Theoretical Computer Science 108 (1993) 357–363
Elsevier

Note

Picture words with invisible lines

Karine Slowinski
CNRS, U.A. 369, L.I.F.L., Université de Lille 1, 59655 Villeneuve d'Ascq Cedex, France

Communicated by M. Nivat
Received September 1990
Revised February 1992

Abstract

Slowinski, K., Picture words with invisible lines, Theoretical Computer Science 108 (1993) 357–363.

A picture word is a word over the alphabet $\{r, \bar{r}, u, \bar{u}, r_b, \bar{r}_b, u_b, \bar{u}_b\}$. With any picture word, we associate a picture as follows: the reading of each letter of the word induces a unit move; the letters r and r_b ($\bar{r}$ and $\bar{r}_b$, u and u_b, $\bar{u}$ and $\bar{u}_b$) stand for a right (left, up, down) move; for each letter from $\{r, \bar{r}, u, \bar{u}\}$, we move by drawing a unit line; for the other letters, we move with "pen-up". We present a rewriting system S which generates exactly all the picture words describing a given picture.

0. Introduction

A *picture* is a finite set of horizontal or vertical unit lines whose extremities have integer coordinates in the Cartesian plane. Such a picture can be represented by a *picture word* defined over the alphabet $\{r, \bar{r}, u, \bar{u}, r_b, \bar{r}_b, u_b, \bar{u}_b\}$. The reading of each letter of the word induces a unit move: r and r_b ($\bar{r}$ and $\bar{r}_b, u$ and $u_b, \bar{u}$ and $\bar{u}_b$) stand for a right (left, up, down) move. For each letter from $\{r, \bar{r}, u, \bar{u}\}$, we move by drawing a unit line; for the other letters, we move with "pen-up". This approach can be compared with the encoding established by Freeman [2] to facilitate the processing of geometric configurations. It can be used to conduct a plotter pen. The aim of this paper is to present a rewriting system which generates exactly all the picture words describing a given picture.

Correspondence to: K. Slowinski, CNRS, U.A. 369, L.I.F.L., Université de Lille 1, 59655 Villeneuve d'Ascq Cedex, France

The description of pictures by words was initiated by [8]. It allows one to use results and techniques from the formal language theory [4, 9]. This paper completes previous works about words which describe connected pictures (there is no letter to pen-up): different rewriting systems on picture words are studied [1, 5] and some of them generate all the words describing a given connected picture [3, 10]. We are interested in picture words with invisible lines to describe connected or nonconnected pictures. To do this, we enlarge the alphabet to simulate the lifting and the sinking of the pen [6] and we define a new rewriting system.

This paper is organized as follows. After the preliminaries (Section 1), we present the rewriting system S (Section 2) and we show that, from any picture word, the system S generates exactly all the words describing the same picture (Section 3).

1. Preliminaries

1.1. Notations

We assume the reader to be familiar with the basic formal language theory [4, 9], and we just remind him of several notations.

Let A be a finite set called *alphabet*. The elements of A are *letters* and finite strings over A are *words*. A^* denotes the free monoid generated by A and ε the empty word. For any word $w \in A^*$, $|w|$ denotes its *length* (in particular, $|\varepsilon| = 0$).

A *rewriting system* σ over A is a subset of $A^* \times A^*$. Each element $(x, y) \in \sigma$ is a *rule* and is denoted by $x \to y$. A *step of derivation* $\Rightarrow_\sigma$ is defined by $(w \Rightarrow_\sigma w') \Leftrightarrow (\exists w_1, w_2 \in A^* \mid w = w_1 x w_2,\ w' = w_1 y w_2$ and $(x, y) \in \sigma)$. We denote by $\Rightarrow_\sigma^*$ the transitive and reflexive closure of $\Rightarrow_\sigma$. If $w \Rightarrow_\sigma^* w'$, we say that *$w'$ is derived from w*. A system σ is *symmetric* if we have $\forall w, w' \in A^*\ (w \Rightarrow_\sigma^* w') \Rightarrow (w' \Rightarrow_\sigma^* w)$. A system σ is *finite* if it contains a finite number of rules (for further details, see [7]).

1.2. Basic notions

In this paper, we use the following definitions.

A *vertex* is an element of $\mathbf{Z}^2$ ($\mathbf{Z}$ denotes the set of integers).

A *segment* is an unordered pair of vertices $\{(m, n), (m+1, n)\}$ or $\{(m, n), (m, n+1)\}$. It can be represented by a horizontal or vertical unit line joining these two vertices in the Cartesian plane $\mathbf{Z}^2$.

A *picture* is a finite set of segments.

A *picture word* is a word over the alphabet $\Pi = \{r, \bar{r}, u, \bar{u}, r_b, \bar{r}_b, u_b, \bar{u}_b\}$. With each letter from Π, we associate a unit move as follows: r ($\bar{r}$, u, $\bar{u}$) induces a unit line and r_b ($\bar{r}_b, u_b, \bar{u}_b$) a unit move with pen-up to the right (left, up, down) (Fig. 1).

So, with any picture word we *construct* a picture as follows: we start from a given vertex, read the word, letter by letter, from left to right, and generate the associated move.

Fig. 1.

We define a morphism sh which associates with any letter from Π the motion vector in $\mathbf{Z}^2$: $\mathrm{sh}(r)=\mathrm{sh}(r_b)=(1,0)$, $\mathrm{sh}(\bar{r})=\mathrm{sh}(\bar{r}_b)=(-1,0)$, $\mathrm{sh}(u)=\mathrm{sh}(u_b)=(0,1)$, $\mathrm{sh}(\bar{u})=\mathrm{sh}(\bar{u}_b)=(0,-1)$; and a morphism b from Π to $\Pi_b=\{r_b,\bar{r}_b,u_b,\bar{u}_b\}$ which "blanks" the letters: if $a\in\Pi-\Pi_b$ then $b(a)=a_b$; otherwise, $b(a)=a$ ($\forall w\in\Pi^*$, $b(w)$ is denoted by w_b).

Let w_1 and w_2 be picture words. The *trace of* w_2 *after* w_1, denoted by $\mathrm{tr}(w_1, w_2)$, is defined inductively by $\mathrm{tr}(w_1,\varepsilon)=\emptyset$; for $w\in\Pi^*$ and $a\in\Pi$, $\mathrm{tr}(w_1,wa)=\mathrm{tr}(w_1,w)\cup\{\{\mathrm{sh}(w_1w),\mathrm{sh}(w_1wa)\}\}$. The *picture of* w_2 *after* w_1, denoted by $\mathrm{pic}(w_1,w_2)$, is defined inductively by $\mathrm{pic}(w_1,\varepsilon)=\emptyset$; for $w\in\Pi^*$ and $a\in\Pi$, $\mathrm{pic}(w_1,wa)=\mathrm{pic}(w_1,w)\cup\{\{\mathrm{sh}(w_1,w),\mathrm{sh}(w_1wa)\}\}$ if $a\in\Pi-\Pi_b$ and $\mathrm{pic}(w_1,wa)=\mathrm{pic}(w_1,w)$ if $a\in\Pi_b$.

Let w be a picture word. The *drawn picture of* w, denoted by $\mathrm{dpic}(w)$, is the pair $(\mathrm{pic}(\varepsilon,w),\mathrm{sh}(w))$. Note that $\mathrm{pic}(\varepsilon,w)$ is the set of segments drawn during the construction of w starting at the origin of the Cartesian plane and $\mathrm{sh}(w)$ represents the coordinates of the ending point. For example, $w=uru_brrur\bar{u}rr\bar{u}_br\bar{u}$; see Fig. 2.

Drawing convention

- The starting (ending) point is represented by ◦ (•).
- On the construction, a segment described several times is represented by parallel traces and a move with pen-up by a dashed line.

Given a picture word, the associated drawn picture is unique. Conversely, given a drawn picture, there exists an infinite number of picture words describing this drawn picture. For example, $p=\mathrm{dpic}(w)=\mathrm{dpic}(w_n)$ for $n\geqslant 1$ (Fig. 3).

The *inverse of* w, denoted by $\mathrm{inv}(w)$, is the word having the same construction of w but in the opposite way. It is defined inductively by $\mathrm{inv}(\varepsilon)=\varepsilon$, $\mathrm{inv}(r)=\bar{r}$, $\mathrm{inv}(\bar{r})=r$, $\mathrm{inv}(u)=\bar{u}$, $\mathrm{inv}(\bar{u})=u$, $\mathrm{inv}(r_b)=\bar{r}_b$, $\mathrm{inv}(\bar{r}_b)=r_b$, $\mathrm{inv}(u_b)=\bar{u}_b$, $\mathrm{inv}(\bar{u}_b)=u_b$ and $\mathrm{inv}(wa)=\mathrm{inv}(a)\,\mathrm{inv}(w)$, where $w\in\Pi^*$ and $a\in\Pi$. In the example shown in Fig. 4, and thereafter, we use the notation $\bar{w}$ instead of $\mathrm{inv}(w)$.

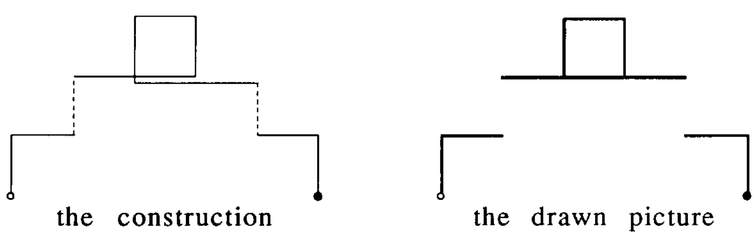


Fig. 2.

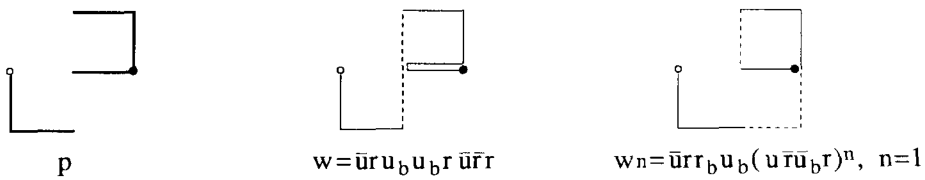


Fig. 3.

Fig. 4.

A *loop* is a picture word such that the starting and ending points of the associated construction are the same. $\mathbb{L}=\{l\in\Pi^*\mid \mathrm{sh}(l)=(0,0)\}$ is the set of loops. Note that $\forall l\in\mathbb{L}$, $\mathrm{dpic}(l)=\mathrm{dpic}(\bar{l})$.

Let s be a segment and w a picture word. The segment *s is described* (*drawn*) *by w* if there exists a decomposition of $w=w_1aw_2$, where $a\in\Pi$, such that $\{s\}=\mathrm{tr}(w_1,a)$ ($\{s\}=\mathrm{pic}(w_1,a)$). A picture word w is *optimal* if each segment is drawn at most once during the associated construction (but it can be described several times with pen-up): $|\{w_1\mid w=w_1aw_2 \text{ and } \{s\}=\mathrm{pic}(w_1,a)\}|\leqslant 1$.

2. The rewriting system S

We introduce a rewriting system S which generates, from any picture words, *exactly* all the words describing the same drawn picture. This system is composed of 6 sets of rewriting rules:

$$S=S1\cup S2\cup S2'\cup S3\cup S3'\cup S4,$$

with

$$S1=\{l\rightarrow\bar{l}\mid l\in\mathbb{L}\},$$

$$S2=\{a\bar{a}\rightarrow a\bar{a}_b\mid a\in\Pi-\Pi_b\},$$

$$S2'=\{a\bar{a}_b\rightarrow a\bar{a}\mid a\in\Pi-\Pi_b\},$$

$$S3=\{a_b\bar{a}_b\rightarrow\varepsilon\mid a_b\in\Pi_b\},$$

$$S3'=\{\varepsilon\rightarrow a_b\bar{a}_b\mid a_b\in\Pi_b\},$$

$$S4=\{a_ba'_b\rightarrow a'_ba_b\mid a_b,a'_b\in\Pi_b\}.$$

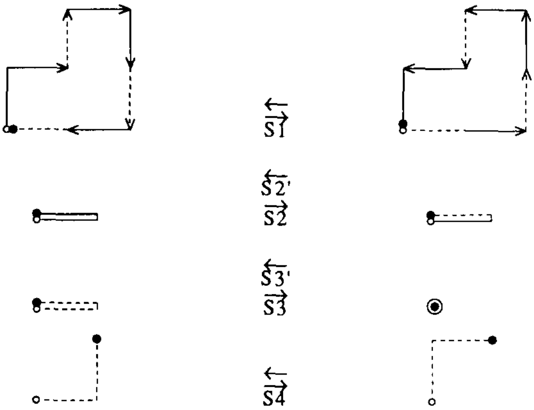


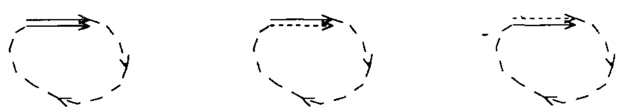
Fig. 5.

Remarks (cf. Fig. 5).
- The system S preserves the drawn picture.
- The system S is not finite: the set of rules $S1$ contain an infinite number of rules.
- The system S is symmetric.

In the sequel, we use the term "rule" instead of "set of rules". Thus, we write "the rule $S1$" instead of "one rule from the set of rules $S1$". The system S will be implicit for any derivations.

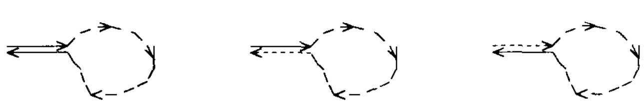
3. Results

The following transformations can be simulated by rules from S.
From $S1$, $S2$ and $S2'$, we deduce



$$T1 = \{aca \leftrightarrow aca_b \leftrightarrow a_b ca \mid a \in \Pi - \Pi_b,\ ca \in \mathbb{L}\},$$

and from $S1$, $S2$, $S2'$, $S3$ and $S3'$



$$T2 = \{al\bar{a} \leftrightarrow al\bar{a}_b \leftrightarrow a_b l\bar{a} \mid a \in \Pi - \Pi_b,\ l \in \mathbb{L}\}.$$

From these two results, we claim the following lemma.

Lemma 3.1. *From any picture word, we can derive an optimal word by S.*

From $S3$, $S3'$ and $S4$, we deduce

$$T3=\{l_b \leftrightarrow \varepsilon \mid l_b \in \mathbb{L} \cap \Pi_b^*\}.$$

From S', we deduce $T4=\{a\bar{a}a \leftrightarrow a \mid a\in\Pi\}$ and $T5=\{ll \leftrightarrow l \mid l\in\mathbb{L}\}$. Note that all the rules of the system defined in [10] can be simulated by rules from S.

A rewriting system σ over Π is *exhaustive* if, from any picture word, we can derive all the words describing the same drawn picture.

Theorem 3.2. *The system S is exhaustive.*

Proof. Let w_1 and w_2 be two picture words such that $\text{dpic}(w_1)=\text{dpic}(w_2)$. We can suppose that w_1 and w_2 are optimal (Lemma 3.1). Let $v_b\in\Pi_b^*$ such that $\text{sh}(v_b)=\text{sh}(w_1)=\text{sh}(w_2)$: the words $\bar{w}_{1b}v_b$ and $\bar{w}_{2b}v_b$ are loops from Π_b^*. So, we can derive w_1 and w_2 as follows:

$$w_1 \Rightarrow_{T3} w'_1=w_1\bar{w}_{2b}v_b \quad \text{and} \quad w_2 \Rightarrow_{T3} w_2\bar{w}_{1b}v_b \Rightarrow_{S1} w'_2=w_{1b}\bar{w}_2 v_b.$$

For each segment s from $\text{pic}(\varepsilon,w_1)=\text{pic}(\varepsilon,w_2)$, there exists a unique letter a_1 in w_1 and a unique letter a_2 in w_2 drawing s. We set $w_1=v_1a_1v'_1$ and $w_2=v_2a_2v'_2$, with $a_1,a_2\in\Pi-\Pi_b$ and $v_1,v'_1,v_2,v'_2\in\Pi'^*$. Clearly, in $w_1\bar{w}_{2b}$, the letters a_1 and $\bar{a}_{2b}$ describe the same segment. Using the transformation $T2$, we obtain

$$w_1 w_{2b}=v_1a_1v'_1\bar{v}'_{2b}\bar{a}_{2b}\bar{v}_{2b} \Rightarrow_{T2} v_1a_1v'_1\bar{v}'_{2b}\bar{a}_2\bar{v}_{2b} \Rightarrow_{T2} v_1a_{1b}v'_1\bar{v}'_{2b}\bar{a}_2\bar{v}_{2b},$$

and repeating this operation for all the segments from $\text{pic}(\varepsilon,w_1)=\text{pic}(\varepsilon,w_2)$, we deduce $w_1\bar{w}_{2b}\Rightarrow^* w_{1b}\bar{w}_2$. Since the system S is symmetric, we have $w_1\Rightarrow^* w'_1\Rightarrow^* w'_2\Rightarrow^* w_2$. $\square$

We do not know if the system S is minimal (exhaustive with a minimal number of rules) but, clearly, no rule from the system S can be simulated by the other sets of rules. Moreover, we show that to be exhaustive, a rewriting system needs an infinite number of rules.

Proposition 3.3. *There is no exhaustive rewriting system which is finite.*

Proof. If a rewriting system σ is finite, there exists $p>0$ such that σ is *bounded* by p: $\forall(x\rightarrow y)\in\sigma$, $|x|\leqslant p$ and $|y|\leqslant p$. Let $w\in\Pi^*$ and $(s,s')\in\text{pic}^2(\varepsilon,w)$; the segment s is *before* the segment s' *in* w, denoted by $s<_w s'$, if there exists a decomposition of $w=v_1av_2a'v_3$

such that $\{s\} = \mathrm{pic}(v_1, a)$ and $\{s'\} = \mathrm{pic}(v_1 a v_2, a')$, where $a, a' \in \Pi$; the *distance between* the segments *s and s'* denoted by $\mathrm{dist}(s, s')$, is defined by $\mathrm{dist}(s, s') = \min\{\, |v_2| \mid \{s\} = \mathrm{pic}(v_1, a), \{s'\} = \mathrm{pic}(v_1 a v_2, a'), v_1, v_2 \in \Pi^* \text{ and } a, a' \in \Pi \}$.

Let σ be a rewriting system preserving the picture, bounded by p, and let w be a picture word. Clearly, if $(s, s') \in \mathrm{pic}^2(\varepsilon, w)$ such that $\mathrm{dist}(s, s') \geqslant p$ then for any derived word w' from w by σ, we have $(s <_w s') \Rightarrow (s <_{w'} s')$. We choose $w = r^n \bar{r}^n r^n$, with $n > p + 1$. The drawn picture is

It is obvious that the minimal word (with respect to the length) describing this drawn picture is $m = r^n$. Studying the following decompositions of w: $w = \mathbf{r} r^{n-1} \bar{\mathbf{r}} \bar{r}^{n-1} r^n = r^n \bar{\mathbf{r}} \bar{r}^{n-1} \mathbf{r} r^{n-1}$, we deduce that $s_1 <_w s_n$ and $s_n <_w s_1$. Since $\mathrm{dist}(s_1, s_n) \geqslant p$, for any derived word w' from w by σ, we have $s_1 <_{w'} s_n$ and $s_n <_{w'} s_1$. This implies that m cannot be derived from w by σ. The system σ is not exhaustive. $\square$

In this paper, we present a rewriting system which generates exactly all the picture words describing a given drawn picture. Could we find an algorithm to derive by S a minimal word from any picture word?

Acknowledgment

The author thanks M. Latteux and P. Séébold for their helpful comments.

References

[1] F.J. Brandenburg and J. Dassow, Reductions of picture words (submitted).
[2] H. Freeman, On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Comput.* **10** (1961) 260–268.
[3] R. Gutbrod, A transformation system for generating description languages of chain code pictures, *Theoret. Comput. Sci.* **68** (1989) 239–252.
[4] M. Harrison, *Introduction to Formal Language Theory* (Addison-Wesley, Reading, MA, 1978).
[5] F. Hinz, Classes of picture languages that cannot be distinguished in the chain code concept and deletion of redundant retreats. in: *Proc. STACS '89*, Lecture Notes in Computer Science, Vol. 349 (Springer, Berlin, 1989) 132–143.
[6] F. Hinz and E. Welzl, Regular chain code picture languages with invisible lines, Report 252, Université de Graz, 1988.
[7] M. Jantzen, *Confluent String Rewriting*, EATCS Monographs on Theoretical Computer Science, Vol. 14 (Springer, Berlin, 1988).
[8] H.A. Maurer, G. Rozenberg and E. Welzl, Using string languages to describe picture languages. *Inform. and Control* **54** (1982) 155–185.
[9] A. Salomaa, *Formal Languages* (Academic Press, London, 1973).
[10] P. Séébold and K. Slowinski, Minimizing picture words, in: *Proc. IMYCS '90*, Lecture Notes in Computer Science, Vol. 464 (Springer, Berlin, 1990). 234–243; Extended version: The shortest way to draw a connected picture, *Comput. Graph. Forum* **10** (1991) 319–327.