

# Linear polyregular functions are regular

Amina, Mikołaj and Nathan

June 5, 2019

## Contents

<b>1 Preliminaries</b>	<b>1</b>
1.1 Polyregular functions . . . . .	1
1.1.1 One pebble transducers . . . . .	1
1.1.2 $k$ -pebble transducers . . . . .	2
1.2 Transition monoids . . . . .	4
<b>2 Deciding if a regular function is in <math>\mathcal{O}(1)</math></b>	<b>5</b>
<b>3 Deciding if a polyregular function is in <math>\mathcal{O}(n^k)</math></b>	<b>6</b>
<b>4 Conclusion</b>	<b>7</b>

## Introduction

**Notation** Throughout the paper, we use the following notations: Given an alphabet  $\Sigma$ , we denote by  $\Sigma^*$  the set of finite words over  $\Sigma$ . The empty word is written  $\epsilon$ . If  $w \in \Sigma^*$ ,  $w[i]$  is the  $i^{\text{th}}$  letter of the word  $w$ .

The powerset of  $E$  is denoted by  $2^E$ .

If  $i, j \in \mathbb{N}$ ,  $[i, j]$  denotes the set  $\{m \in \mathbb{N} \mid i \leq m \leq j\}$ . In particular, if  $i > j$  then  $[i, j] = \emptyset$ .

## 1 Preliminaries

### 1.1 Polyregular functions

#### 1.1.1 One pebble transducers

A one pebble transducer of input alphabet  $\Sigma$  and output alphabet  $\Gamma$  is a two way automaton (meaning that it has a reading head, called here a pebble, which can scan the word on both directions) which reads words over  $\Sigma^*$ , and which has the ability to output words over  $\Gamma^*$  on every transition. A configuration looks like this:

## Picture

The output of a one pebble transducer is the concatenation of the outputs of the transitions it took, in the chronological order of their emission.

**Definition 1.1.** *A 1-pebble transducer is a tuple  $(\Sigma, \Gamma, Q, q_I, q_F, \delta)$ , which consists of:*

- *a finite input alphabet  $\Sigma$  and a finite output alphabet  $\Gamma$ ;*
- *a finite set of states  $Q$ ;*
- *two designated states  $q_I$  and  $q_F$ : the initial and final one;*
- *a transition function of type*

$$\delta : Q \times \Sigma \cup \{\vdash, \dashv\} \rightarrow Q \times \{\rightarrow, \leftarrow\} \times \Gamma^*$$

*The symbols  $\vdash$  and  $\dashv$  are the endmarkers of the word.*

*We assume that the transducer can only move to the right when it is on the left endmarker  $\vdash$ , and only to the left when it is on the right endmarker  $\dashv$ . We also assume that the endmarkers don't output anything, meaning  $\delta(Q \times \{\vdash, \dashv\}) \subseteq Q \times \{\rightarrow, \leftarrow\} \times \{\epsilon\}$ .*

Let us define the behavior of the transducer over an input word  $w \in \Sigma^*$ . The transducer reads the word  $\vdash w \dashv$  and we denote by  $\Sigma_{\vdash \dashv}$  the set  $\Sigma \cup \{\vdash, \dashv\}$ . A *configuration* is a pair  $(q, i)$  where  $q$  is the control state and  $i$  the position of the pebble on  $\vdash w \dashv$ .

Let  $(p, i)$  be a configuration and suppose that  $\delta(p, \vdash w \dashv[i]) = (q, d, \gamma)$ . The *successor* of  $(p, i)$  is the configuration  $(q, j)$  where:

$$\begin{aligned} j &= i + 1 & \text{if} & & d &= \rightarrow \\ j &= i - 1 & \text{if} & & d &= \leftarrow \end{aligned}$$

The *output* of  $(p, i)$  is the word  $\gamma$ . A *run* on  $w$  is a sequence of configurations over  $w$  related by the successor relation defined above. The output of a run is the word obtained by concatenating the outputs of its configurations.

The initial configuration is  $(q_I, 0)$ . The automaton accepts a word if there is an accepting run, i.e. a run where the first configuration is initial, the last one has an accepting state, and no other configuration has an accepting state. The accepting run, if it exists, is unique, by determinism of the transition function.

### 1.1.2 $k$ -pebble transducers

In the literature [1], a  $k$ -pebble transducer is a transducer with  $k$  reading heads. The movement of these heads is subject to a stack discipline: only the pebble on top of the stack can move. Add details, and a precise definition of this version of  $k$ -pebble automata. In this paper, we will work with a different yet equivalent definition of  $k$ -pebble automata. Here a  $k$ -pebble transducer is a collection of  $k$  one-pebble automata. Add an informal definition.

## Picture

Let  $\Sigma$  be a finite alphabet and let  $S$  be a finite set of *labels*, we define the alphabet of labelled letters  $\Sigma(S) = \Sigma \times 2^S$ . For  $S \cap T = \emptyset$ , we identify the sets  $(\Sigma(S))(T)$  and  $\Sigma(S \cup T)$ , as well as the sets  $\Sigma(\emptyset)$  and  $\Sigma$ . Let  $a = (b, T) \in \Sigma(S)$ , and let  $s \in S$ , then we denote by  $a(s)$  the letter  $(b, T \cup \{s\})$ .

**Definition 1.2.** A  $k$ -pebble transducer of input alphabet  $\Sigma$  and output alphabet  $\Gamma$  is a tuple  $\mathcal{T} = \langle T_1, \dots, T_k \rangle$  such that for every  $i \leq k$ :

- $T_i$  is a 1-pebble transducer, whose set of states is denoted  $Q_i$ ;
- The input alphabet of  $T_i$  is  $\Sigma(Q_{>i})$  (with  $Q_{>i} = \bigcup_{j \in [i+1, k]} Q_j$ );
- The output alphabet of  $T_i$  is  $\Gamma \cup [1, i-1]$ .

In particular, the input alphabet of  $T_k$  is  $\Sigma$  and the output alphabet of  $T_1$  is  $\Gamma$ .

For every  $k$ -pebble transducer  $\mathcal{T} = \langle T_j \rangle_{j \in [1, k]}$  and  $i \in [1, k]$ , the sequence  $\langle T_j \rangle_{j \in [1, i]}$  can be seen as an  $i$ -pebble transducer, of input alphabet  $\Sigma(Q_{>i})$  and output alphabet  $\Gamma$ . We denote this transducer by  $\mathcal{T}_i$ .

**Terminology:** A  $k$ -pebble transducer  $\mathcal{T}$  can be seen as the one-pebble transducer  $T_k$ , which outsources a part of the computation to the other one-pebble transducers  $T_i$ ,  $i < k$ . Then  $T_i$  can itself call as subroutines the transducers of lower index, and so on. For this reason, we call *the states of  $\mathcal{T}$*  the states of  $T_k$ , and *the initial state of  $\mathcal{T}$*  the initial state of  $T_k$ .

Let us define the function realized by a  $k$ -pebble transducer.

**Definition 1.3.** We define, by induction on  $k$ , the function realized by a  $k$ -pebble transducer. The case  $k = 1$  has been treated in Definition 1.1.

Consider a  $k+1$  pebble transducer  $\mathcal{T} = \langle T_1, \dots, T_{k+1} \rangle$ , and let  $Q_i$  be the set of states of  $T_i$ , for  $i \in [1, k+1]$ . By induction assumption let  $f_i : \Sigma(Q_{>i})^* \rightarrow \Gamma^*$  be the transduction realized by  $\mathcal{T}_i$ , for  $i \in [1, k]$ . Let us define the image of a word  $w$  of  $\Sigma^*$  by the transduction realized by  $\mathcal{T}$ :

- Let  $r = (q_j, p_j)_{j \in [1, n]}$  be the accepting run of  $T_{k+1}$  over  $w$  and  $(\gamma_j)_{j \in [1, n]}$  be the outputs of the corresponding configurations.
- For every  $j \in [1, n]$ , let  $w_j$  be the word obtained from  $w$  by replacing the letter  $a$  at position  $p_j$  by  $a(q_j)$ . For every  $j \in [1, n]$ , let  $u_j$  be the word obtained from  $\gamma_j$  by replacing each occurrence of a letter  $i \in [1, k]$  by  $f_i(w_j)$ .

The image of  $w$  by  $\mathcal{T}$  is the word  $u_1 \cdots u_n$ .

**Definition 1.4.** Let  $f : \Sigma^* \rightarrow \Gamma^*$  be a word to word function. Its size function,  $|f| : \mathbb{N} \rightarrow \mathbb{N}$  is defined by  $|f|(n) = \max_{|w| \leq n} |f(w)|$ . For  $k \in \mathbb{N}$ , we say that the function  $f$  has degree  $k$  size output if  $|f|(n)$  is equal to  $\mathcal{O}(n^k)$ . Similarly, we say that  $f$  has an output size of degree exactly  $k$  if  $|f|(n) = \Theta(n^k)$ . For  $k = 0, 1, 2, \dots$ , we use the usual terms bounded, linear, quadratic, etc. If there exists  $k$  such that  $f$  has a degree  $k$  size output, then we say that  $f$  has polynomial size output.

**Example:**

Add an example.

□

**Proposition 1.5** ([1]). *The function realized by a  $k$ -pebble transducer has degree  $k$  size output.*

## 1.2 Transition monoids

We present in this section a tool used to summarize the behaviour of one-pebble automata, and called its transition monoid. A chaque mot  $w$ , on associera un element de ce monoid qui nous dira: si j'entre dans le mot  $w$  (par la droite ou par la gauche) en etant dans un certain etat  $q$ , alors je vais sortir du mot par la droite ou la gauche avec un certain etat. Suivant le besoin, on aura besoin d'enregistrer en plus des information sur l'output. Soit le mot emis en entier (c'est le monoid  $M_{\Gamma^*}$ ), ou alors une information binaire qui dit si oui ou non j'ai emis un mot (c'est le monoid  $M_2$ ) ou alors pas du tout d'information sur l'output (c'est le monoid  $M_0$ ).

**Definition 1.6.** *Let  $T$  be a 1-pebble automaton with set of states  $Q$ . Let  $\Delta$  be a monoid, and let  $\star$  be its multiplication.*

*We define the transition monoid  $M_\Delta$  of  $T$  as follows:*

- *its elements are functions of the form  $f : Q \times \{\rightarrow, \leftarrow\} \rightarrow Q \times \{\rightarrow, \leftarrow\} \times \Delta$ ;*
- *the composition  $\cdot$  is defined as follows. Let  $f, g$  be two elements of  $M(\Delta)$ ,  $q \in Q$  and  $d \in \{\rightarrow, \leftarrow\}$ . We define the transition sequence between  $f$  and  $g$  starting from  $(q, d)$  and its output sequence to be respectively the sequences  $(q_i, d_i)_{i \in [0, n]}$  and  $(w_i)_{i \in [1, n]}$  satisfying the following conditions:*
  - $(q_0, d_0) = (q, d)$ ;
  - $d_{n-1} = d_n$  and  $d_i \neq d_{i+1}$  for every  $i \in [1, n-2]$ ;
  - if  $d_0 = \rightarrow$  then for every even  $i$ ,  $f(q_i, d_i) = (q_{i+1}, d_{i+1}, w_{i+1})$  and for every odd  $i$ ,  $g(q_i, d_i) = (q_{i+1}, d_{i+1}, w_{i+1})$ ;
  - if  $d_0 = \leftarrow$  then for every even  $i$ ,  $g(q_i, d_i) = (q_{i+1}, d_{i+1}, w_{i+1})$  and for every odd  $i$ ,  $f(q_i, d_i) = (q_{i+1}, d_{i+1}, w_{i+1})$ .

*We set  $(f \cdot g)(q, d)$  to be  $(q_n, d_n, w_1 \star \dots \star w_n)$ .*

We will mainly instantiate  $M_\Delta$  in the following three cases:

1.  $\Delta$  is the monoid  $\Gamma^*$  of words over  $\Gamma$ .
2.  $\Delta$  is the boolean monoid  $\mathbf{2}$ .
3.  $\Delta$  is the singleton monoid  $\mathbf{0}$ .

In the last case, the third component of the codomain of the elements of  $M_1$  is useless, one can see them as functions of type  $Q \times \{\rightarrow, \leftarrow\} \rightarrow Q \times \{\rightarrow, \leftarrow\}$ .

**Example:**

Show two boxes and their composition.  
Give an example of transition sequence.

□

Let us show define the transition morphisms associated with a 1-pebble transducer.

**Definition 1.7.** Let  $T = (\Sigma, \Gamma, Q, q_I, q_F, \delta)$  be a 1-pebble transducer. We define the morphism  $\mu : (\Sigma_{\rightarrow\leftarrow})^* \rightarrow M_{\Gamma^*}$  as follows:

$$\text{For every } d \in \{\rightarrow, \leftarrow\} \quad \mu(a)(q, d) = \delta(a, q)$$

Let  $\Delta \subseteq \Gamma$ . We define the morphism  $\mu_\Delta : \Sigma^* \rightarrow M_2$  as follows. Let  $1_\Delta : \Gamma^* \rightarrow \mathbf{2}$  be the morphism defined on letters as follows  $1_\Delta(a) = 1$  if  $a \in \Delta$  and  $1_\Delta(a) = 0$  otherwise. We set then for every  $d \in \{\rightarrow, \leftarrow\}$ :

$$\text{If } \delta(a, q) = (q', d', w) \text{ then } \mu(a)(q, d) = (q', d', 1_\Delta(w))$$

We define the morphism  $\mu_0 : (\Sigma_{\rightarrow\leftarrow})^* \rightarrow M_0$  as follows. For every  $d \in \{\rightarrow, \leftarrow\}$ ,

$$\text{if } \delta(a, q) = (q', d', w) \text{ then } \mu(a)(q, d) = (q', d')$$

## 2 Deciding if a regular function is in $\mathcal{O}(1)$

**Lemma 2.1.** Let  $(M, \cdot)$  be a monoid and  $\mu : \Sigma^* \rightarrow M$  be a morphism. Let  $w_1, w_2, w_3 \in \Sigma^*$  such that there exists  $x, y, z, t, e, f \in M$  satisfying:

- $\mu(w_1 w_2) = x \cdot e$  and  $\mu(w_3) = e \cdot y$ ,
- $\mu(w_1) = z \cdot f$  and  $\mu(w_2 w_3) = f \cdot t$ ,
- $e$  and  $f$  are idempotent.

For every  $u, v \in \Sigma^*$  such that  $\mu(u) = e$  and  $\mu(v) = f$  we have that:

- $\mu(w_1 v w_2) = x \cdot e$ ,
- $\mu(w_2 u w_3) = f \cdot t$ .

**Proof**

We have that  $\mu(w_1 v) = z \cdot f \cdot f = z \cdot f = \mu(w_1)$ . Thus  $\mu(w_1 v w_2) = \mu(w_1 v) \cdot \mu(w_2) = \mu(w_1) \cdot \mu(w_2) = \mu(w_1 \cdot w_2) = x \cdot e$ . We proceed in the same way for the other equality. □

**Definition 2.2** (Producing loop). Let  $T = (\Sigma, \Gamma, Q, q_I, q_F, \delta)$  be a 1-pebble transducer, and let  $\Delta \subseteq \Gamma$ . Let  $x, e, y \in M_\Delta$ , with  $xy \in \mu_\Delta(\vdash \Sigma^* \dashv)$ .

We say that the triplet  $(x, e, y)$  is  $\Delta$ -linear if the transition sequence of  $(xe, ey)$   $(q_i, d_i)_{i \in [0, n]}$  satisfies the following conditions:

- $q_0 = q_I$  and  $q_n = q_F$ ;
- $e$  is idempotent i.e.  $e.e = e$ ;
- there exists  $i \in [1, n - 1]$  such that  $e(q_i, d_i)$  is of the form  $(q, d, 1)$ .

**Definition 2.3.** Let  $f : \Sigma^* \rightarrow \Gamma^*$  be a function and  $\Delta \subseteq \Sigma$ . We say that  $f$  is bounded (resp. linear, etc) in  $\Delta$  if  $\pi_\Delta \circ f : \Gamma^* \rightarrow \Delta^*$  is bounded (resp. linear, etc), where  $\pi_\Delta : \Sigma^* \rightarrow \Delta^*$  is the morphism defined on letters as follows:

$$\begin{aligned} \pi_\Delta(a) &= a \text{ if } a \in \Delta \\ &= \epsilon \text{ otherwise.} \end{aligned}$$

**Theorem 2.4.** A 1-pebble transducer is linear in  $\Delta$  if and only if it has a  $\Delta$ -linear triplet.

### 3 Deciding if a polyregular function is in $\mathcal{O}(n^k)$

**Definition 3.1.** Let  $\mu : \Sigma^* \rightarrow M$  be a monoid morphism and let  $w$  be in  $\Sigma^*$ . A  $(k, p)$ -factorization of  $w$  in the monoid  $M$  is given as a tuple of words  $(w_0, v_{1,1}, \dots, v_{1,p}, w_1, \dots, v_{k,1}, \dots, v_{k,p}, w_k)$  verifying:

- $w = w_0 v_{1,1} \cdots v_{1,p} w_1 \cdots v_{k,1} \cdots v_{k,p} w_k$
- for all  $i \in [1, k]$ ,  $\mu(v_{i,1}) = \dots = \mu(v_{i,p}) = \mu(v_{i,1} v_{i,1})$

We say that such a factorization is according to  $(x_0, e_1, x_1, \dots, e_k, x_k)$  if for all  $i \in [0, k]$ ,  $\mu(w_i) = x_i$  and for all  $i \in [1, k]$ ,  $\mu(v_{i,1}) = e_i$ .

**Lemma 3.2.** Let  $\langle T_1, \dots, T_{k+1} \rangle$  be a pebble transducer realizing a function  $f$ , such that  $T_{k+1}$  is bounded in  $\{k\}$ . Then  $f$  can be realized by a  $k$ -pebble transducer.

**Lemma 3.3** (Named Lemma). Let  $\mathcal{T} = \langle T_1, \dots, T_k \rangle$  be a pebble transducer over input alphabet  $\Sigma$  realizing a function  $f$ . There exists a morphism in a finite monoid  $\mu : (\Sigma_{\vdash \dashv})^* \rightarrow M$  and a set  $P \subseteq M^{2k+1}$  such that for all  $p \in \mathbb{N}$ :

- For any  $w \in \Sigma^*$  with a  $(k, p)$  factorization  $(w_0, v_{1,1}, \dots, v_{1,p}, w_1, \dots, v_{k,1}, \dots, v_{k,p}, w_k)$  according to an element of  $P$ ,  $|f(w_1 v_{1,1}^n \cdots v_{k,1}^n w_n)| = \Theta(n^k)$ .
- $f$  restricted to words without  $(k, p)$  factorization according to any element of  $P$  can be realized by a  $k-1$ -pebble transducer.

*Proof.* This is shown by induction on  $k$ . For  $k = 1$ , it is a consequence of the proof of Theorem 2.4.

We assume that the lemma holds for  $k$ , let us show that it holds for  $k+1$ . Let  $\mathcal{T} = \langle T_1, \dots, T_k, T_{k+1} \rangle$  be a pebble transducer realizing a function  $f : \Sigma^* \rightarrow \Gamma^*$ , and let  $\mathcal{T}_k = \langle T_1, \dots, T_k \rangle$ . Let  $f_k : \Sigma(Q_{k+1})^* \rightarrow \Gamma^*$  be the function realized by  $\mathcal{T}_k$ .

Let us apply the induction assumption to  $\mathcal{T}_k$ , and let  $\mu : (\Sigma(Q_{k+1})_{\vdash \vdash})^* \rightarrow M$  and  $P$  be given as in the lemma. For any  $p$ , let  $\mathcal{S}_p$  be a  $k-1$ -pebble transducer realizing the function  $f_k$  restricted to words without any  $(k, p)$  factorization according to elements of  $P$ , and let  $g_p$  denote the function it realizes.

The main idea of the proof is to modify the transducer  $T_{k+1}$  into a new transducer which only outputs  $k$  when it is *absolutely necessary*, i.e. when the word can be factorized in such a way that, by pumping idempotents, one can obtain an output in  $\Theta(n^k)$ . Otherwise, we have according to the lemma that we can outsource the computation to a transducer with only  $k-1$  pebbles.

Let us define a new transducer  $T_{k+1}^p$  which behaves as  $T_k$ , except that at each step where it should output the letter  $k$ , it checks, using some regular look-around if the word has a  $(k, p)$  factorization according to an element of  $P$ . If yes then it outputs  $k$  normally, otherwise, it calls  $\mathcal{S}_p$  instead.

The look-around is implemented by a rational function  $\ell$  which labels each position by additional information. Let  $L = Q_{k+1} \rightarrow \{\mathcal{S}_p, \mathcal{T}_k\}$  be the labelling alphabet, then  $\ell : (\Sigma(Q_{k+1}))^* \rightarrow (\Sigma(Q_{k+1}) \times L)^*$  is defined below: Let  $w \in (\Sigma(Q_{k+1}))^*$ , the word  $z = \ell(w)$  has the same size as  $w$  and  $z[i] = (w[i], h)$  with  $h(q) = \mathcal{T}_k$  if and only if the word obtained by replacing  $w[i]$  with  $w[i](q)$  has a  $(k, p)$  factorization according to an element of  $P$ .

Let

**Claim 3.4.** *Let  $\mu : \Sigma^* \rightarrow M$  be a monoid morphism and let  $w = v_1 v_2 v_3$  be in  $\Sigma^*$ , with  $\mu(v_2) = \mu(v_2 v_2)$ . Let us assume that  $w$  has a  $k$  factorization according to  $t$ , then  $v_1 v_2 v_2 v_3$  also does.*

□

**Theorem 3.5.** *Given a pebble transducer  $\mathcal{T}$  realizing a function  $f$  and a natural number  $k$ , one can decide if  $f$  can be realized by a  $k$ -pebble transducer. In particular, one can decide if a polyregular function is regular.*

**Corollary 3.6.** *A word-to-word function can be defined by an MSO interpretation of dimension  $k$  if and only if it can be realized by a  $k$ -pebble transducer.*

## 4 Conclusion

## References