An extension of the wreath product principle for finite Mazurkiewicz traces

G. Guaiana

DIR, Faculté des Sciences et Techniques, Université de Rouen, F-76821 Mont-Saint-Aignan Cedex
Giovanna.Guaiana@dir.univ-rouen.fr

R. Meyer, A. Petit

LSV, URA 2236 CNRS, ENS de Cachan, 61, av. du Prés. Wilson, F-94235 Cachan Cedex
{rmeyer,petit}@lsv.ens-cachan.fr

P. Weil

LIAFA, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05
Pascal.Weil@liafa.jussieu.fr

Key words: Concurrency. Formal languages.

1 Introduction

The notion of wreath product of transformation monoids has been well established in automata theory since the Krohn-Rhodes theorem in the early sixties [3]. The Krohn-Rhodes theorem states that every transformation monoid can be simulated by (technically: divides) a wreath product of simpler transformation monoids: copies of the 2-state flip-flop, or reset automaton, and simple permutation groups.

In the late seventies, Straubing stated and proved the wreath product principle [10], a result describing the languages recognized by a wreath product $S \circ T$, in terms of sequential functions and in terms of the languages recognized by S and T. Straubing's result has been used, along with the Krohn-Rhodes theorem, to give elegant syntactic proofs of several important theorems in different fields of computer science: in particular in language theory with Schützenberger's theorem on star-free languages [9]; in logic with McNaughton and Papert's theorem on first-order definable languages [7]; and in verification with Kamp's theorem on PTL (propositional temporal logic) definable languages of finite sequences [6].

In the study of (Mazurkiewicz) trace languages (see [1]), the notion of recognizing morphisms has found deep applications, in particular the extension of Ochmański's theorem on concurrent iteration to infinite traces [4]. However, very few papers used the structure of the recognizing semigroups to handle specific sub-classes of recognizable trace languages. In fact the only such results concern the star-free trace languages, their algebraic characterization [5] and their logical characterization [2].

The difficulty has come from certain algebraic technical manipulations, which are well established when one considers the free monoid, and which are either difficult or false in the case of trace monoids, so must be side-stepped.

The objective of this paper is to clear up one, rather evolved, such algebraic tool for the study of trace languages, namely the wreath product principle. This result can certainly be of great interest, as in classical language theory. It has already provided a key argument in the proof of a result of Meyer and Petit [8] on PTrL-definable trace languages.

We would like to mention that Straubing's book [11] provides a remarkable presentation of applications of the wreath product to the logical study of formal languages. We would be happy if this paper would contribute to the proof of results as useful in the context of trace languages.

The paper is organized as follows. Section 2 contains the basic definitions of traces, transformation monoids and the wreath product. Our main results are stated and proved in Section 3.

2 Basic definitions

Let us review basic definitions and results. We will give here only what is needed for the proof of our main results. For more details on trace languages, the reader is referred to the famous Trace Book [1]. On recognizability by transformation monoids and the wreath product, good reference books are Eilenberg's [3] and Straubing's [11].

2.1 Finite traces

A *dependence alphabet* is a pair (A, I) where A is a finite set (or *alphabet*) and $I \subseteq A \times A$ is a symmetric and irreflexive relation called the *independence rela-*

tion. Two letters $a, b \in A$ are said *independent* if $(a, b) \in I$. The complement relation, namely $D = (A \times A) \setminus I$, is called the *dependence relation*.

Recall that a *monoid* is a set closed by product and containing an identity element. Let A^* be the free monoid (with respect to concatenation) of finite (and empty) words on the alphabet A, and let $\sim_I$ be the congruence induced on A^* by the relation $\{(ab, ba) \mid (a, b) \in I\}$. The set of finite (and empty) *traces* $\mathbf{M}(A, I)$ can be defined as the quotient of the free monoid A^* by the congruence $\sim_I$. In the sequel, we denote by φ the canonical projection mapping from A^* onto $\mathbf{M}(A, I)$. Thus a *finite trace* u is an equivalence class of words w such that $u = \varphi(w)$. Besides, since $\sim_I$ is a congruence, $\mathbf{M}(A, I)$ is naturally equipped with a monoid structure, and the mapping φ is a morphism.

2.2 Transformation monoids

A *transformation monoid* (a *tm* for short) is a pair (P, S), where P is a finite set and S is a subset of mappings from P into itself which has a monoid structure (i.e. which is closed under composition and which contains the identity mapping id_P). In contradiction with [3] but in keeping with usual practice, we write mappings on the left: if $p \in P$ and $s \in S$, then the image of p under the mapping s is written $s[p]$.

Example 1 *Let $\mathit{2}$ be the 2-element set $\{a, b\}$. Let α (resp. β) be the constant mapping defined on $\mathit{2}$, mapping both elements to a (resp. b). Finally, let $U_2 = \{id_{\mathit{2}}, \alpha, \beta\}$. It is immediately verified that U_2 is closed under composition, so $(\mathit{2}, U_2)$ is a tm.*

Example 2 *Let S be a finite monoid. Then S naturally induces a transformation monoid (S, S), where the action of an element $s \in S$ on the set S is given by the right multiplication by s: if $p \in S$, we have $s[p] = ps$.*

Let M be a monoid, and let L be a subset of M. We say that L is *recognized* by a tm (P, S) if there exists a morphism $\psi : M \to S$, an element $p_0 \in P$ and a subset $F \subseteq P$ such that $L = \{u \in M \mid \psi(u)[p_o] \in F\}$. It is easily seen that recognizability by a tm is equivalent to the usual notion of recognizability (by a finite index congruence, by a morphism into a finite monoid, or in the case where M is the free monoid, by a finite state automaton).

2.3 Wreath product

Let $X = (P, S)$ and $Y = (Q, T)$ be two tms. The *wreath product* $X \circ Y$ of X by Y is defined to be the tm $Z = (P \times Q, S^Q \times T)$ where, for any $f \in S^Q$

(that is, f is a mapping from Q into S) and any $t \in T$, the mapping (f, t) is defined on $P \times Q$ as follows.

$$(f, t)[(p, q)] = (f(q)[p], t[q]) \text{ for any } (p, q) \in P \times Q.$$

An easy computation shows that the composition of two such applications, say (f_1, t_1) and (f_2, t_2), is given by:

$$(f_1, t_1)(f_2, t_2) = (f, t_1 t_2) \text{ with } f(q) = f_1(q) + f_2(t_1[q]) \text{ for all } q \in Q$$

where $+$ denotes the composition in S.

The wreath product is an associative operation on transformation monoids.

3 The wreath product principle on traces

We now proceed with the statement and the proof of our main results, connecting wreath products to sequential functions for traces.

3.1 Sequential functions

A *sequential transducer* on A^* with output in a monoid M is a "deterministic automaton with output", that is, $\mathcal{T} = (Q, i, \cdot, *)$ where Q is the (finite) set of states, $i \in Q$ is the initial state, $\cdot$ is the transition function from $Q \times A$ into Q, and $*$ is the output function from $Q \times A$ into M. As usual, the transition function is extended to $Q \times A^*$.

The transducer $\mathcal{T}$ defines a *sequential function* $\sigma' \colon A^* \to M$ as follows. First we extend the output function $*$ to $Q \times A^*$ by letting $q * \varepsilon = q$ and $q * (ua) = (q * u)((q \cdot u) * a)$. Then we let $\sigma'(u) = q_0 * u$ for each $u \in A^*$.

The idea is that one reads the word u in the automaton, starting from the initial state (there is no ambiguity since our (underlying) automaton is deterministic), and each edge traversed (of the form (q, a)) yields some output (an element of M). The image of u under σ' is the concatenation of the outputs yielded by the edges along the path from i labeled u.

Let $\tau \colon A^* \to Q^Q$ be the *transition morphism* of the underlying automaton of $\mathcal{T}$, which maps each word $u \in A^*$ to the transition $q \mapsto q \cdot u$ $(q \in Q)$, and let $T = \tau(A^*)$. The tm (Q, T) is called the *transition tm* of $\mathcal{T}$.

We now assume that (A, I) is a dependence alphabet, and we consider a sequential transducer $\mathcal{T}$ on A^* with output in a monoid M. We say that $\mathcal{T}$ is *compatible* with (A, I) if, whenever $(a, b) \in I$ and $q \in Q$, we have

$$q \cdot ab = q \cdot ba$$
$$q * (ab) = q * (ba)$$

In that case, there exists a mapping $\sigma\colon \mathbf{M}(A, I) \to M$ such that $\sigma\varphi(u) = \sigma'(u)$ for each $u \in A^*$. The mapping σ is called the *sequential function* from $\mathbf{M}(A, I)$ into M defined by $\mathcal{T}$. The transition morphism τ also extends to $\mathbf{M}(A, I)$: the resulting morphism from $\mathbf{M}(A, I)$ into T is again called the *transition morphism* of $\mathcal{T}$.

The following example describes an important class of sequential functions on trace monoids.

Example 3 *Let (Q, T) be a tm and let $\psi\colon \mathbf{M}(A, I) \to T$ be a morphism. Let $q_0 \in Q$. Let $B = Q \times A$ and let $\sim$ be the congruence on B^* generated by the pairs of the form*

$$(q, a)(\psi(a)[q], b) \sim (q, b)(\psi(b)[q], a), \quad \text{where } q \in Q \text{ and } (a, b) \in I.$$

The sequential transducer associated with ψ and q_0 *is the transducer $\mathcal{T} = (Q, q_0, \cdot, *)$ with output in $B^*/\sim$, where for each $a \in A$, $q \cdot a = \psi(a)[q]$ and $q * a = (q, a)$. It follows from the construction that $\mathcal{T}$ is compatible with (A, I). The corresponding sequential function from $\mathbf{M}(A, I)$ into $B^*/\sim$ is called the* sequential function associated with ψ and q_0.

3.2 The wreath product principle: recognizing $\sigma^{-1}(L)$

Let (A, I) be a dependence alphabet, and let M be a monoid. Let $\mathcal{T} = (Q, i, \cdot, *)$ be a transducer on A^* with output in M which is compatible with (A, I), and let $\sigma\colon \mathbf{M}(A, I) \to M$ be the sequential function defined by $\mathcal{T}$. Finally, let (Q, T) be the transition tm of $\mathcal{T}$ and let $\tau\colon \mathbf{M}(A, I) \to T$ be its transition morphism.

Theorem 4 *With the above notation, if L is a subset of M which is recognized by a tm (P, S), then $\sigma^{-1}(L)$ is recognized by $(P, S) \circ (Q, T)$.*

PROOF. Let $\psi\colon M \to S$ be a morphism, let $p_0 \in P$ and let $F \subseteq P$ be such that $L = \{m \in M \mid \psi(m)[p_0] \in F\}$. Let $\chi'\colon A^* \to S^Q \times T$ be the morphism defined by $\chi'(a) = (\psi_a, \tau(a))$ with

$$\begin{aligned} \psi_a \,:\, & Q \longrightarrow S \\ & q \longmapsto \psi(q * a) \end{aligned}$$

for each letter $a \in A$. It is immediately verified that $\chi'(ab) = \chi'(ba)$ whenever $(a,b) \in I$, since $\mathcal{T}$ is compatible with (A, I). We let $\chi \colon \mathbf{M}(A, I) \to S^Q \times T$ be the induced morphism.

Now let $u = a_1 a_2 \cdots a_n \in A^*$. Then $\chi'(u) = (\psi_u, \tau(u))$ with

$$\begin{aligned} \psi_u[q] &= \psi_{a_1}[q] + \psi_{a_2}[q \cdot a_1] + \cdots + \psi_{a_n}[q \cdot (a_1 \cdots a_{n-1})] \\ &= \psi(q * a_1) + \psi((q \cdot a_1) * a_2) + \cdots + \psi((q \cdot (a_1 \cdots a_{n-1})) * a_n) \\ &= \psi\Big((q * a_1)((q \cdot a_1) * a_2) \cdots ((q \cdot (a_1 \cdots a_{n-1})) * a_n)\Big) \\ &= \psi(q * u). \end{aligned}$$

So $u \in \sigma^{-1}(L)$ if and only if $\psi_u[i] \in F$, if and only if $\chi(u)[(i, 1_T)] \in F \times T$. Thus the wreath product $(P, S) \circ (Q, T)$ recognizes $\sigma^{-1}(L)$, as required.

3.3 The wreath product principle: the converse statement

Let $X = (P, S)$ and $Y = (Q, T)$ be two transformation monoids and let $Z = X \circ Y = (P \times Q, S^Q \times T)$ be their wreath product. Let (A, I) be a dependence alphabet and let $L \subseteq \mathbf{M}(A, I)$ be a trace language recognized by Z. Then, by definition, there exists a morphism ψ from $\mathbf{M}(A, I)$ into $S^Q \times T$, a state (p_0, q_0) of $P \times Q$ and a subset $F \subseteq P \times Q$ such that

$$L = \{u \in \mathbf{M}(A, I) \mid \psi(u)[(p_0, q_0)] \in F\}$$

We would like to express L in terms of languages recognized by either the tm X or the tm Y. For this purpose, let ψ_1 and ψ_2 be the component functions of ψ so that, for any u in $\mathbf{M}(A, I)$, $\psi(u) = (\psi_1(u), \psi_2(u))$ with $\psi_1(u) \in S^Q$ and $\psi_2(u) \in T$. Let also π be the projection of $S^Q \times T$ onto T (so $\pi\psi(u) = \psi_2(u)$ for each $u \in \mathbf{M}(A, I)$).

It is immediate from the definition of the product in $S^Q \times T$ that ψ_2 is a morphism from $\mathbf{M}(A, I)$ into T. In contrast, ψ_1 is not in general a morphism. More precisely, we have

$$\psi_1(uv)[q] \;=\; \psi_1(u)[q] + \psi_1(v)[\psi_2(u)[q]] \text{ for any } u, v \in \mathbf{M}(A, I) \text{ and any } q \in Q.$$

Let σ be the sequential function on $\mathbf{M}(A, I)$, associated with ψ_2 and the initial state q_0. With the notation of Section 3.1, σ is a mapping from $\mathbf{M}(A, I)$ into

$B^*/\sim$, where $B = Q \times A$.

Theorem 5 *If L is a subset of $\mathbf{M}(A, I)$ recognized by the wreath product $X \circ Y$, with $X = (P, S)$ and $Y = (Q, T)$, then L is a finite union of languages of the form $U \cap \sigma^{-1}(V)$ where $U \subseteq \mathbf{M}(A, I)$ is a trace language recognized by Y and V is a subset of $B^*/\sim$ recognized by X, and where $B = Q \times A$, $\sim$ is a particular congruence on B^*, and σ is a particular sequential function from $\mathbf{M}(A, I)$ into $B^*/\sim$.*

PROOF. For any $(p, q) \in P \times Q$, let us denote by $L_{(p,q)}$ the set of traces u such that $\psi(u)[(p_0, q_0)] = (p, q)$. The set F being finite, L is a finite union of languages of the languages $L_{(p,q)}$ for some $(p, q) \in F$. Now, by definition of ψ, $L_{(p,q)}$ is the set of traces u such that $(\psi_1(u)[q_0])[p_0] = p$ and $\psi_2(u)[q_0] = q$. Denote by L_q the set of traces u such that $\psi_2(u)[q_0] = q$, and by L_p the set of traces u such that $(\psi_1(u)[q_0])[p_0] = p$, then $L_{(p,q)} = L_p \cap L_q$. Since ψ_2 is a morphism, L_q is recognized by the tm $Y = (Q, T)$.

It now remains to show that $L_p = \sigma^{-1}(V)$ for some subset V of $B^*/\sim$ which is recognized by the tm X. To this end, we define the morphism $\overline{\psi}_1$ from $B^*/\sim$ into S by

$$\overline{\psi}_1(q, a) \quad = \quad \psi_1(a)[q] \text{ for each } (q, a) \in B.$$

Our first task is to verify that this morphism is well-defined. It suffices to show that if $(a, b) \in I$, then $\overline{\psi}_1(q, a) + \overline{\psi}_1(\psi_2(a)[q], b) = \overline{\psi}_1(q, b) + \overline{\psi}_1(\psi_2(b)[q], a)$ for each $q \in Q$. By definition, we have

$$\overline{\psi}_1(q, a) + \overline{\psi}_1(\psi_2(a)[q], b) = \psi_1(a)[q] + \psi_1(b)[\psi_2(a)[q]] = \psi_1(ab)[q].$$

But $ab = ba$ in $\mathbf{M}(A, I)$, so we have the desired equality.

Now it is straightforward to see that $\psi_1(t)[q_0] = \overline{\psi}_1(\sigma(t))$. Thus L_p is the set of traces u such that $(\overline{\psi}_1(\sigma(u)))[p_0] = p$. Let $V = \{m \in B^*/\sim \mid \overline{\psi}_1(m)[p_0] = p\}$. Since $\overline{\psi}_1$ is a morphism, V is recognized by the tm $X = (P, S)$, and clearly $L_p = \sigma^{-1}(V)$. This concludes the proof.

4 Applications

In this short section, we give a few direct applications of our main result, the wreath product principle for traces. Recall from Example 2 that a monoid S induces in a natural way a transformation monoid (S, S) that can be denoted also by S, as in the following propositions.

Proposition 6 *Let $L \subseteq \mathbf{M}(A, I)$ be recognized by $\{0,1\} \circ Y$, where Y is a transformation monoid. Then L is a boolean combination of languages recognized by Y and of languages of the form $Ka\mathbf{M}(A, I)$, where K is recognized by Y and $a \in \Sigma$.*

PROOF. Using Theorem 5, we know that L is a finite union of languages of the form $U \cap \sigma^{-1}(V)$, where $U \subseteq \mathbf{M}(A, I)$ is recognized by Y and $V \subseteq B^*/\sim$ is recognized by $\{0,1\}$. It is straightforward to prove that $\sigma^{-1}(V)$ a boolean combination of languages of the form $\sigma^{-1}(B^*B_0B^*/\sim)$, where $B_0 \subseteq B$ (remember that σ^{-1} commutes with the boolean operators). Now, we shall prove that $\sigma^{-1}(B^*B_0B^*/\sim)$ is a union of languages of the form $Ka\mathbf{M}(A, I)$, where K is recognized by Y and $a \in \Sigma$. Let $u \in \mathbf{M}(A, I)$. If $u \in \sigma^{-1}(B^*B_0B^*/\sim)$, then $\sigma(u) \sim vbw$, where $v, w \in B^*$ and $b \in B_0$. Thus there exists a word $a_1 \dots a_k$ such that $u = \varphi(a_1 \dots a_k)$ and $\sigma'(a_1 \dots a_k) = vbw$. Let $b = (q, a)$. Hence there is some i such that $a_i = a$ and $\eta_2(\varphi(a_1 \dots a_{i-1}))[q_0] = q$. This shows that $u \in Ka\mathbf{M}(A, I)$ where K is recognized by Y. Reciprocally, if $u \in Ka\mathbf{M}(A, I)$ then $\sigma(u) \in B^*B_0B^*/\sim$.

The two following propositions can be proved in a similar fashion; in the first one, $\mathbf{2}$ denotes the transformation *semigroup* $(\mathbf{2}, \{\alpha, \beta\})$, see Example 1).

Proposition 7 *Let $L \subseteq \mathbf{M}(A, I)$ recognized by $\mathbf{2} \circ Y$, where Y is a transformation monoid. Then L is a boolean combination of languages recognized by Y and of languages of the form Ka, where K is recognized by Y and $a \in \Sigma$.*

Proposition 8 *Let $L \subseteq \mathbf{M}(A, I)$ recognized by $U_2 \circ Y$, where Y is a transformation monoid. Then L is a boolean combination of languages recognized by Y and of languages of the form $K\mathcal{S}L$, where K and L are boolean combinations of languages of the form Na, with N recognized by Y and $a \in \Sigma$, and where $K\mathcal{S}L$ is the language of traces u such that $u = vw$ with $v \in L$ and for every non-empty prefix x of w, $vx \in K$.*

Meyer and Petit have shown, using the wreath product principle, that the temporal logic $PTrL_{red}$ on finite traces (patterned after the temporal logic LTL on words) is expressively complete (i.e. equivalent in expressive power to the first order theory on finite traces). The heart of this proof lies in Proposition 8. This proof, together with a detailed proof of Proposition 8 can be found in [8].

References

[1] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.

[2] W. Ebinger and A. Muscholl. Logical definability on infinite traces. In *Proceedings of ICALP'93*, number 700 in Lecture Notes in Computer Science, 1993.

[3] S. Eilenberg. *Automata, Languages, and Machines*, volume B. Academic Press, New York and London, 1976.

[4] P. Gastin, A. Petit, and W. Zielonka. An extension of Kleene's and Ochmański's theorems to infinite traces. *Theoretical Computer Science*, 125:167–204, 1994.

[5] G. Guaiana, A. Restivo, and S. Salemi. On aperiodic trace languages. In *Proceedings of STACS'91*, number 480 in Lecture Notes in Computer Science, pages 76–88, 1991.

[6] H. Kamp. *Tense Logic and the Theory of Linear Order*. Phd thesis, University of California, 1968.

[7] R. McNaughton and S. Papert. *Counter-free Automata*. MIT Press, 1971.

[8] R. Meyer and A. Petit. Expressive completeness of LTrL on finite traces: an algebraic proof. In *Proceedings of STACS'98*, number 1373 in Lecture Notes in Computer Science, pages 533–543, 1998.

[9] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[10] H. Straubing. Families of recognizable sets corresponding to certain varieties of finite monoids. *Journal of Pure and Applied Algebra*, 15:305–318, 1979.

[11] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Boston, Basel and Berlin, 1994.