Growth properties of power-free languages

Author: Arseny M. Shur

November 16, 2021

Table of contents

Terminology (formal languages)

Basic nomenclature

- alphabet (Σ) - nonempty set of letters
- word - sequence of letters, $w = a_1 a_2 ... a_n$, λ - empty word
- factor - subword u of word w: $w = xuz$
- set of all words over Σ: Σ^*
- language L over alphabet Σ: $L \subseteq \Sigma^*$
- factorial language - closed under taking factors of its words
- regular language - accepted by some DFA

Terminology (formal languages)

Forbidden word

A word w is forbidden in a language L if words from L don't contain w as a factor

Antidictionary of L

Set of all minimal forbidden words in a language L.
Factorial language defined on Σ^* is determined by it's antidictionary M:
$L = \Sigma^* - \Sigma^* \cdot M \cdot \Sigma^*$

Graph index

Frobenius root

Frobenius root of a nonnegative matrix A is its maximal absolute value eigenvalue.

Graph index

Every digraph G has corresponding adjacency matrix A.
Frobenius root for A is called index of G and marked by Ind(G).

Morphism

Morphism definition

function $h : \Sigma_1^* \to \Sigma_2^*$, $h(uv) = h(u)h(v)$

Example: Thue-Morse morphism

$\theta : \{a, b\}^* \to \{a, b\}^*$
$a \to ab$
$b \to ba$

β-power of a word

$$w^{\beta} = \underbrace{w \cdot \ldots \cdot w}_{\lfloor \beta \rfloor} \cdot w'$$

$\frac{|w^{\beta}|}{|w|} \geq \beta$, $\frac{|w^{\beta}|-1}{|w|} < \beta$

β^{+}-power of a word

$$w^{\beta^{+}} = \underbrace{w \cdot \ldots \cdot w}_{\lfloor \beta \rfloor} \cdot w'$$

$\frac{|w^{\beta +}|}{|w|} > \beta$, $\frac{|w^{\beta +}|-1}{|w|} \leq \beta$

$\beta(\beta^+)$-free word - word which doesn't contain factor u^β for some u
Example:
2-free word - does not contain subfactor $w \cdot w$
2^+-free word - can contain subfactor $w \cdot w$, but cannot contain $w \cdot w \cdot a$, where a is a first letter of w as a subfactor
$\beta(\beta^+)$-free language - language of all $\beta(\beta^+)$-free words over a given alphabet

Repetition Threshold (RT(k))

RT(k) - defined for languages over alphabets of size k.

Definition

Infimum of all numbers β such that the k-ary β-free language is infinite

Example

RT(2) = 2

All words in binary alphabet without squares: a, b, ab, ba, aba, bab. So any 2-free language is finite

Language generated by Thue-Morse sequence does not contain any power greater than 2 as a factor:

a, ab, abba, abbabaab, ...

So 2 is the infimum of inifinite β-power free languages

$\mathrm{C}_L(N)$

Combinatorial complexity function. $C_L(n)$ - amount of words in language L that have length n

Gr(L)

Growth rate of a language L

$Gr(L) = lim_{n\to\infty}[C_L(n)]^{\frac{1}{n}}$

Gr(L) $>$ 1: exponential complexity, big language

Gr(L) = 1: subexponential complexity, small language

Gr(L) $<$ 1, then Gr(L) = 0: degenerate, finite language

Two-dimensional representation of the set of all power-free languages.

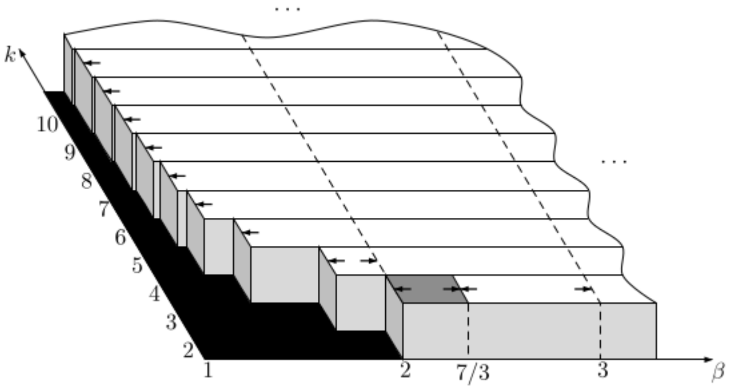


Image source: Growth properties of power-free languages, Arseny M. Shur, Fig. 1

Small languages: polynomial plateau

Gr(L) = 1

Exponential Conjecture

For every $k \geq 3$ k-ary threshold language has exponential complexity. (Confirmed for $k \leq 10$)

Overlap-Free languages (2^+-free)

Lemma 1

Let w be an overlap-free word. Then w can be obtained from a θ-image of some word u by applying one transformation from each of the two sets:
(a) delete the first letter/replace the first letter by the other letter/do nothing;
(b) delete the last letter/replace the last letter by the other letter/do nothing.
Moreover, if $|w| \geq 7$, then this pair of transformations is unique, and if $|w|$ is odd, then u must be overlap-free.
w is almost a θ-image of u.

Overlap-Free languages (2^+-free)

Lemma 1 can be used to obtain upper bounds for number of OF words. Let $|u| = n$. Let's estimate the number of almost θ-images of u of length 2n + 1.

1) We need to add one letter at one end and replace or do nothing at the other end - the upper bound is 8

2) It can be shown, that if a and b can be added to one end, then it's not possible to change letter at that and - the upper bound is 5

C_{OF} bound

$C_{OF}(2n) < C_{OF}(2n+1) \leq 5 \cdot C_{OF}(n)$

$C_{OF}(n) = O(n^{log5})$

Overlap-Free languages (2^+-free)

It turns out that there is no such α that $C_{OF}(n) = \Theta(n^\alpha)$

Theorem 1

(1) $\alpha = lim\ inf_{n\to\infty} \frac{log\ C_{OF}(n)}{logn} \in [1.2690, 1.2736]$

(2) $\beta = lim\ sup_{n\to\infty} \frac{log\ C_{OF}(n)}{logn} \in [1.3322, 1.3326]$

(3) The ratio $\frac{log\ C_{OF}(n)}{logn}$ tends to a limit σ as n appreaches infinity along some subset $N' \subset N$ of density 1 and
$\sigma \in [1.3005, 1.3098]$

Languages other than (2^+-free) from the plateau

Equivalent of Theorem 1 was proved for $\beta \in [2^+, \frac{7}{3}]$.

Conclusion

Growth rate of Small languages can be estimated with rather high accuracy.

Big languages - upper bounds

Basic idea

To compute upper bound for factorial language L one can take it's simple to compute superset L' and compute it's complexity. Languages with a finite antidictionary (FAD-languages) is considered to be simple. To estimate complexity of a factorial language L and antidictionary M one can take the sequence of subsets $\{M_i\}$ of M and calculate growth rate of languages defined by antidictionaries $\{M_i\}$.

M_i - set of all words from M of period $\leq$ i

$$M_1 \subseteq M_2 \subseteq ... \subseteq M, \ \bigcup_{i=1}^{\infty} M_i = M$$

$$L \subseteq ... \subseteq L_i \subseteq ... \subseteq L_1, \ \bigcap_{i=1}^{\infty} L_i = L$$

$$lim_{i \to \infty} Gr(L_i) = L$$

Big languages - upper bounds

Problems to solve

- for Language L and integer i, calculate antidictionary M_i
- calculate growth rate of a language defined by antidictionary M_i.

Note: above approach is not usable for small languages

Find growth rate of FAD-language

Generating function

Use generating function for combinatorial complexity. FAD-languages have rational generating functions. The least positive pole of generating function is the reciprocal of the growth rate of corresponding language. Drawback: Consumes lot's of resources.

Finite automata

Growth rate of a FAD-language L is equal to index of deterministic finite automaton (DFA) accepting L. Far more effective than generating function approach.

Fast calculation of index of digraph

Simple iteration method

Method finds Frobenius root of adjacency matrix of a digraph.
Method:

1. Choose initial vector x
2. Calculate sequence $x, Ax, ..., A^n x$
3. Return $\frac{|A^n x|}{|A^{n-1} x|}$

Building smaller digraphs from FAD

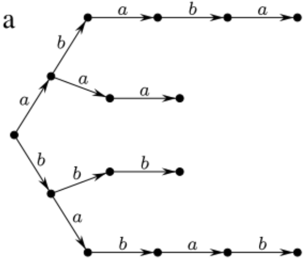
Motivation

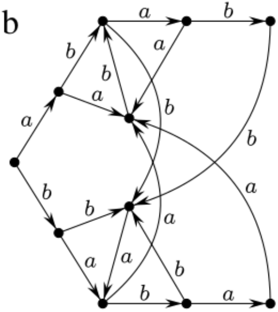
Having a digraph G for a dictionary it's easier to compute index of a graph for a smaller graph with the same index.

Idea

1. Having antidictionary M create a trie containing words of M
2. Convert trie into Aho-Corasick automaton accepting language defined by antidictionary M
3. Instead of trie and Aho-Corasick automaton we can use factor trie and factor automaton

From trie to Aho-Corasick

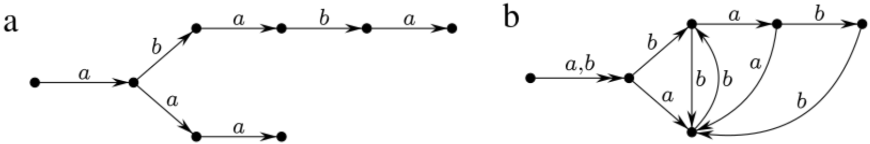




Antidictionary: $\{aaa, bbb, ababa, babab\}$

Image source: Growth properties of power-free languages, Arseny M. Shur, Fig. 2

From factor trie to factor automaton



Antidictionary: $\{aaa, bbb, ababa, babab\}$

Image source: Growth properties of power-free languages, Arseny M. Shur, Fig. 3

Aho-Corasick automaton vs factor automaton

Size

Factor automaton size is reduce by Σ!

lexmin words

$M' \subset M$ - all words that are lexicographically minimal if we treat words w and u as equal if w=h(u) for some morphism h.

Example

Lexmin words for $\{aaa, bbb, ababa, babab\}$ are $\{aaa, ababa\}$

Big languages - lower bounds

Theorem 2

Suppose that k and i are positive integers, $\beta \geq 2$, L is the k-ary β-free language, M_i is the set of all words of period $\leq$ i from the antidictionary of L, L_i is the regular approximation of L with the antidictionary M_i , and the factor-graph $F(M_i)$ is almost strongly connected. Then any number γ such that

$$\gamma + \frac{1}{\gamma^{i-1}(\gamma - 1)} \leq Gr(L_i)$$

satisfies the inequality $\gamma < Gr(L)$.
(G is almost strongly connected - all but one strongly connected components have one vertex

Big languages - lower bounds

What if $\beta < 2$?

Existing methods require lots of computing and are prone to approximation errors.
Example: Computations for 10-letter alphabet required 176GB of space

Big languages: asymptotic formulas: $\beta \geq 2$

$\alpha(k, \beta)$ - two-variable function representing growth rate

Theorem 3

Let $p \geq 2$ be an integer, $\beta \in [p^+, p+1]$. Then the following equality holds:

$$\alpha(k,\beta) = \begin{cases} k - \frac{1}{k^{p-1}} + \frac{1}{k^p} - \frac{1}{k^{2p-2}} + O(\frac{1}{k^{2p-1}}), \ if \beta \in [p^+, p+\frac{1}{2}] \\ k - \frac{1}{k^{p-1}} + \frac{1}{k^p} + O(\frac{1}{k^{2p-1}}), \ if \beta \in [(p+\frac{1}{2})^+, p+1] \end{cases}$$

Big languages: asymptotic formulas: $\beta < 2$

Conjecture

The following equalities hold for any fixed integers p, k such that $k > p \geq 3$:

$$\alpha(k, \frac{p}{p-1}^{+}) = k + 2 - p - \frac{p-1}{k} + O(\frac{1}{k^2})$$

$$\alpha(k, \frac{p}{p-1}) = k + 1 - p - \frac{p-1}{k} + O(\frac{1}{k^2})$$

Big languages: asymptotic formulas

Growth Rate Conjecture

Growth rates of k-ary threshold languages tend to the limit $\alpha_0 \approx 1.242$ as k approaches infinity.

Summary

$\beta \geq 2$

- we can calculate growth rate of each language with arbitrary precision (resource need is not too high)
- asymptotic behaviour of growth rate of a language is described by formulas up to $O(\frac{1}{k^3})$ term
- languages or polynomial plateau have described combinatorial complexity and minimal, maximal and typical growth rate of smallest and biggest language from the plateau are calculated

Summary

$\beta < 2$

- there is an algorithm that can calculate lower bounds for the growth rate of a language, but requires lots of resources
- asymptotic behavior of a language is described by a conjecture

Summary

Future study

- prove Exponential Conjecture
- prove Conjecture about asymptotic behavior of languages where $\beta < 2$
- prove Growth Rate Conjecture

Thank you

Sources

1. Thue Morse sequence
2. Square-free word
3. Growth rates of power-free languages