqu’il n’y a pas d’ambiguïté entre figure et figure pointée. ■

Soit  $F$  un ensemble de figures pointées, on note  $F^*$  le plus petit ensemble contenant  $F$  et  $f_\varepsilon$  et fermé par concaténation.

Dans la suite de l'article, nous restreignons notre étude aux figures connexes, c'est-à-dire aux ensembles  $\mathcal{F}^c$  et  $\mathcal{F}_p^c$ .

## 5 Le monoïde des figures connexes

Nous nous proposons ici de montrer le résultat ci-dessous et d'en déduire un certain nombre de propriétés.

**Théorème 5.1** *L'ensemble des figures pointées connexes muni de la concaténation est un monoïde inversif finiment engendré.*

Afin de montrer ce théorème, nous avons besoin d'un certain nombre de résultats intermédiaires. Notamment, nous devons montrer que la concaténation est bien une opération interne de  $\mathcal{F}_p^c$ .

**Proposition 5.2** *La concaténation de deux figures pointées connexes est une figure pointée connexe.*

*Preuve.* Si  $f = f_\varepsilon$  ou  $g = f_\varepsilon$ , la propriété est évidente. Si  $f \neq f_\varepsilon$  et  $g \neq f_\varepsilon$ , nous raisonnons sur deux représentants  $p \in f$  et  $q \in g$  tels que  $p.q$  est défini. On peut passer de tout pixel de  $p$  à tout autre pixel de  $p$  en n'utilisant que des points de l'armature de  $p$ . De même pour  $q$ . Puisque  $\text{arr}(p) = \text{dép}(q)$  est un point commun aux deux armatures, le dessin  $p.q$  est bien connexe. ■

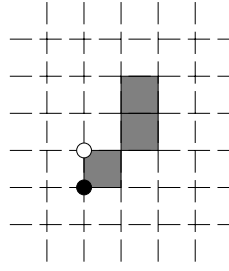


Figure 5: Figure pointée connexe ne pouvant être “décomposée”

On en déduit, avec la proposition 4.2 :

**Corollaire 5.3** *L’ensemble des figures connexes pointées muni de la concaténation est un monoïde.*

On remarque que :

**Fait 5.4** *Tout dessin connexe de poids au moins deux peut être décomposé (au sens de l’union) en deux dessins connexes disjoints non vides.*

Mais à l’opposé, une figure pointée connexe  $f$  de poids au moins deux n’est pas toujours la concaténation de deux figures pointées connexes non vides strictement plus petites. Par exemple, la figure de la figure 5 ne comporte pas ce type de décomposition. Par contre, on peut toujours trouver trois figures pointées connexes strictement plus petites dont  $f$  est la concaténation. Ceci nous est montré par le lemme suivant :

**Lemme 5.5** *Soit  $f \in \mathcal{F}_p^c$ , une figure pointée connexe. Si  $\|f\| \geq 2$ , il existe trois figures pointées connexes  $f_1, f_2$  et  $f_3$  toutes de poids strictement inférieur à  $\|f\|$  tels que  $f = f_1 \cdot f_2 \cdot f_3$ .*

*Preuve.* Raisonnons sur un représentant de  $f$  :

$$f = [(p, d, a)]_{\simeq}$$

Considérons le dessin  $p$ . D’après le fait 5.4, on peut décomposer  $p$  en deux dessins connexes disjoints  $q_1$  et  $q_2$  non vides.

Notons que l’on a  $\|q_1\| < \|p\|$  et  $\|q_2\| < \|p\|$ . De plus, il est clair qu’il existe un point  $e$  qui est à la fois dans l’armature de  $q_1$  et dans l’armature de  $q_2$  car  $p = q_1 \cup q_2$  est connexe :

$$e \in \text{Arm}(q_1) \cap \text{Arm}(q_2)$$

Nous pouvons distinguer 4 cas selon la position de  $d$  et  $a$ .

1. si  $d \in \text{Arm}(q_1)$  et  $a \in \text{Arm}(q_2)$  : on peut considérer les figures pointées suivantes :

$$f_1 = [(q_1, d, e)]_{\simeq} \quad f_2 = [(q_2, e, a)]_{\simeq} \quad f_3 = f_e$$



2. si  $d \in \text{Arm}(q_2)$  et  $a \in \text{Arm}(q_1)$  : cas similaire au précédent.
3. si  $d \in \text{Arm}(q_1)$  et  $a \in \text{Arm}(q_1)$  : on considère les figures pointées :

$$f_1 = [(q_1, d, e)]_{\simeq} \quad f_2 = [(q_2, e, e)]_{\simeq} \quad f_3 = [(q_1, e, a)]_{\simeq}$$

4. si  $d \in \text{Arm}(q_2)$  et  $a \in \text{Arm}(q_2)$  : cas similaire au précédent.

On peut donc toujours trouver  $f_1, f_2$  et  $f_3$  de poids strictement inférieur au poids de  $f$  et avec  $f = f_1 \cdot f_2 \cdot f_3$ . ■

Enfin, avant de montrer que  $(\mathcal{F}_p^c, \cdot)$  est inversif, nous allons avoir besoin de la définition d'un inverse pour les figures pointées :

**Définition 5.6** Soit  $f = [(p, d, a)]_{\simeq}$  une figure pointée, l'inverse de  $f$ , noté  $f^{-1}$  est la figure  $f^{-1} = [(p, a, d)]_{\simeq}$ .

*Preuve.* (théorème 5.1) Le corollaire 5.3 nous indique déjà que  $(\mathcal{F}_p^c, \cdot)$  est un monoïde. Il nous reste donc à montrer qu'il est finiment engendré et inversif.

Nous montrons tout d'abord qu'il est finiment engendré. Considérons l'ensemble  $O$  qui contient toutes les figures pointées connexes de poids 1 :

$$O = \{f \in \mathcal{F}_p^c \mid \|f\| = 1\}$$

Il est clair que  $O$  est fini : il ne contient en effet que 16 figures suivant la position du point de départ et du point d'arrivée. Nous montrons que  $\mathcal{F}_p^c = O^*$ . Pour ceci, il nous suffit que toute figure pointée  $f$  de taille au moins 1 soit dans  $O^*$  (la figure vide étant par définition dans  $O^*$ ).

On raisonne par récurrence sur le poids de  $f$ . Si  $f$  est de poids 1, la propriété est vraie puisque  $f \in O$ . Sinon, si  $\|f\| > 1$ , on suppose la propriété vraie pour toute figure de poids strictement inférieur à  $\|f\|$  et on la montre vraie pour  $f$ . La figure  $f$  vérifie les conditions d'application du lemme 5.5. Aussi, on peut trouver  $f_1, f_2$  et  $f_3$  dans  $\mathcal{F}_p^c$  toutes de poids strictement inférieur à celui de  $f$  et telles que  $f = f_1 \cdot f_2 \cdot f_3$ . Or, par hypothèse de récurrence, on a pour  $i \in \{1, 2, 3\}$ ,  $f_i \in O^*$  et donc  $f \in O^*$ .

Enfin, il nous faut montrer que  $(\mathcal{F}_p^c, \cdot)$  est inversif. Pour ceci, il nous faut montrer que pour toute figure pointée  $f = [(p, d, a)]_{\simeq}$ ,  $f^{-1} = [(p, a, d)]_{\simeq}$  est la seule figure  $g = [(q, a, a')]_{\simeq}$  satisfaisant :

$$f \cdot g \cdot f = f \tag{1}$$

$$g \cdot f \cdot g = g \tag{2}$$

Ce qui est clair, puisque (1) impose que  $a' = d$  et  $q \subseteq p$  et que (2) contraint  $p \subseteq q$ . Il nous suffit maintenant de constater que l'inverse défini en 5.6 associe à une figure connexe pointée une figure connexe pointée. Ceci nous assure l'existence de l'inverse et l'unicité de l'inverse. ■

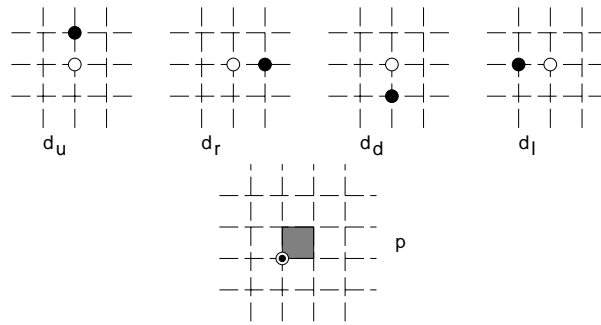


Figure 6: Figures engendrant le monoïde  $(\mathcal{F}_p, \cdot)$

Notons que l'ensemble des figures pointées  $(\mathcal{F}_p, \cdot)$  est aussi un monoïde inversif finiment engendré. Il est engendré par  $A = \{d_u, d_r, d_d, d_l, p\}$  illustré figure 6.

Nous avons vu dans la preuve précédente que le monoïde des figures pointées connexes était engendré par l'ensemble des figures pointées connexes de poids 1 (noté  $O$  dans la preuve). Cet ensemble comporte 16 éléments, il nous a semblé intéressant de chercher à savoir s'il n'existe pas un ensemble de figures plus petit qui génère  $\mathcal{F}_p^c$ .

**Corollaire 5.7** Soit  $\mathcal{U} = \{\text{■}, \text{■}, \text{■}, \text{■}\}$ , on a  $\mathcal{F}_p^c = \mathcal{U}^*$ .

Les symboles du type  $\text{■}, \text{■}, \dots$  désignent sans ambiguïté des figures de  $\mathcal{F}_p^c$ . Par exemple, le symbole  $\text{■}$  représente la figure  $[(\{\text{pix}(0, 0)\}, (0, 1), (1, 1))]_{\simeq}$ .

*Preuve.* Pour montrer ce corollaire, il suffit de montrer que toutes les figures pointées connexes de taille 1 peuvent être obtenues par concaténation des figures de  $\mathcal{U}$ . Nous n'examinons que les cas où le point de départ est dans le coin supérieur gauche, les autres cas étant symétriques. Il y a 4 cas possibles :

1.  $\text{■}$  est bien dans  $\mathcal{U}$
2.  $\text{■} = \text{■} \cdot \text{■}$
3.  $\text{■} = \text{■} \cdot \text{■} \cdot \text{■}$
4.  $\text{■} = \text{■} \cdot \text{■} \cdot \text{■} \cdot \text{■}$

■

Notons que cet ensemble de générateurs de  $\mathcal{F}_p^c$  est de taille minimale. Il n'existe pas d'ensemble de taille strictement inférieure à 4 qui génère  $\mathcal{F}_p^c$ .

Comme dans tout monoïde, nous pouvons distinguer les parties reconnaissables et les parties rationnelles de  $\mathcal{F}_p^c$  [1] :

**Définition 5.8** Une partie  $F$  de  $\mathcal{F}_p^c$  est reconnaissable s'il existe un monoïde fini  $M$ , un homomorphisme  $h$  de  $\mathcal{F}_p^c$  dans  $M$  et une partie  $N$  de  $M$  tels que  $F = h^{-1}(N)$ . Nous notons  $\text{Rec}(\mathcal{F}_p^c)$  l'ensemble des parties reconnaissables de  $\mathcal{F}_p^c$ .

**Définition 5.9** La classe des parties rationnelles de  $\mathcal{F}_p^c$ , notée  $\text{Rat}(\mathcal{F}_p^c)$ , est la plus petite classe des parties de  $\mathcal{F}_p^c$  contenant les parties finies et close par union, concaténation et étoile.

On sait déjà grâce à McKnight (1964) (voir proposition 2.4 p. 57 de [1]), que l'on a  $\text{Rec}(\mathcal{F}_p^c) \subseteq \text{Rat}(\mathcal{F}_p^c)$ . Néanmoins  $\mathcal{F}_p^c$  n'est pas "de Kleene" [9, 7], c'est-à-dire que  $\text{Rat}(\mathcal{F}_p^c) \neq \text{Rec}(\mathcal{F}_p^c)$ . Pour s'en convaincre, il suffit de considérer l'ensemble  $L = \{\blacksquare\}^*$ . L'ensemble  $L$  est évidemment rationnel mais il est facile de montrer qu'il n'est pas reconnaissable.

**Proposition 5.10** Les parties finies de  $\mathcal{F}_p^c$  sont rationnelles et reconnaissables.

*Preuve.* Les parties finies de  $\mathcal{F}_p^c$  étant par définition dans  $\text{Rat}(\mathcal{F}_p^c)$ , nous nous intéressons aux reconnaissables. Les reconnaissables étant clos par union, il nous suffit de montrer qu'un ensemble contenant une unique figure est reconnaissable (l'ensemble vide étant reconnaissable).

Soit  $L = \{f\}$  avec  $f \in \mathcal{F}_p^c$ . Nous montrons que  $L$  est reconnaissable en montrant que le nombre de classe d'équivalence de la congruence à droite induite par  $L$  est fini (voir proposition 12.1 p. 68 de [3]). Notons, pour  $g \in \mathcal{F}_p^c$ ,  $L/g = \{m \in \mathcal{F}_p^c \mid g.m \in L\}$ . Deux figures  $g$  et  $g'$  de  $\mathcal{F}_p^c$  sont en relation par la congruence ssi  $L/g = L/g'$ . Remarquons que si  $\|g\| > \|f\|$ , alors  $L/g = \emptyset$ . Le nombre de figures connexes de poids inférieur ou égal à  $f$  étant fini, le nombre de classes d'équivalence est, a fortiori, fini. ■

Le fait que l'ensemble des figures pointées connexes est un monoïde finiment engendré nous pousse à nous interroger sur ses systèmes générateurs possibles ; c'est-à-dire représenter  $\mathcal{F}_p^c$  par le biais d'un monoïde libre engendré par un alphabet fini [8]. Nous consacrons la suite de l'article à l'étude d'un système générateur de ce monoïde.

## 6 Des mots de figures

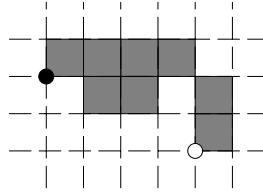
Nous cherchons donc, en étudiant un système générateur de  $(\mathcal{F}_p^c, \cdot)$ , à modéliser les figures, qui sont de manière intrinsèque des objets à deux dimensions, à l'aide de mots. Il est en effet souvent malaisé de travailler sur des objets à deux dimensions alors que les mots sont des objets linéaires que nous savons bien manipuler.

Ici, cette approche nous est facilitée par le fait que  $(\mathcal{F}_p^c, \cdot)$  est un monoïde finiment engendré. Ainsi, nous avons vu dans le corollaire 5.7 qu'il était engendré par l'ensemble  $\mathcal{U}$  composé de 4 figures. Nous nous appuyons donc sur ce résultat pour construire un système générateur du monoïde.

**Proposition 6.1** Soient l'alphabet  $\Pi = \{u, r, d, l\}$  et l'homomorphisme  $\text{FigP} : \Pi^* \rightarrow \mathcal{F}_p^c$  défini par :

$$\begin{aligned} \text{FigP}(u) &= \blacksquare \\ \text{FigP}(r) &= \blacksquare \\ \text{FigP}(d) &= \blacksquare \\ \text{FigP}(l) &= \blacksquare \end{aligned}$$

Le couple  $(\Pi, \text{FigP})$  est un système générateur de  $(\mathcal{F}_p^c, \cdot)$ .

Figure 7: figure pointée décrite par *uulldull*

*Preuve.* Immédiate d'après le corollaire 5.7. ■

Par exemple, le mot *uulldull* décrit la figure pointée illustrée en 7. Le fait que  $\mathcal{F}_p^c$  est inversif, implique qu'il existe, pour une figure pointée non vide, une infinité de mots qui lui correspondent. En effet, pour une figure pointée  $f$  non vide, si l'on considère les mots  $\omega, \omega' \in \Pi^*$  tels que  $\text{FigP}(\omega) = f$  et  $\text{FigP}(\omega') = f^{-1}$ , le mot  $\omega\omega'\omega$  décrit également  $f$ , ainsi que les mots  $(\omega\omega')^n\omega$  avec  $n \in \mathbb{N}$ . Nous notons  $\equiv$  la congruence associée à ce système générateur :

**Définition 6.2** La congruence associée au système générateur  $(\Pi, \text{FigP})$ , notée  $\equiv$ , est définie par :

$$\forall \omega, \omega' \in \Pi^* \quad \omega \equiv \omega' \Leftrightarrow \text{FigP}(\omega) = \text{FigP}(\omega')$$

Notons qu'ainsi nous pouvons considérer le monoïde  $\mathcal{F}_p^c$  comme le quotient de  $\Pi^*$  par la congruence  $\equiv$ . Nous nous proposons donc d'étudier les classes d'équivalence de cette congruence  $\equiv$ .

**Proposition 6.3** Les classes d'équivalence de  $\equiv$  sont des langages rationnels de  $\Pi^*$ .

*Preuve.* Formellement, on veut montrer que l'on a, pour tout mot  $\omega$  sur  $\Pi$  :  $[\omega]_{\equiv} \in \text{Rat}(\Pi^*)$ . Ceci est clair grâce à la proposition 5.10. En effet, il suffit de constater que  $[\omega]_{\equiv}$  est l'image par l'homomorphisme inverse  $\text{FigP}^{-1}$  de l'ensemble fini  $\{\text{FigP}(\omega)\}$ . ■

Cette propriété des classes d'équivalence de  $\equiv$  a pour conséquence directe que pour tout ensemble de figures pointées  $F \subseteq \mathcal{F}_p^c$  engendré par un langage algébrique  $A$  sur  $\Pi$  (c'est-à-dire que l'on a  $A \in \text{Alg}(\Pi^*)$  et  $F = \text{FigP}(A)$ ), le problème de l'appartenance à  $F$  est décidable. En effet, il est facile de voir que l'on peut construire effectivement, pour toute figure pointée  $f \in \mathcal{F}_p^c$ , un automate qui reconnaît  $\text{FigP}^{-1}(f)$ .

Enfin, on peut, à l'aide de ce système générateur, caractériser les parties rationnelles et les parties reconnaissables de  $\mathcal{F}_p^c$  :

**Fait 6.4**

1.  $F \in \text{Rat}(\mathcal{F}_p^c) \Leftrightarrow \exists L \in \text{Rat}(\Pi^*) \quad \text{FigP}(L) = F$  ;
2.  $F \in \text{Rec}(\mathcal{F}_p^c) \Leftrightarrow \text{FigP}^{-1}(F) \in \text{Rat}(\Pi^*)$ .

Cette méthode de représentation de figures à l'aide d'un monoïde libre a déjà été appliquée par H.A. Maurer, G. Rozenberg et E. Welzl dans [5] pour représenter, par des mots, des figures composées de segments unitaires. On peut en effet interpréter les mots sur  $\Pi$  comme une série de commandes données à un traceur. La lettre  $u$  provoque un déplacement unitaire vers le haut (up), la lettre  $r$  un déplacement unitaire vers la droite (right), la lettre  $d$  vers le bas (down) et  $l$  vers la gauche (left). À chaque déplacement, on allume le pixel se trouvant à droite du déplacement.

Dans l'étude de la représentation proposée par H.A. Maurer et al., P. Séébold et K. Slowinski, donnent dans [10] un système de réécriture qui génère tous les mots qui décrivent la même figure. En fait, la relation induite par le système de réécriture est la congruence associée au système générateur qu'ils utilisent. Nous montrons dans la section suivante que des systèmes de réécriture similaires peuvent être utilisés pour des systèmes générateurs de monoïdes inversifs.

## 7 Étude de la congruence associée à un système générateur d'un monoïde inversif

Soit  $M$  un monoïde admettant un système générateur  $(\Sigma, h)$ . Nous notons  $X$  l'ensemble "généralisé" par  $\Sigma$  :  $X = h(\Sigma)$ . On a alors  $M = X^*$ . Nous rappelons que  $X$  peut être infini. Nous notons  $\lambda$  l'élément neutre de la concaténation dans  $M$ . Nous notons  $\asymp$  la congruence associée à  $(\Sigma, h)$ . Nous cherchons un système de réécriture  $S$  qui "engendre" la congruence  $\asymp$ , c'est-à-dire que l'on ait :

$$\forall \omega, \omega' \in \Sigma^* \quad \omega \asymp \omega' \Leftrightarrow \omega \xrightarrow[S]{*} \omega' \text{ et } \omega' \xrightarrow[S]{*} \omega$$

**Définition 7.1** On dit qu'un élément  $x$  de  $M$  est un idempotent ssi  $x.x = x$ . On note  $I_M$  l'ensemble des idempotents de  $M$ .

Dans le cas où  $M$  est inversif, on a les propriétés suivantes :

1.  $\forall x \in I_M \quad x = x^{-1}$  ;
2.  $\forall x \in M \quad x.x^{-1} \in I_M$  ;
3.  $\forall x, y \in I_M \quad x.y = y.x \in I_M$ .

On supposera dans la suite que  $M$  est inversif. Dans le monoïde  $M$ , nous disposons d'un inverse. Il est intéressant de pouvoir bénéficier d'un outil similaire dans le monoïde libre qu'on lui associe.

**Définition 7.2** Une fonction inverse pour  $(\Sigma, h)$  est une fonction  $\text{Inv}$  de  $\Sigma$  dans  $\Sigma^*$  telle que  $\forall a \in \Sigma$ , on a  $h(\text{Inv}(a)) = h(a)^{-1}$ . On étend cette fonction sur les mots, pour tout  $\omega \in \Sigma^*$  :

1. si  $\omega = \varepsilon$ ,  $\text{Inv}(\omega) = \varepsilon$  ;
2. si  $\omega = \omega'a$  avec  $\omega \in \Sigma^*$  et  $a \in \Sigma$ ,  $\text{Inv}(\omega) = \text{Inv}(a)\text{Inv}(\omega')$ .

On note aussi  $\text{Inv}(\omega) = \bar{\omega}$ .

On déduit immédiatement de cette définition les propriétés suivantes :

**Fait 7.3**

1.  $\forall \alpha \in \Sigma^* \quad h(\bar{\alpha}) = h(\alpha)^{-1}$  ;
2.  $\forall \alpha\beta \in \Sigma^* \quad \overline{\alpha\beta} = \bar{\beta}\bar{\alpha}$ .

On considère une fonction inverse et une première approche du problème nous donne le système de réécriture  $R$  défini ci-dessous :

**Définition 7.4** Soient  $M$  un monoïde inversif,  $(\Sigma, h)$  un système générateur de  $M$  et  $\text{Inv}$  une fonction inverse de  $(\Sigma, h)$ , on définit le système de réécriture  $R$  :

$$R = R_1 \cup R_2$$

avec  $\begin{cases} R_1 = \{a \leftrightarrow a\bar{a}a \mid a \in \Sigma\} \\ R_2 = \{\alpha \leftrightarrow \beta \mid \alpha, \beta \in \Sigma^* \text{ et } h(\alpha) = h(\beta) \in I_M\} \end{cases}$

**Proposition 7.5** La relation  $\xrightarrow{*}_R$  coïncide avec la congruence  $\asymp$ .

*Preuve.* On déduit d'abord de  $R$  l'ensemble de règles  $R_3$  :

$$R_3 = \{\alpha \leftrightarrow \alpha\bar{\alpha}\alpha \mid \alpha \in \Sigma^*\}$$

On montre par récurrence sur la longueur de  $\alpha$  que l'on peut déduire la règle  $\alpha \leftrightarrow \alpha\bar{\alpha}\alpha$  de  $R$ . Si  $\alpha = \varepsilon$ , on a  $\alpha = \alpha\bar{\alpha}\alpha$ . Si  $|\alpha| = 1$ , on utilise les règles de  $R_1$ . Sinon, on suppose la propriété vraie pour tout mot de longueur strictement inférieure à celle de  $\alpha$  et on la montre vraie pour  $\alpha$ . On a  $\alpha = \alpha'a$  avec  $\alpha' \in \Sigma^*$  et  $a \in \Sigma$ . On a :

$$\alpha = \alpha'a \xrightarrow{\text{réc.}} \alpha'\bar{\alpha}'\alpha'a \xrightarrow{R_1} \alpha'\bar{\alpha}'\alpha'a\bar{a}a \xrightarrow{R_2} \alpha'a\bar{\alpha}\bar{\alpha}'\alpha'a = \alpha\bar{\alpha}\alpha$$

L'application des règles de  $R_2$  nous permet en effet d'intervertir deux mots représentant des idempotents. Ici, en l'occurrence, il s'agit de  $\bar{\alpha}'\alpha'$  et  $a\bar{a}$  que l'on permute.

Une fois cet ensemble de règles déduit, nous montrons que pour deux mots  $\omega, \omega' \in \Sigma^*$ , tels que  $\omega \asymp \omega'$  alors, on a  $\omega \xrightarrow{*}_R \omega'$  :

$$\omega \xrightarrow{R_3} \omega\bar{\omega}\omega \xrightarrow{R_2} \omega'\bar{\omega}'\omega$$

Les mots  $\bar{\omega}'\omega$  et  $\omega'^{-1}\omega'$  représentent tous deux le même idempotent, on peut donc appliquer la règle  $R_2$  :

$$\omega'\bar{\omega}'\omega \xrightarrow{R_2} \omega'\bar{\omega}'\omega' \xrightarrow{R_3} \omega'$$

Ceci nous assure que si deux mots représentent le même élément du monoïde  $M$  alors, on peut les dériver l'un de l'autre par  $R$ . De plus, les règles de  $R$  nous indiquent qu'à chaque étape de dérivation, on ne change pas l'élément décrit. ■

Néanmoins, ce résultat n'est pas étonnant puisque nous mettons dans  $R$  énormément de règles, notamment la réécriture de tous les mots représentant le même idempotent. Nous allons montrer qu'avec certaines conditions, nous pouvons affiner ce résultat.

**Définition 7.6** Soient  $M$  un monoïde inversif,  $(\Sigma, h)$  un système générateur de  $M$  et  $\text{Inv}$  une fonction inverse de  $(\Sigma, h)$ , on définit le système de réécriture  $S$  :

$$S = S_1 \cup S_2 \cup S_3$$

$$\text{avec } \begin{cases} S_1 = \{a \leftrightarrow a\bar{a}a \mid a \in \Sigma\} \\ S_2 = \{\alpha\alpha \rightarrow \alpha \mid h(\alpha) \in I_M\} \\ S_3 = \{\alpha \leftrightarrow \bar{\alpha} \mid h(\alpha) \in I_M\} \end{cases}$$

**Lemme 7.7** Du système de réécriture  $S$ , on peut déduire les ensembles de règles :

$$T_1 = \{\alpha\beta \leftrightarrow \beta\alpha \mid h(\alpha), h(\beta) \in I_M\}$$

$$T_2 = \{\alpha \leftrightarrow \alpha\bar{\alpha}\alpha \mid \alpha \in \Sigma^*\}$$

$$S'_2 = \{\alpha \rightarrow \alpha\alpha \mid h(\alpha) \in I_M\}$$

*Preuve.* Les règles de  $T_1$  sont dérivées comme ceci :

$$\alpha\beta \xleftrightarrow{S_3} \bar{\alpha}\bar{\beta} = \bar{\beta}\bar{\alpha} \xleftrightarrow{S_3} \beta\alpha$$

On montre que l'on peut déduire les règles de  $T_2$  par récurrence sur la taille de  $\alpha$ . Si  $\alpha = \varepsilon$ ,  $\alpha = \alpha\bar{\alpha}\alpha$ . Si  $|\alpha| = 1$ , on utilise les règles de  $S_1$ . Sinon, on suppose que la propriété est vraie pour tous les mots de taille strictement inférieure à celle de  $\alpha$  et on la montre vraie pour  $\alpha$ . On a  $\alpha = \alpha'a$  avec  $\alpha' \in \Sigma^*$  et  $a \in \Sigma$  :

$$\alpha = \alpha'a \xleftrightarrow{\text{réc.}} \alpha'\bar{\alpha'}\alpha'a \xleftrightarrow{S_1} \alpha'\bar{\alpha'}\alpha'a\bar{a}a \xleftrightarrow{T_1} \alpha'a\bar{a}\bar{\alpha'}\alpha'a = \alpha\bar{\alpha}\alpha$$

Les règles de  $S'_2$  sont les règles réciproques de celles de  $S_2$  et sont dérivées de la manière suivante :

$$\alpha \xleftrightarrow{T_2} \alpha\bar{\alpha}\alpha \xleftrightarrow{S_3} \alpha\alpha\alpha \xleftrightarrow{S_2} \alpha\alpha$$

■

La clôture transitive du système  $S$  ne correspond pas toujours à la congruence  $\approx$ . Par exemple, s'il existe un diviseur de l'élément neutre autre que  $\lambda$ , on a un mot non vide  $\omega$  équivalent à  $\varepsilon$ , or aucune règle de  $S$  ne permet de passer du mot vide à un mot non vide.

Notons que même si l'élément neutre n'a pas de diviseur,  $S$  peut ne pas être complet. Considérons le sous-monoïde inversif généré par l'ensemble de figures  $A$  :

$$A = \left\{ \begin{array}{c} \bullet \\ \blacksquare \\ \blacksquare \end{array}, \begin{array}{c} \bullet \\ \blacksquare \\ \blacksquare \end{array}, \begin{array}{c} \bullet \\ \bullet \\ \blacksquare \\ \blacksquare \end{array}, \begin{array}{c} \bullet \\ \bullet \\ \blacksquare \\ \blacksquare \end{array} \right\}$$

On représente les figures de  $A^*$  avec le système générateur  $(X, h)$  avec l'alphabet  $X = \{a, a', b, b'\}$  et  $h$  qui fait correspondre les lettres de  $X$  avec les figures de  $A$  dans l'ordre ci-dessus. Les mots  $bb$  et  $aa'bb$  représentent tous deux la figure  $f$  :

$$f = \begin{array}{|c|c|} \hline \circ & \bullet \\ \hline \hline \hline \end{array}$$

Si l'on construit une fonction inverse  $\text{Inv}$ , on voit que  $\text{Inv}(b)$  et  $\text{Inv}(b')$  ne peuvent contenir que des  $b$  et des  $b'$ . Toutes les règles de  $S$  qui ne contiennent que des  $b$  et des  $b'$  d'un côté de la règle ne contiennent donc que des  $b$  et des  $b'$  de l'autre. On ne peut donc pas dériver  $aa'bb$  à partir de  $bb$ .

**Théorème 7.8** Soient  $M$  un monoïde inversif,  $(\Sigma, h)$  un système générateur de  $M$ ,  $\text{Inv}$  une fonction inverse de  $(\Sigma, h)$  et le système de réécriture  $S$  défini ci-dessus. La relation  $\xrightarrow[S]{*}$  coïncide avec la congruence  $\simeq$  associée à  $(\Sigma, h)$  si et seulement si :

$$\forall \alpha \in \Sigma^*, a \in \Sigma \quad h(\alpha) = h(a\bar{a}\alpha) \Rightarrow \alpha \xrightarrow[S]{*} a\bar{a}\alpha \quad (P)$$

Afin d'obtenir ce théorème, nous montrons les résultats intermédiaires suivants :

**Lemme 7.9** Si la condition (P) est respectée, on a alors :

$$\forall \alpha, \beta \in \Sigma^* \quad h(\alpha) = h(\beta\bar{\beta}\alpha) \Rightarrow \alpha \xrightarrow[S]{*} \beta\bar{\beta}\alpha$$

*Preuve.* Nous montrons cette propriété par récurrence sur la taille de  $\beta$ . Si  $\beta = \varepsilon$ , la propriété est immédiate. Si  $|\beta| = 1$ , elle se déduit directement de (P). Sinon on suppose que la propriété est vraie pour tous les mots de longueur strictement inférieure à la longueur de  $\beta$  et on la montre vraie pour  $\beta$ . On a  $\beta = a\beta'$  avec  $\beta' \in \Sigma^*$  et  $a \in \Sigma$ .

Notons que l'on a  $h(a\bar{a}\alpha) = h(\alpha)$  et  $h(\beta'\bar{\beta}'\bar{a}\alpha) = h(\bar{a}\alpha)$ . En effet, on a :

$$\begin{aligned} h(\alpha) &= h(\beta\bar{\beta}\alpha) = h(a\beta'\bar{\beta}\alpha) = h(a).h(\beta'\bar{\beta}\alpha) = \\ &h(a).h(\bar{a}).h(a).h(\beta'\bar{\beta}\alpha) = h(a\bar{a}\beta'\bar{\beta}\alpha) = h(a\bar{a}\alpha) \end{aligned}$$

et

$$\begin{aligned} h(\bar{a}\alpha) &= h(\bar{a}).h(\alpha) = h(\bar{a}).h(a\beta'\bar{\beta}'\bar{a}\alpha) = h(\bar{a}a).h(\beta'\bar{\beta}').h(\bar{a}\alpha) = \\ &h(\beta'\bar{\beta}').h(\bar{a}a).h(\bar{a}\alpha) = h(\beta'\bar{\beta}').h(\bar{a}).h(a).h(\bar{a}).h(\alpha) = h(\beta'\bar{\beta}'\bar{a}\alpha) \end{aligned}$$

On a donc, d'après (P) :

$$\alpha \xrightarrow[S]{*} a\bar{a}\alpha \xleftrightarrow[\text{réc.}]{*} a\beta'\bar{\beta}'\bar{a}\alpha = \beta\bar{\beta}\alpha$$

■

**Lemme 7.10** Si la condition (P) est respectée, on a alors :

$$\forall \alpha, \beta \in \Sigma^* \quad h(\alpha) = h(\beta\alpha) \text{ et } h(\beta) \in I_M \Rightarrow \alpha \xrightarrow[S]{*} \beta\alpha$$

*Preuve.* Comme  $\beta$  décrit un idempotent, on a :

$$\begin{aligned} h(\beta\bar{\beta}\alpha) &= h(\beta).h(\bar{\beta}).h(\alpha) = h(\beta).h(\beta)^{-1}.h(\alpha) = \\ &h(\beta).h(\beta).h(\alpha) = h(\beta).h(\alpha) = h(\beta\alpha) = h(\alpha) \end{aligned}$$



On a donc (d'après le lemme 7.9) :

$$\alpha \xleftrightarrow[S]{*} \beta \bar{\beta} \alpha \xleftrightarrow[S_3]{} \beta \beta \alpha \xrightarrow[S_2]{} \beta \alpha$$

et dans l'autre sens :

$$\beta \alpha \xleftrightarrow[T_2]{} \beta \bar{\beta} \beta \alpha \xleftrightarrow[S_3]{} \beta \bar{\beta} \bar{\beta} \alpha \xrightarrow[S_2]{} \beta \bar{\beta} \alpha \xleftrightarrow[S]{*} \alpha$$

■

**Lemme 7.11** *Si la condition (P) est respectée, on a :*

$$\forall \alpha, \beta \in \Sigma^* \quad h(\alpha) = h(\alpha\beta) \text{ et } h(\beta) \in I_M \Rightarrow \alpha \xleftrightarrow[S]{*} \alpha\beta$$

*Preuve.* Il suffit de noter que si  $h(\alpha) = h(\alpha\beta)$  alors on a  $h(\bar{\alpha}) = h(\bar{\beta}\bar{\alpha})$ . Ceci nous permet d'appliquer le lemme 7.10 :

$$\alpha \xleftrightarrow[T_2]{} \alpha \bar{\alpha} \alpha \xleftrightarrow[S]{*} \alpha \bar{\beta} \bar{\alpha} \alpha \xleftrightarrow[T_1]{} \alpha \bar{\alpha} \alpha \bar{\beta} \xleftrightarrow[T_2]{} \alpha \bar{\beta} \xleftrightarrow[S_3]{} \alpha \beta$$

■

*Preuve.* (Théorème 7.8) Si la clôture transitive du système de réécriture correspond à la congruence  $\approx$ , il est trivial que  $\alpha$  se dérive en  $a\bar{a}\alpha$  s'il décrivent le même élément. Dans l'autre sens, les règles de  $S$  nous assurant qu'à chaque réécriture, l'élément de  $M$  décrit n'est pas modifié, il nous suffit de montrer que :

$$\forall \omega, \omega' \in \Sigma^* \quad h(\omega) = h(\omega') \Rightarrow \omega \xleftrightarrow[S]{*} \omega'$$

Si  $h(\omega) = h(\omega')$ , alors on a :

$$\begin{aligned} h(\omega \bar{\omega} \omega') &= h(\omega) = h(\omega') \\ h(\omega \bar{\omega}) &\in I_M \\ h(\bar{\omega} \omega') &\in I_M \end{aligned}$$

On a donc, en appliquant les lemmes 7.10 et 7.11 :

$$\omega \xleftrightarrow[S]{*} \omega \bar{\omega} \omega' \xleftrightarrow[S]{*} \omega'$$

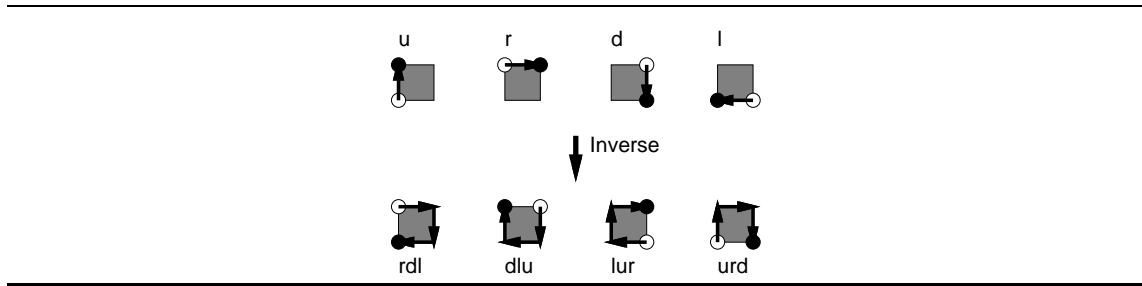
■

La relation induite par le système de réécriture  $S$  défini en 7.6 coïncide donc bien avec la congruence  $\approx$ , si la condition (P) est respectée. La propriété (P) est équivalente à la suivante :

$$\forall \alpha \in \Sigma^*, a \in \Sigma \quad h(\alpha) = h(a\bar{a}\alpha) \Rightarrow \exists \beta, \gamma \in \Sigma^* \quad \alpha \xleftrightarrow[S]{*} \beta a \gamma \quad \text{avec } h(\beta) \in I_M \quad (P')$$

En effet, on peut alors dériver  $\alpha$  de la manière suivante :

$$\alpha \xleftrightarrow[S]{*} \beta a \gamma \xleftrightarrow[S_1]{} \beta a \bar{a} a \gamma \xleftrightarrow[T_1]{} a \bar{a} \beta a \gamma \xleftrightarrow[S]{*} a \bar{a} \alpha$$

Figure 8: Inversion des lettres de  $\Pi$ 

Cette condition signifie que l'on peut faire apparaître un  $a$  dans  $\alpha$  au “bon endroit”, c'est-à-dire après un idempotent. En particulier,  $(P')$  est vraie si :

$$\begin{aligned} \forall \alpha \in \Sigma^*, a \in \Sigma \quad h(\alpha) = h(a\bar{a}\alpha) &\Rightarrow \exists \beta, \gamma \in \Sigma^* \\ \alpha\bar{a}\alpha = \beta a \gamma \quad \text{avec } h(\beta) \in I_M & \end{aligned} \quad (C)$$

Nous allons maintenant appliquer ce résultat à notre système générateur du monoïde des figures pointées connexes qui vérifie cette dernière condition.

## 8 Équivalence entre mots de figures

Nous pouvons utiliser le système de réécriture  $S$  pour notre système générateur de  $\mathcal{F}_p^c$ . Il nous suffit de définir la fonction inverse que l'on associe à  $(\Pi, \text{FigP})$ .

**Définition 8.1** La fonction inverse  $\text{Inv}$  que nous utilisons est définie par :

$$\text{Inv}(u) = rdl \quad \text{Inv}(r) = dlu \quad \text{Inv}(d) = lur \quad \text{Inv}(l) = urd$$

Nous donnons une illustration de l'inverse des lettres de  $\Pi$  en figure 8.

Afin de pouvoir utiliser le système de réécriture  $S$ , nous devons montrer que notre système générateur  $(\Pi, \text{FigP})$  avec  $\text{Inv}$  définie ci-dessus vérifie la condition  $(P)$  du théorème 7.8. Nous pouvons aisément identifier les idempotents de  $\mathcal{F}_p^c$  :

**Fait 8.2** Soit  $f = [(p, d, a)]_{\simeq} \in \mathcal{F}_p^c$ ,  $f \in I\mathcal{F}_p^c$  ssi  $d = a$ .

Lorsque nous parlerons des idempotents de  $\mathcal{F}_p^c$ , nous utiliserons le terme *boucles* puisque ce sont les figures pour lesquelles les points de départ et d'arrivée sont confondus. Nous notons  $\mathcal{B}$  l'ensemble des boucles.

**Proposition 8.3** Le monoïde  $\mathcal{F}_p^c$ , avec le système générateur  $(\Pi, \text{FigP})$  et la fonction inverse  $\text{Inv}$  définie ci-dessus respecte la condition (P).

*Preuve.* On montre que la condition (C) est vérifiée. Considérons  $\alpha \in \Pi^*$  et  $a \in \Pi$  avec  $f = \text{FigP}(\alpha) = \text{FigP}(a\bar{a}\alpha)$ . Raisonnons sur un représentant de  $p$  de  $f$  tel que  $\text{dép}(p) = (0, 0)$ . Supposons que  $a = r$ , les autres cas étant traités de manière similaire. On est sûr que  $\alpha \neq \varepsilon$  car  $\varepsilon$  est le seul représentant de  $f_\varepsilon$ . On peut donc noter  $\alpha = a_1 \dots a_n$  avec  $\forall 1 \leq i \leq n, a_i \in \Pi$ .

À chaque lettre de  $\alpha$ , on peut associer le pixel de  $p$  qu'elle “allume”. Pour ceci nous définissons les suites  $(x_i)_{i \in [0, n]}$ ,  $(y_i)_{i \in [0, n]}$  et  $(p_i)_{i \in [1, n]}$  :

$$\forall 1 \leq i \leq n \quad x_i = \begin{cases} x_0 = 0 \\ x_{k-1} + 1 & \text{si } a_k = r \\ x_{k-1} - 1 & \text{si } a_k = l \\ x_{k-1} & \text{sinon} \end{cases}$$

$$\forall 1 \leq i \leq n \quad y_i = \begin{cases} y_0 = 0 \\ y_{k-1} + 1 & \text{si } a_k = u \\ y_{k-1} - 1 & \text{si } a_k = d \\ y_{k-1} & \text{sinon} \end{cases}$$

$$\forall 1 \leq i \leq n \quad p_i = \begin{cases} \text{pix}(x_{k-1}, y_{k-1}) & \text{si } a_k = u \\ \text{pix}(x_{k-1}, y_{k-1} - 1) & \text{si } a_k = r \\ \text{pix}(x_{k-1} - 1, y_{k-1} - 1) & \text{si } a_k = d \\ \text{pix}(x_{k-1} - 1, y_{k-1}) & \text{si } a_k = l \end{cases}$$

Le couple  $(x_k, y_k)$  représente les coordonnées où l'on se trouve après avoir lu les  $k$  premières lettres de  $\alpha$ . Le pixel  $p_k$  désigne le pixel “allumé” par la  $k$ -ième lettre de  $\alpha$ .

Il est clair que :

$$\text{base}(p) = \bigcup_{i=1}^n p_i$$

Comme  $\text{FigP}(r\bar{r}\alpha) = \text{FigP}(\alpha) = p$ , il existe donc  $1 \leq k_0 \leq n$  tel que  $p(k_0) = \text{pix}(0, -1)$  ( $\text{pix}(0, -1)$  étant l'unique pixel de  $p$  allumé par  $r\bar{r}$ ). Deux possibilités :

1.  $a_{k_0} = r$ , on voit facilement que l'on peut prendre  $\beta = a_1 \dots a_{k_0-1}$  ;
2.  $a_{k_0} \neq r$ , on a alors  $\text{Inv}(a_{k_0}) = \bar{a}_{k_0} = \delta r \delta'$  avec  $\delta, \delta' \in \Pi^*$  et on pose  $\beta = \alpha \bar{a}_n \dots \bar{a}_{k_0+1} \delta$ .

La condition (P) est donc bien vérifiée. ■

On peut donc utiliser le système de réécriture  $S$  pour servir de base à la congruence. Néanmoins, dans le cas particulier du monoïde  $\mathcal{F}_p^c$ , nous allons pouvoir utiliser un système légèrement simplifié.

**Définition 8.4** Le système de réécriture  $S'$  est défini sur les mots de  $\Pi^*$  :

$$S' = S'_1 \cup S'_2 \cup S'_3$$

$$\text{avec } \begin{cases} S'_1 = \{a \leftrightarrow a\bar{a} \mid a \in \Pi\} \\ S'_2 = \{\alpha\alpha \rightarrow \alpha \mid \text{FigP}(\alpha) \in \mathcal{B}\} \\ S'_3 = \{\alpha \rightarrow \bar{\alpha} \mid \text{FigP}(\alpha) \in \mathcal{B}\} \end{cases}$$

La seule différence avec le système de réécriture  $S$ , c'est que les règles de l'ensemble  $S'_3$  ne vont que dans un seul sens. Mais les règles réciproques des règles de  $S'_3$  peuvent se déduire de  $S'$ .

**Proposition 8.5** Du système de réécriture  $S'$ , nous pouvons déduire les ensembles de règles :

$$T'_1 = \{\alpha \leftrightarrow \bar{\alpha} \mid \alpha \in \Pi^*\}$$

$$T'_2 = \{\bar{\alpha} \rightarrow \alpha \mid \text{FigP}(\alpha) \in \mathcal{B}\}$$

*Preuve.* Nous considérons les ensembles de règles une à une :

1.  $T'_1$  : Il suffit de montrer que les règles avec  $\alpha$  ne comportant qu'une lettre peuvent être déduites de  $S'$  :

$$\forall a \in \Pi \quad a \xrightarrow[S']{*} \bar{a}$$

Supposons que  $a = r$ , les autres cas se résolvant de manière symétrique. On a :

$$\bar{r} = \overline{dlu} = rdlurdlur = r\bar{r}r\bar{r}r \xrightarrow[S'_1]{2} r$$

On voit facilement que l'on peut généraliser aux mots sur  $\omega \in \Pi^*$ , en effet si  $\omega = a_1 \dots a_n$ , on a  $\bar{\omega} = \bar{a}_1 \dots \bar{a}_n$ . Notons que si  $\omega = \varepsilon$ , on a  $\bar{\omega} = \varepsilon$ .

2.  $T'_2$  : Il s'agit des règles réciproques de celles de  $S'_3$ . Soit  $\alpha \in \Pi^*$  tel que  $\text{FigP}(\alpha) \in \mathcal{B}$ , on a :

$$\bar{\alpha} \xrightarrow[S'_3]{\rightarrow} \bar{\alpha} \xrightarrow[T'_1]{\leftrightarrow} \alpha$$

■

Ceci nous permet d'assurer que la relation  $\xrightarrow[S']{*}$  est bien la congruence  $\equiv$ .

**Proposition 8.6** La relation  $\xrightarrow[S']{*}$  coïncide avec la congruence  $\equiv$  :

$$\forall \omega, \omega' \in \Pi^* \quad \omega \equiv \omega' \Leftrightarrow \omega \xrightarrow[S']{*} \omega'$$

*Preuve.* Immédiat d'après le théorème 7.8 et les propositions 8.3 et 8.5. ■

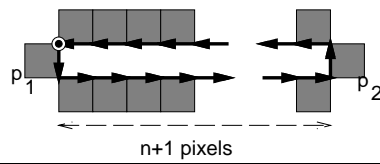


Figure 9: Le mot  $\omega$  ne pouvant être inversé

Notre système de réécriture  $S'$  contient un nombre infini de règles ( $S'_2$  et  $S'_3$ ) et il ne peut exister de système de réécriture fini possédant la propriété requise.

**Proposition 8.7** *Tout système de réécriture dont la relation induite est la congruence  $\equiv$  est infini.*

*Preuve.* Par l'absurde. Supposons qu'il existe un système de réécriture  $S''$  fini qui satisfait la propriété. Il existe donc  $n$  la longueur maximale des parties gauches et droites des règles de ce système.

On considère le mot  $\omega = dr^{n+1}ul^{n+1}$ . Ce mot représente une figure comme celle de la figure 9. Dans  $\omega$ , le pixel  $p_1$  est allumé avant le pixel  $p_2$ .

Soit le mot  $\omega' = ur^{n+1}drdl^{n+1}ulur$ . Les mots  $\omega$  et  $\omega'$  décrivent bien la même figure. Donc on a  $\omega \xrightarrow[S'']{*} \omega'$ . Or dans  $\omega'$ , le pixel  $p_2$  est allumé avant le pixel  $p_1$ . Il existe donc dans la dérivation  $\omega_1$  et  $\omega_2$  tels que  $\omega_1$  se dérive en  $\omega_2$  en une seule étape :

$$\omega \xrightarrow[S'']{*} \omega_1 \xrightarrow[S'']{\rightarrow} \omega_2 \xrightarrow[S'']{*} \omega'$$

et tels que dans  $\omega_1$  le pixel  $p_1$  soit allumé avant  $p_2$ , dans  $\omega_2$  le pixel  $p_2$  soit allumé avant  $p_1$ . Il existe donc dans  $S''$  une règle  $\alpha_1 \rightarrow \alpha_2$  et on a  $\omega_1 = \beta\alpha_1\beta'$  et  $\omega_2 = \beta\alpha_2\beta'$ . Clairement,  $p_1$  et  $p_2$  sont tracés dans  $\alpha_1$  et  $\alpha_2$  sinon l'ordre ne serait pas inversé. Donc  $\alpha_1$  trace  $p_1$  et  $p_2$ . Or la distance entre  $p_1$  et  $p_2$  est strictement supérieure à  $n$ , donc la longueur de  $\alpha_1$  est strictement supérieure à  $n$ , d'où contradiction. ■

## 9 Langages de figures classiques

Comme nous l'avons déjà cité, le théorème 7.8 est aussi une généralisation d'un résultat montré par P. Séebold et K. Slowinski [10]. En effet, dans la sémantique "classique" des langages de figures proposée par H.A. Maurer et al. dans [5], les mots sur  $\Pi$  décrivent des figures composées de segments.

Les dessins sont alors des ensembles finis de segments unitaires reliant des points voisins de  $\mathbb{Z}^2$ . Les dessins pointés sont des dessins munis d'un point de départ et d'un point d'arrivée. La concaténation est définie de manière identique à celle proposée à la définition 3.1. Les figures sont également des classes d'équivalence de dessins modulo une translation.

L'ensemble étudié dans cette sémantique est l'ensemble des figures connexes qui sont engendrées par l'ensemble des segments unitaires  $Q = \{s_u, s_r, s_d, s_l\}$  illustré figure 10. Il s'agit également d'un monoïde inversif qui a comme système générateur le couple  $(\Pi, \text{dpic})$  avec  $\Pi = \{u, r, d, l\}$  et  $\text{dpic}(u) = s_u$ ,  $\text{dpic}(r) = s_r$ ,  $\text{dpic}(d) = s_d$  et  $\text{dpic}(l) = s_l$ .

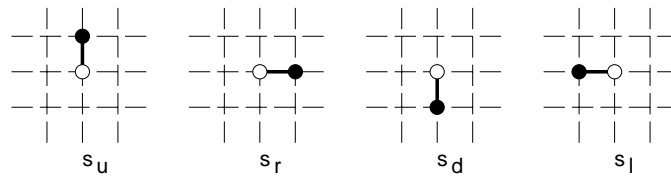


Figure 10: Ensemble des segments unitaires

On montre de manière similaire à la proposition 8.3 que la condition (P) est vérifiée par ce système générateur doté de la fonction inverse  $\text{Inv} : \text{Inv}(u) = d$ ,  $\text{Inv}(r) = l$ ,  $\text{Inv}(d) = u$  et  $\text{Inv}(l) = r$ . Ceci nous permet d'utiliser le système de réécriture  $S$  pour la congruence associée à  $(\Pi, \text{dpic})$ . Notons qu'ici, nous avons pour tout mot  $\alpha$  sur  $\Pi$  :  $\bar{\alpha} = \alpha$  et, ainsi, l'ensemble de règle  $S_3$  est  $\{\alpha \rightarrow \bar{\alpha} \mid h(\alpha) \in I_{Q^*}\}$ . On retrouve donc exactement le système de réécriture de [10]. De même, R. Gutbrod a proposé dans [4] un système de réécriture ayant les mêmes propriétés mais contenant  $(S)$ .

## 10 Conclusion

Dans cet article, nous proposons un formalisme permettant de décrire des figures composées de pixels à l'aide de mots. Il s'agit d'une adaptation des langages de figures "classiques" décrit par H.A. Maurer et al. [5]. La théorie des langages formels (monoïdes inversifs, systèmes générateurs,...) nous a permis d'étayer et de faciliter notre approche. Dans la recherche d'un système de réécriture engendrant la congruence associée au système générateur du monoïde des figures, nous montrons qu'une adaptation du système de P. Séebold et K. Slowinski [10] peut être utilisée pour un système générateur d'un monoïde inversif vérifiant certaines conditions.

La sémantique que nous proposons nous incite à nous poser un certain nombre de questions. À l'instar de [5] on peut définir la complexité descriptive d'une figure  $f$  comme étant le rapport entre la longueur du plus petit mot qui décrit  $f$  et le poids de  $f$ . Quelle est la borne de complexité descriptive des figures ? Cette sémantique nous permet également de décrire des polyominos [2]. Quelles sont les bornes de complexité pour les différentes classes de polyominos (polyominos quelconques, polygones, polyominos piles...) ?

## Remerciements

Nous remercions Bodonirina Ratoandromanana pour sa participation aux travaux préliminaires de ce papier. Nous remercions également le rapporteur anonyme pour ses remarques constructives.

## Références

- [1] J. Berstel, *Tranductions and Context-Free Languages*, Teubner Studienbücher, Stuttgart (1979).

- [2] M. Bousquet-Mélou, Polyominoes and polygons, *Contemporary Mathematics* **178** (1994) 55–70.
- [3] S. Eilenberg, *Automata, Languages and Machines*, volume A, Academic Press (1974).
- [4] R. Gutbrod, A transformation system for generating description languages of chain code pictures, *Theoretical Computer Science* **68** (1989) 239–252.
- [5] H.A. Maurer, G. Rozenberg, and E. Welzl, Using string languages to describe picture languages, *Information and Control* **54** (1982) 155–185.
- [6] M. Petrich, *Inverse Semigroups*, Wiley, New York (1984).
- [7] M. Pelletier et J. Sakarovitch, Easy multiplication II – extensions of rational semigroups, *Information and Computation* **88** (1990) 18–59.
- [8] J. Sakarovitch, Description des monoïdes de type fini, *Journal of Information Processing and Cybernetics* **17**, 8 (1981) 417–434.
- [9] J. Sakarovitch, Easy multiplication I – the realm of Kleene’s theorem, *Information and Computation* **74** (1987) 173–197.
- [10] P. Séébold et K. Slowinski, The shortest way to draw a connected picture, *Computer Graphics Forum* **10** (1991) 319–327.

Michel Latteux, Denis Robilliard et David Simplot  
C.N.R.S. U.A. 369, L.I.F.L. Université de Lille I  
Bât. M3, Cité Scientifique  
F-59655 Villeneuve d’Ascq Cedex (France)  
{latteux,robillia,simplot}@lifl.fr