Linear polyregular functions are regular

Amina, Mikołaj and Nathan

June 7, 2019

Contents

Introduction

Notation Let Σ be an alphabet, we denote by Σ^* the set of finite words over Σ. The empty word is written ϵ. If $w \in \Sigma^*$, $w[i]$ is the i^{th} letter of the word w.

The powerset of E is denoted by $\mathbf{2}^E$.

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

1 Preliminaries

1.1 Polyrational functions

1.1.1 One-pebble transducers

A one-pebble transducer of input alphabet Σ and output alphabet Γ 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 Σ^*, and which has the ability to output words over Γ^* 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 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* Σ *and a finite output alphabet* Γ*;*
- *a finite set of states* Q*;*
- *two designated states* q_I *and* q_F*: the initial and final one;*
- *a transition function of type*

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

  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:

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

The output of (p, i) is the word γ. *A run* on w is a sequence of configurations over w related by a 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 final configuration is $(q_F, |w| + 1)$ The automaton accepts w if there is an accepting run, i.e. a run where the first configuration is initial, the last one is final, and no other configuration is final. The accepting run, if it exists, is unique, by determinism of the transition function.

1.1.2 k-pebble transducers

In the literature [], 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. In this paper, we will work with a different yet equivalent definition of k-pebble transducer. Here a k-pebble transducer is a collection of k one-pebble automata. The idea is that the transducer number k

can, along its run, call transducer $k-1$ to run over its current configuration. Then transducer $k-1$ can itself call transducer $k-2$ to run over its current configuration, and so on.

Picture

Let Σ be a finite alphabet and let S be a finite set of *labels*, we define the alphabet of labelled letters $\Sigma(S) = \Sigma \times \mathbf{2}^S$. For $S \cap T = \varnothing$, we identify the sets $(\Sigma(S))(T)$ and $\Sigma(S \cup T)$, as well as the sets $\Sigma(\varnothing)$ and Σ. 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 Σ and output alphabet Γ is a tuple $\mathcal{T} = \langle T_1, \ldots, T_k \rangle$ such that for every $i \leq n$:*

- *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 Σ and the output alphabet of T_1 is Γ.

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 Γ. 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, $i < k$. A T_i can then itself call as subroutines the transducers of lower indexes, 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 $\mathcal{T} = \langle T_1, \ldots, T_{k+1} \rangle$, and let Q_i be the set of states of T_i, for $i \in [1, k+1]$. By induction let $f_i : \Sigma(Q_{>i})^* \to \Gamma^*$ denote the transduction realized by $\mathcal{T}_i$, for $i \in [1, k]$. Let us define the image of a word w of Σ^* by the transduction realized by $\mathcal{T}$:

- *Let $r = (q_j, p_j)_{j \in [1,n+1]}$ 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 γ_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 $u_1 \cdots u_n$.

Example:

Add an example.

□

Definition 1.4. *Let $f : \Sigma^* \to \Gamma^*$ be a word to word function. Its* size function*, $|f| : \mathbb{N} \to \mathbb{N}$ 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* strictly k *if $|f|(n) = \Theta(n^k)$. For $k = 0, 1, 2, \ldots$, we use the usual terms bounded, linear, quadratic, etc. If there exists k such that f has a degree k size output, then we says that f has polynomial size output.*

Proposition 1.5 ([])**.** *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 (resp. morphism). We map each word w to an element of the monoid which gives the folliwing kind of information: if the automaton enters the word from the left in state q, then it exits to the *e.g* left in state q', *etc.* Moreover, we will sometimes need to record information about the output produced in such a pass of the transducer; this information can be for instance the whole output (in which case the monoid is infinite) or information of the kind "a letter a has been produced at least once" (here we recover finiteness).

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

We define the transition monoind M_N of T as follows:

- *its elements are functions of the form $f : Q \times \{\to, \leftarrow\} \to Q \times \{\to, \leftarrow\} \times N$;*
- *the composition $\cdot$ is defined as follows. Let f, g be two elements of M_N, $q \in Q$ and $d \in \{\to, \leftarrow\}$. 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 $d_0 =\to$ 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 $(q_n, d_n, w_1 \star \cdots \star w_n)$.

We will mainly instanciate M_N in the following three cases:

1. N is the monoid Γ^* of words over Γ.

2. N is the boolean monoid $\mathbf{2}$.

3. N is the singleton monoid $\mathbf{0}$.

In the last case, the third component of the codomain of the elements of $M_{\mathbf{1}}$ is useless, one can be 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 $\mu : (\Sigma_{\vdash\dashv})^* \rightarrow M_{\Gamma^*}$ as follows:*

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

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

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

We define the morphism $\mu_{\mathbf{0}} : (\Sigma_{\vdash\dashv})^* \rightarrow M_{\mathbf{0}}$ 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 exits $x, y, z, t, e, f \in M$ 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 $x, e, y \in M_{\mathbf{2}}$, with $xey \in \mu_\Delta(\vdash \Sigma^* \dashv)$.*

We say that the triplet (x, e, y) is Δ-linear *if the transition sequence of (xe, ey) starting from $(q_0, \rightarrow)$, $(q_i, d_i)_{i \in [0,n]}$ 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 $f : \Sigma^* \to \Gamma^*$ be a function and $\Delta \subseteq \Gamma$. We say that f is bounded (resp. linear, etc) in Δ if $\pi_\Delta \circ f : \Sigma^* \to \Delta^*$ is bounded (resp. linear, etc), where $\pi_\Delta : \Gamma^* \to \Delta^*$ 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 strictly linear in Δ if and only if it has a Δ-linear triple.*

Proof. We know that a one-pebble transducer realizes a linear function, from Proposition 1.5. Let T be a one-pebble transducer realizing a function $f : \Sigma^* \to \Gamma^*$, and let $\Delta \subseteq \Gamma$.

Let us first assume that there exists a Δ-linear triple (m_0, e_1, m_1), and let w be a word such that $\vdash w \dashv$ has a 1 factorization $(w_0, x_1, y_1, z_1, w_1)$ according to this triple. Then we show that since (m_0, e_1, m_1) is Δ-linear, $|f(w_0 x_1 y_2^n z_1 w_1)| = \Theta(n)$. By definition of Δ-linear triple, the output while reading a y_1 factor is non-empty, hence f is strictly linear.

Let us now assume that there are no Δ-linear triples. According to the Factorization Forest Theorem of Simon, there exists an integer d such that any word of length greater than d has a 1-factorization. Let w be a word with a 1-factorization $(w_0, x_1, y_1, z_1, w_1)$. Since there are no Δ-linear triple, $(\mu_\Delta(w_0), \mu_\Delta(y_1), \mu_\Delta(w_1))$ is not Δ-linear. This means that the outputs corresponding to the factor y_1 in the run over w are all empty, and thus we have $|f(w_0 x_1 y_1 z_1 w_1)| = |f(w_0 x_1 z_1 w_1)|$. Hence we have $\{|f(w)| \mid w \in \Sigma^*\} = \{|f(w)| \mid w \in \Sigma^{\leq d}\}$, hence f is bounded. □

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

Definition 3.1. *Let $\mu : \Sigma^* \to M$ be a monoid morphism and let w be in Σ^*. A k-factorization of w in the morphism μ is given as a tuple of words $(w_0, x_1, y_1, z_1, w_1, \ldots, x_k, y_k, z_k, w_k)$ verifying:*

- $w = w_0 x_1 y_1 z_1 w_1 \cdots x_k y_k z_k w_k$
- *for all* $i \in [1,k]$, $\mu(x_i) = \mu(y_i) = \mu(z_i) = \mu(x_i x_i)$

We say that such a factorization is according to $(m_0, e_1, m_1, \ldots, e_k, m_k)$ *if for all* $i \in [0,k]$, $\mu(w_i) = m_i$ *and for all* $i \in [1,k]$, $\mu(x_i) = e_i$.

Lemma 3.2. *Let* $\langle T_1, \ldots, T_{k+1} \rangle$ *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* $\mathcal{T} = \langle T_1, \ldots, T_k \rangle$ *be a pebble transducer over input alphabet* Σ *realizing a function* f*. There exists a morphism in a finite monoid* $\mu : (\Sigma_{\vdash\dashv})^* \to M$ *and a set* $P \subseteq M^{2k+1}$ *such that for all* $p \in \mathbb{N}$*:*

- *For any* $w \in \Sigma^*$ *with a* k *factorization* $(w_0, x_1, y_1, z_1, w_1, \ldots, x_k, y_k, z_k, w_k)$ *according to an element of* P, $|f(w_1 x_1 y_1^n z_1 w_1 \cdots x_k y_k^n z_k w_k)| = \Theta(n^k)$.
- f *restricted to words without* k *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 $\mathcal{T} = \langle T_1, \ldots, T_k, T_{k+1} \rangle$ be a pebble transducer realizing a function $f : \Sigma^* \to \Gamma^*$, and let $\mathcal{T}_k = \langle T_1, \ldots, T_k \rangle$. Let $f_k : \Sigma(Q_{k+1})^* \to \Gamma^*$ be the function realized by $\mathcal{T}_k$.

Let us apply the induction assumption to $\mathcal{T}_k$, and let $\mu : (\Sigma_{k,\vdash\dashv})^* \to M$ and P be given as in the lemma, with $\Sigma_k = \Sigma(Q_{k+1})$. For any p, let $\mathcal{S}$ be a $k-1$-pebble transducer realizing the function f_k restricted to words without any k factorization according to elements of P, and let g 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 R_{k+1} 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 factorization according to an element of P. If yes then it outputs k normally, calling $\mathcal{T}_k$, otherwise it calls $\mathcal{S}$ instead.

The look-around is implemented by a rational function ℓ which labels each position by additional information. Let $L = Q_{k+1} \to \{\mathcal{S}, \mathcal{T}_k\}$ be the labelling alphabet, then $\ell : (\Sigma)^* \to (\Sigma \times L)^*$ is defined below: Let $w \in (\Sigma)^*$, 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 factorization according to an element of P. Let $\Lambda = \Sigma \times L$ in the following.

Claim 3.4. *Let $u = \ell(v)$ be in Λ^*. Let us consider $(w_0, x_1, y_1, z_1, w_1)$ be a 1 factorization of v. Let $u_n = \ell(w_0 x_1 y_1^n z_1 w_1)$, then there exists $\alpha, \beta, \gamma \in \Lambda^*$ such that $u_n = \alpha\beta^n\gamma$, for all $n \in \mathbb{N}$.*

Claim 3.4. Shown using Lemma 2.1. □

From the above claim, we have that we can pump an idempotent of M without affecting the labelling. This will not be sufficient however to obtain our final result. We also want to be able to pump an idempotent without changing the *shape* of a run of R_{k+1}, that is its transitions without regard to the outputs. For this we consider the transition morphism $\mu_{\mathbf{0}} : (\Sigma_{\vdash,\dashv})^*$ of T_{k+1}. Indeed, the shape of a run of R_{k+1} over a word $\ell(v)$ is the same as the shape of a run of T_{k+1} over v. Let $\nu = \mu \times \mu_{\mathbf{0}}$ denote this product.

Let us consider the $\{k\}$-linear triplets of R_{k+1}. We need to show two things, 1) that the function over words whithout $\{k\}$-linear triplets can be done by a $k-1$-pebble transducer, and 2) that otherwise we can find a k-factorization which yields $\Theta(n^k)$ output by pumping idempotents n times. The first can be obtained from Lemma 3.2. The second point is shown using Claim 3.4. □

Theorem 3.5. *A polyregular function is in $\mathcal{O}(n^k)$ if and only if it can be realized by a k-pebble transducer.*

Theorem 3.6. *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 3.7. *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