Linear polyregular functions are regular

Amina, Mikołaj and Nathan

April 12, 2019

1 Introduction

Notation Throughout the paper, we use the following notations: Given an alphabet Σ, 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 $[i, j] = \emptyset$.

2 Preliminaries

2.1 Polyrational functions

2.1.1 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 chronological order of their emission.

Definition 2.1 *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* $\dashv$.

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

Let (p, i) be a configuration and suppose that $\delta(p, w_{\vdash\dashv}[i]) = (q, d, o)$. *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 o. *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 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 configurations have an accepting state. The accepting run, if it exists, is unique, by determinism of the transition function.

2.1.2 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. 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 $k, i \in \mathbb{N}$. We set $\Sigma_{[i,k]} = \Sigma \cup \Sigma \times \mathbf{2}^{[i,k]}$. In the alphabet $\Sigma_{[k,i+1]}$, we have the possibility to mark some letters of Σ by the information of wich pebbles are under head.

Definition 2.2 *A k-pebble transducer of input alphabet* Σ *and output alphabet* Γ *is a tuple* $\mathcal{T} = (T_k, \ldots, T_1)$ *such that for every* $i \leq n$*:*

- T_i *is 1-pebble transducer, whose set of states is* Q_i*;*
- *The input alphabet of* T_i *is* $\Sigma_{[i+1,k]}$*;*

- *The output alphabet of T_i is $\Gamma \cup \bigcup_{1\leq j<i} Q_j$.*

In particular, the input alphabet of T_k is Σ and the output alphabet of T_1 is Γ. We denote by $\mathcal{T}_i$ the transducer T_i.

The alphabet $(\Gamma_{[l,m]})_{[m+1,n]}$ can be seen as the alphabet $\Gamma_{[l,n]}$. Indeed, every letter of $(\Gamma_{[l,m]})_{[m+1,n]}$ of the form $((a,e),f)$ where $e \subseteq [l,m]$ and $f \subseteq [m+1,n]$ can be seen a a letter $(a, e\cup f)$ of $\Gamma_{[l,n]}$ and conversely.

With this in mind, for every k-pebble transducer $\mathcal{T} = (T_j)_{j\in[1,k]}$ and $i \in [1,k]$, the sequence $(T_j)_{j\in[1,i]}$ can be seen as an i-pebble transducer, of input alphabet $\Gamma_{[i+1,k]}$ and output alphabet Γ. Similarly, the sequence $(T_j)_{j\in[i+1,k]}$ can be seen as a $(k-i)$-pebble transducer, of input alphabet Γ and output alphabet $\Sigma \cup \bigcup_{j\in[1,i]} Q_j$. We denote them respectively by $\mathcal{T}[1,i]$ and $\mathcal{T}[i+1,k]$.

Terminology: A k-pebble transducer $\mathcal{T}$ can be seen as the one pebble transducer $\mathcal{T}_k$, which outsources a part of the computation to the other one pebble transducers $\mathcal{T}_i$, $i<k$. For this reason, we call *the states of* $\mathcal{T}$ the states of $\mathcal{T}_k$, and *the initial state of* $\mathcal{T}$ the initial state of $\mathcal{T}_k$.

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

Definition 2.3 *We define, by induction on k the function, realized by a k-pebble automation. The case $k=1$ has been treated in Definition 2.1.*

Consider a $k+1$ pebble transducer $\mathcal{T}$. We set Q_i to be the set of states of $\mathcal{T}_i$. Let us define the image of a word w of Σ^* by the transduction realized by $\mathcal{T}$. Let $r = (q_j, \mathsf{pos}_j)_{j\in[1,n]}$ be the accepting run of $\mathcal{T}_{k+1}$ over w and $(o_j)_{j\in[1,n]}$ be the outputs of the corresponding configurations.
For every $q \in Q_i$, let $f^i_q : \Sigma^*_{[k,i+1]} \to \Gamma^*$ be the transduction realized by $\mathcal{T}[1,i]$, considering q as its initial state.

For every $j \in [1,n]$, let w_j be the word obtained from w by replacing the letter a at position pos_j by $(a, k+1)$. For every $j \in [1,n]$, let u_j be the word obtained from o_j by replacing every state $q \in Q_i$ by $f^i_q(w_j)$.

The image of w by $\mathcal{T}$ is the word $(u_j)_{j\in[1,n]}$.

Example:

Add an example.

□

Proposition 2.4 *The function realized by a k-pebble transducer is $\mathcal{O}(n^k)$ in the lenght of the input word.*

2.2 Transition monoids

We present in this section a tool used to summarize the behaviour of one-pebble automata, and called transition monoids. 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 atat q, alors je vais sortir du mot par la droite ou la 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_{Γ^*}), ou alors une information binaire qui dit si oui ou non j'ai emis un mot (c'est le monoid $M_{\mathbf{2}}$) ou alors pas du tout d'information sur l'output (c'est le monoid $M_{\mathbf{0}}$).

Definition 2.5 *Let T be a 1-pebble automaton with set of states Q. Let Δ be a monoid, and let $\star$ be its multiplication.*

We define the transition monoind M_Δ 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:*
  - $(q_0, d_0) = (q, d)$;
  - $d_0 = d_1$, $d_{n-1} = d_n$ *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* $d_0 =\rightarrow$ *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 $(q_n, d_n, w_0 \star \cdots \star w_n)$.

We will mainly instanciate M_Δ in the following three cases:

1. Δ is the monoid Γ^* of words over Γ.
2. Δ is the boolean monoid **2**.
3. Δ is the singleton monoid **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 how to interpret words as elements of the monoid

Definition 2.6 *Let $T = (\Sigma, \Gamma, Q, q_I, q_F, \delta)$ be a 1-pebble transducer. We define the morphism $\mu : \Sigma^* \rightarrow M_{\Gamma^*}$ as follows:*

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

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

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

We define the morphism $\mu_{\mathbf{0}} : \Sigma^* \to M_{\mathbf{0}}$ as follows. For every $d \in \{\rightarrow, \leftarrow\}$,

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

3 Deciding if a regular function is bounded

Lemma 3.1 *Let $(M, \cdot)$ be a monoid and $\mu : \Sigma^* \to M$ 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 3.2 (Producing loop) *Let $T = (\Sigma, \Gamma, Q, q_I, q_F, \delta)$ be a 1-pebble transducer, and let $\Delta \subseteq \Gamma$. Let $x, e, y \in \mathsf{Im}(\mu_\Delta(\vdash \Sigma^* \dashv))$.*

We say that the triplet (x, e, y) is Δ-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 3.3 *Let $f : \Gamma^* \to \Sigma^*$ be a function and $\Delta \subseteq \Sigma$. We say that f is linear in Δ if $\pi_\Delta \circ f : \Gamma^* \to \Delta^*$ is linear, where $\pi_\Delta : \Sigma^* \to \Delta^*$ is the morphism defined on letters as follows:*

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

Theorem 3.4 *A 1-pebble transducer is linear in Δ if and only if it has no Δ-linear triplet.*

4 Deciding if a polyregular function is regular

Lemma 4.1 *Let $\mathcal{T} = \langle T_k, \dots, T_1\rangle$ be a k-pebble transducer over input alphabet Σ. One can obtain a new k-pebble transducer $\mathcal{T}' = \langle T'_k, \dots, T'_1\rangle$ over $\Sigma' = \Sigma \times L$ such that:*

- *for any $w \in \Sigma'^*$, $\mathcal{T}'(w) = \mathcal{T} \circ \pi_\Sigma(w)$*
- *For any $\bigcup_{j<k} Q'_j$-producing triple (w_1, w_2, w_3), we have $\mathcal{T}'(w_1 w_2^k w_3) = \Theta(|w_2|^k)$*

5 Conclusion

References