

Transducers

Session 12: Growth Rate

Version: v0.0.10

Aliaume LOPEZ
TA mail* Course page[†] Exercises page[‡]

June 24, 2024

1 Deciding Growth of Unary Output

The goal of this longer exercise is to prove the following statement relating the **growth rate** of a function to its **dimension** in a specific case.

Let f be an **N-polyregular function**. The following are equivalent

1. f is computed by an **MISO**-interpretation of dimension d .
2. The growth rate of f is $O(n^d)$.

Exercise 1 (Playing with unary output). Prove the following equivalence:

1. f is **N-polyregular** (resp. **Z-polyregular**) of dimension d
2. f is a **Z**-linear combination (resp. **N**-linear combination) of functions of the form $\#\varphi$ where φ is an **MISO**-formula with at most d variables.

Exercise 2 (A non-result). Is the following function **N-polyregular**? **Z-polyregular**?

$$f(w) := (|w|_a - |w|_b)^2 \quad .$$

Exercise 3 (Simple operations). Let f and g be two **N-polyregular** functions. Prove that $f \times g$ is **N-polyregular**, where $f \times g$ is defined by $(f \times g)(w) := f(w) \times g(w)$. Similarly, prove that $f \otimes g$ is **N-polyregular**, where $f \otimes g$ is defined by $(f \otimes g)(w) := \sum_{uv=w} f(u) \times g(v)$.

Exercise 4 (Alternative definition). Prove that for all $n \in \mathbb{N}$, we have $\mathbf{Z}\text{-Poly}_{n+1} = \mathbf{Span}(\{\mathbf{1}_L \otimes f \mid f \in \mathbf{Z}\text{-Poly}_n\})$.

Exercise 5 (Polyregular Functions and Matrices). Let f be a **Z-rational series** defined by a minimal representation (I, μ, F) . Let $w \in \Sigma^*$ and $\lambda \in \mathbf{Spec}(\mu(w))$. There exist coefficients $\alpha_{i,j} \in \mathbb{C}$ and words $u_1, v_1, \dots, u_n, v_n \in \Sigma^*$ such that

$$\lambda^X = \sum_{i,j=1}^n \alpha_{i,j} f(v_i w^X u_j) \quad \forall X \geq 0 \quad .$$

To prove the statement, you can use the fact that in a minimal representation, and therefore that $\mathbf{Span}_{\mathbb{Q}}(\{\mu(w)F \mid u \in \Sigma^*\})$ equals \mathbb{Q}^n . Similarly, for a minimal representation, $\mathbf{Span}_{\mathbb{Q}}(\{I\mu(w) \mid u \in \Sigma^*\})$ equals \mathbb{Q}^n .

*ad.lopez@uw.edu.pl

[†]<https://www.mimuw.edu.pl/~bojan/2023-2024/przekształcenia-automatowe-transducers>

[‡]<https://aliaumel.github.io/transducer-exercices/>

Exercise 6 (Series of Polynomial Growth). Let f be a \mathbb{Z} -rational series of growth rate $O(n^d)$. Prove that f is \mathbb{Z} -polyregular of dimension d .

You may use the following corollary of [BR10, Corollary 2.6 page 159] stating

Let f be a \mathbb{Z} -rational series of growth rate $O(n^d)$. Then f belongs to span of the products of at most $q + 1$ characteristic series of rational languages.

Exercise 7 (Ultimately polynomial functions). Let f be a function from Σ^* to \mathbb{Z} . Prove that the following are equivalent:

1. f is a \mathbb{Z} -rational series that has polynomial growth,
2. f is a \mathbb{Z} -rational series that is ultimately N -polynomial,
3. f is a \mathbb{Z} -rational series with eigenvalues in $\mathbb{U} \cup \{0\}$,
4. f is a \mathbb{Z} -rational series with eigenvalues in $B(0, 1)$.

Where ultimately N -polynomial means that for all pumping patterns, there exists a polynomial P such that

$$f(\alpha_0 w_1^{N X_1} \alpha_1 w_2^{N X_2} \dots \alpha_{n-1} w_n^{N X_n} \alpha_n) = P(X_1, \dots, X_n) \quad \text{when the } X_i \text{'s are large enough} \quad .$$

You may use the following corollary of [Bel05, Theorem 2.6] stating

Let A be a finite set of $d \times d$ complex matrices and $\langle A \rangle$ be the semigroup generated by A . Assume that all the eigenvalues of the elements of $\langle A \rangle$ are in or on the unit disc. Then for any matrix norm $\|\cdot\|$, and product P of at most n matrices in A , $\|P\| = O(n^{d-1})$.

Exercise 8 (Growth of Unary Output). Let f be an N -polyregular function. Prove the following equivalences:

1. f is computed by an MISO-interpretation of dimension d ,
2. f is not $(d + 1)$ -pumpable,
3. f has growth rate $O(n^d)$.

Where a function is d -pumpable if there exists a family of words such that

$$f(\alpha_0 w_1^{X_1} \alpha_1 w_2^{X_2} \dots \alpha_{n-1} w_n^{X_n} \alpha_n) = \Omega(|X_1 + \dots + X_n|^d) \quad .$$

To that end, you may use the forest factorisation theorem of Simon, which states that for all N -polyregular functions f , there exists a regular function g from Σ^* to trees of bounded depth such that $f(w) = h \circ g(w)$ where h counts the number of patterns of certain shapes in the tree.

Exercise 9 (Growth when negative values are allowed). Let f be a \mathbb{Z} -polyregular function. Prove the following equivalences:

1. f is computed by an MISO-interpretation of dimension d ,
2. f is not $(d + 1)$ -pumpable,
3. f has growth rate $O(n^d)$.

Use the following decomposition of f into $f_{\text{dep}} + f_{\text{indep}}$ where f_{dep} counts independent tuples of variables, and f_{indep} counts dependent tuples of variables. Then, conclude by induction.

2 Compression

Exercise 10 (Straight line program evaluation). Let e be the evaluation function from [straight line programs](#) to strings. Is e a polyregular function?
 ▷ Solution 1 (Solution)

Exercise 11 (Efficient compression). What is the minimal size (number of instructions) needed to express $\text{prefixes}(a^n)$ as a [straight line program](#)?
 ▷ Hint 1
 ▷ Solution 2 (Solution)

Exercise 12 (Straight-line homomorphic programs). A function f is straight-line homomorphic if there exists a polynomial time algorithm P such that for all straight line program X , $f(e(X)) = e(P(X))$, where e is the expansion function.

1. Prove that **prefixes** is not straight-line homomorphic.
2. Prove that [regular functions](#) are straight-line homomorphic.

▷ Solution 3 (Proof of the first implication)
 ▷ Solution 4 (Sketch of the second implication using Monoids)
 ▷ Solution 5 (Proof of the second implication using Krohn–Rhodes decompositions)

3 Cheat-Sheet

Definition 1 (Rational Series). Recall that a [rational series](#) is a triple (I, μ, F) where I is a vector of initial values, μ is a matrix of transitions, and F is a vector of final values. The function f associated to the rational series is defined by

$$f(w) := I\mu(w)F \quad .$$

If the coefficients are restricted to belong to \mathbb{N} , \mathbb{Z} or \mathbb{Q} , we call them respectively [N-rational series](#), [Z-rational series](#), and [Q-rational series](#).

Definition 2 (Growth rate). Let $f: \Sigma^* \rightarrow \Gamma^*$, its [growth rate](#) is a function that maps $n \in \mathbb{N}$ to the maximal value of $|f(w)|$ for $w \in \Sigma^n$.

Definition 3 (Dimension). In this exercise sheet, an interpretation of [dimension \$d\$](#) is an **MISO**-interpretation where the maximal arity of the polynomial functor describing the domain is d .

Definition 4 (Z-Polyregular Functions). A function f is [Z-polyregular](#) if there exists a polyregular function $g: \Sigma^* \rightarrow \{-1, +1\}^*$ such that $f(w) = \sum(g(w))$ for all $w \in \Sigma^*$. If the function g has outputs in $\{1\}^*$, then it is called [N-polyregular](#). Furthermore, if the function g is star-free (i.e., first-order) then we say that f is [Z-star-free](#) (resp. [N-star-free](#)).
 This provides us with the following table of functions

	$\{-1, +1\}^*$	$\{1\}^*$
Polyregular	Z-polyregular (Z-Poly)	N-polyregular (N-Poly)
Star-free	Z-star-free (Z-SF)	N-star-free (N-SF)

Definition 5 (Straight line program). A [straight line program](#) over an alphabet Σ is a finite sequence of instructions of the form $x_i := u$ where u is a single letter, or $x_i := x_j x_k$ with $i > j, k$. The value of a straight line program is the value of the last variable.

References

- [Bel05] Jason P. Bell. "A gap result for the norms of semigroups of matrices". In: *Linear Algebra and its Applications* 402 (2005), pp. 101–110. ISSN: 0024-3795. DOI: <https://doi.org/10.1016/j.laa.2004.12.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0024379504005300>.
- [BR10] Jean Berstel and Christophe Reutenauer. *Noncommutative Rational Series with Applications* (Encyclopedia of Mathematics and its Applications). 1st ed. *Encyclopedia of Mathematics and its Applications* 137. Cambridge University Press, 2010. ISBN: 0521190223,9780521190220. DOI: 10.1017/CB09780511760860.002.

A Hints

Hint 1 (Exercise 11 What about variables containing two hashes?). Let x_i be a variable in a [straight line program](#) that evaluates to $\text{prefixes}(a^n)$ and that contains a string with two hashes. Can it be used twice?

B Solutions

Solution 1 (Solution to Exercise 10). The function e is not polyregular, because e can have exponential size output, for instance by compressing a^n into $O(\log(n))$ operations.

Solution 2 (Solution to Exercise 11). We prove that a [straight line program](#) that evaluates to a factor of $\text{prefixes}(a^n)$ having k hashes must contain at least k variables evaluating to words containing at least two hashes.

Let us first remark that every variable in the straight line program containing more than one hash can only be used once, as otherwise the output word contains two hash-separated words of the same size, which contradicts the definition of $\text{prefixes}(a^n)$.

For the base case, if the factor contains at least one hash, then the [straight line program](#) must contain at least one instruction.

For the induction, let x_ℓ be the last variable of the straight line program. It cannot be a constant assignment because the factor contains at least two hashes. Therefore, $x_\ell = x_i x_j$ for some $i, j < \ell$. By the induction hypothesis, x_i and x_j . If x_i contains at least two hashes, then it can only be used once, and $x_i \neq x_j$, otherwise, x_j contains a factor of $\text{prefixes}(a^n)$ with at least k hashes, but of smaller size and we proceed by induction. Now, because x_i and x_j are distinct variables, and because every variable containing at least two hashes can only be used once, we can partition the variables of the straight line program into three sets: the ones containing at most one hash, the ones containing at least two hashes used to build x_i , and the ones containing at least two hashes used to build x_j . Using the induction hypothesis we conclude as desired.

Solution 3 (Solution to Exercise 12). It is clear that prefixes is not straight-line homomorphic because $\text{prefixes}(a^n)$ cannot be compressed in less than n instructions, while a^n can be compressed in $O(\log(n))$ instructions. If prefixes were straight-line homomorphic, then the compression of $\text{prefixes}(a^n)$ would be doable in $O(P(\log(n)))$ instructions.

Solution 4 (Solution to Exercise 12). For the second part. Let f be computed by a copyless SST with states Q , registers $\{1, \dots, n\}$, and output function F . The sketch of the proof is as follows: we construct the transition monoid of the SST, and prove that we can encode it inside a straight line program.

To a word w we associate the function from $Q \times (\Gamma^*)^n \rightarrow Q \times (\Gamma^*)^n$ that maps a state q and a tuple of registers r to the state q' and the tuple of registers r' that the SST would be in after reading w starting in state q and with registers r . Note that this monoid is finitely generated, but is not finite, because of the unbounded values in the registers.

However, the program is copyless, meaning that the value in a register is a function of the form $r'_i \leftarrow \alpha_0 r_{j_0} \alpha_1 r_{j_1} \dots$ where the α 's are constants in Γ^* , and the indices j_k are distinct. In particular, since we can only use each register once, we have at most $n + 1$ words $\alpha_0, \dots, \alpha_n$ in the update function.

Now, we will encode the transition of a word w in a straight line program by using variables $y_{i,q,r}$ that encode the i th word α in the update function of the register r after reading w if starting in state q . These can be composed to simulate the transition of the SST on the input word.

This new straight-line program is constructed in polynomial time.

Solution 5 (Solution to Exercise 12). Remark that functions that can be efficiently evaluated over compressed strings are closed under composition. Therefore, it is enough to prove that the basic functions in a Krohn–Rhodes decomposition can be evaluated efficiently.

It is clear that any homomorphism can be efficiently evaluated. For mealy machines, it is very simple, as one can encode the transition monoid inside the decomposition. For map-duplicate and map-reverse, a simple straight-line program can be constructed.