Open Problems Column
Edited by William Gasarch

This Issues Column! This issue's Open Problem Column is by Lance Fortnow and William Gasarch and is *The CFG Complexity of Singleton Sets*.

Request for Columns! I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere inbetween, and (2) really important or really unimportant or anywhere inbetween.

The Context-Free Complexity of Singleton Sets

Lance Fortnow*
Illinois Institute of Technology
lfortnow@iit.edu

William Gasarch†
University of Maryland at College Park
gasarch@umd.edu

March 26, 2024

1 Introduction

For a string w, how hard is it to recognize w. For Turing machines, this relates to Kolmogorov complexity, where we know most w require a program of length nearly $|w|$ to find w. However there is no computational process that will find an infinite set of such w.

In this paper, we look at similar questions but using context-free grammars as our computational device.

First, some preliminaries.

Notation 1.1

1. Let $w \in \{0,1\}^*$. Then $L_w = \{w\}$.
2. DFA means Deterministic Finite Automata.
3. NFA means Nondeterministic Finite Automata.
4. The *size* of a DFA or NFA is the number of states.
5. CFG means Context Free Grammar. We will assume that all CFG's are in Chomsky Normal Form (which we define later).
6. The *size* of a CFG is the number of rules.

The following is easy to show.

*College of Computing, Ill. Inst. of Tech., IL 60616
†Dept. of Comp. Sci., Univ.of Maryland, MD 20742

Theorem 1.2 *Let $w \in \{0,1\}^*$. Let $n = |w|$.*

1. *There is a DFA for L_w of size $n+1$.*
2. *Any DFA for L_w requires size $\ge n+1$.*
3. *There is an NFA for L_w of size n.*
4. *Any NFA for L_w requires size $\ge n$.*
5. *There is a regular expression for L_w of size n.*
6. *Any regular expression for L_w has size $\ge n$.*

What about the sizes of CFG's for L_w? In order to make the question about size of CFG's interesting we restrict to CFG's in Chomsky Normal Form.

Def 1.3 A CFG is in *Chomsky Normal Form* if every rule is in one of the following forms:

1. $A \rightarrow BC$ where A, B, C are all nonterminals.
2. $A \rightarrow \sigma$ where A is a nonterminal and $\sigma \in \Sigma$.
3. $S \rightarrow e$ where S is the start symbol and e is the empty string.

Henceforth all CFG's are assumed to be in Chomsky Normal Form.
The following are easy to show.

Theorem 1.4

1. *Let $w = 0^n$. There is a CFG of size $O(\log n)$ for L_w.*
2. *Assume n is divisible by d. Let $w = 0^{n/d}1^{n/d}0^{n/d} \cdots 0^{n/d}1^{n/d}$. There is a CFG of size $O(\log n)$ for L_w (independent of d).*
3. *(Informal Statement) Let $s_1, \ldots, s_k$ be your favorite sequence of natural numbers. Assume k is even (for k odd a similar theorem holds). Let $n = s_1 + \cdots + s_k$. Let $w = 0^{s_1}1^{s_2} \cdots 0^{s_k}1^{s_k}$. There is a CFG of size $O(\sum_i \log s_i) \le O(k \log n)$ for L_w.*

The following questions arise:

- Is there a string w such that any CFG for L_w is large (for some definition of large). (Spoiler Alert: Yes.)
- Is there a *natural string* w such that any CFG for L_w is large (for some definitions of natural and large). Theorem **??** can be considered a failed attempt at getting a natural string w such that L_w requires a large CFG.

We will need the following easy lemma.

Lemma 1.5 *Let w be a string of length n. There exists a CFG G of size $\le n + |\Sigma| - 1$ such that $L(G) = L_w$. Note that for a fixed alphabet the CFG is of size $n + O(1)$—not $O(n)$.*

Proof:

For simplicity, assume $\Sigma = \{0, 1\}$. The proof easily generalizes to larger alphabets.
Let $w = w_1 \cdots w_n$.
Here is the CFG for L_A.
$S \rightarrow A_{w_1} R_2$
$R_2 \rightarrow A_{w_2} R_3$
$R_3 \rightarrow A_{w_3} R_4$
$\vdots$
$R_{n-2} \rightarrow A_{w_{n-2}} R_{n-1}$
$R_{n-1} \rightarrow A_{w_{n-1}} A_n$
$A_0 \rightarrow 0$
$A_1 \rightarrow 1$
This CFG is of size $n + 1$.
$\blacksquare$

2 Strings w Such That L_w Has a Large CFG

Notation 2.1 Let $x, y \in \{0, 1\}^*$.

1. $C(x)$ is the Kolmogorov complexity of x. (We assume some model of computation but note that if we chose a different one it would only affect $C(x)$ by an additive constant.)

2. $C(x \mid y)$ is the conditional Kolmogorov Complexity of x given y. (Same model comments apply here.)

Theorem 2.2 *There is a function $w : \mathbb{N} \rightarrow \{0, 1\}^*$ such that, for all $n \in \mathbb{N}$, the following hold:*

1. *$|w(n)| = n$.*

2. *There is a CFG for $L_{w(n)}$ of size $n + O(1)$. (This follows from Lemma **??**.)*

3. *Any CFG that generates $L_{w(n)}$ has size $\Omega(\frac{n}{\log n})$.*

Proof: We define the function w as follows: $w(n)$ is the lexicographically least string of length n such that $C(w(n)) \geq n$. (The lex-least is not needed and is only there for definiteness.) We denote $w(n)$ by w.

Let G be a CFG that generates L_w. Let s be the size (number of rules) of G. If there are s rules, then each nonterminal can be represented with $O(\log(s))$ bits. Hence each rule can be represented with $O(\log(s))$ bits. Therefore the CFG can be represented with $O(s \log s)$ bits.

We use G to create a Turing Machine of size $O(s \log s)$ that, on input the empty string, outputs w:

Try all possible derivations to generate a string. The first time a string is generated, output it and stop.

We have.

$$n \geq \Omega(s \log s)$$

so

$$s \geq \Omega(\frac{n}{\log n})$$

.

∎

Is the string w natural? We would say no. We explore this more in the next section.

We now have two extremes:

- If $w = 0^n$ then L_w can be generated by a CFG of size $O(\log n)$.
- If w is Kolmogorov random then any CFG that generates L_w is of size $\Omega(\frac{n}{\log n})$.

Are there w such that L_w can be generated by a CFG of size intermediary between $O(\log n)$ and $O(n)$? Yes. We won't dwell on this, but the key is to take a string of the form $w0^{n-f(n)}$ where (1) f is chosen carefully depending on which intermediary function you want, and (1) w is a Kolmogorov random string of length $f(n)$.

3 Is There a Natural example that is probably hard?

The strings w from Theorem **??** seems unnatural. One way to pin this down is to note that the function w is not computable.

Is there a computable w with the same properties?

Yes. Since we can compute the set of strings of length n generated by a given CFL, we can simply search for the first w such that no CFL of size at most $\frac{n}{\log n}$ computes $\{w\}$.

However such a w may not be natural. Below we show that a de Bruijn sequence of length n must have a CFG of size at least $\Omega(\frac{n}{\log n})$.

A de Bruijn sequence of order n is a binary string w such that all n-bit strings occur exactly once as a subsequence of w, where we allow the subsequence to wrap around. There is a simple construction of de Bruijn sequences for any length that is a power of 2. An example of a de Bruijn sequence of order 4 is 0000111101100101.

Fix $n = 2^k$ and let w be a de Bruijn sequence of order k and length n. Suppose you have a CFG that generates $\{w\}$ and a variable A occurs at least twice in the derivation tree for w. All variables A must generate the same string z below them, or you could swap derivations and create a new string. Then $|z| < k$ since any sequence of length at least k can occur at most once in w be definition.

In the derivation tree consider all the variables that generate strings of length less than k but whose parents generate strings of length at least k. There are at least $\frac{n}{\log n}$ such variables and at least $\frac{n}{2\log n}$ parents, all of who much be distinct variables. The CFG has size at least $\frac{n}{2\log n}$ since every variable must occur on the left side of a production rule.

4 Acknowledgments

We would like to thank Vitanyi and Li for commentary.