Theoretical Computer Science 412 (2011) 6432–6450

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Immunity and pseudorandomness of context-free languages

Tomoyuki Yamakami *

School of Computer Science and Engineering, University of Aizu, 90 Kami-Iawase, Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima 965-8580, Japan

ARTICLE INFO

Article history:
Received 17 February 2009
Received in revised form 10 July 2011
Accepted 19 July 2011
Communicated by Z. Esik

Keywords:
Regular language
Context-free language
Immune
Simple
Primeimmune
Pseudorandom
Pseudorandom generator
Swapping lemma

ABSTRACT

We discuss the computational complexity of context-free languages, concentrating on two well-known structural properties—immunity and pseudorandomness. An infinite language is REG-immune (resp., CFL-immune) if it contains no infinite subset that is a regular (resp., context-free) language. We prove that (i) there is a context-free REG-immune language outside REG/n and (ii) there is a REG-bi-immune language that can be computed deterministically using logarithmic space. We also show that (iii) there is a CFL-simple set, where a CFL-simple language is an infinite context-free language whose complement is CFL-immune. Similar to the REG-immunity, a REG-primeimmune language has no polynomially dense subsets that are also regular. We further prove that (iv) there is a context-free language that is REG/n-bi-primeimmune. Concerning pseudorandomness of context-free languages, we show that (v) CFL contains REG/n-pseudorandom languages. Finally, we prove that (vi) against REG/n, there exists an almost 1–1 pseudorandom generator computable in nondeterministic pushdown automata equipped with a write-only output tape and (vii) against REG, there is no almost 1–1 weakly pseudorandom generator computable deterministically in linear time by a single-tape Turing machine.

1. Motivations and a quick overview

The notion of *context-free languages* [9] is one of the most fundamental concepts in formal language theory. Besides its theoretical interest, the context-freeness has drawn, since the 1960s, practical applications in key fields of computer science, including programming languages, compiler implementation, and markup languages, mainly attributed to unique traits of context-free grammars or phrase-structure grammars. Some of the traits can be highlighted by, for instance, pumping and swapping lemmas [7,31], normal form theorems [10,15], and undecidability theorems [7,13], all of which reveal certain hidden substructures of the context-free languages. The literature over half a century has successfully explored numerous basic properties (inclusive of operational closure, normal forms, and minimization) of the family CFL of all context-free languages. We wish to continue promoting our understandings of CFL further. This family CFL contains a number of non-regular languages, such as $L_{eq} = \{0^n1^n \mid n \geq 0\}$ and $Equal = \{w \in \{0,1\}^* \mid \#_0(w) = \#_1(w)\}$, where $\#_b(w)$ denotes the number of b's in w. An effective use of a pumping lemma, for example, easily separates them from the family REG of regular languages (see, e.g., [19] for their proofs). Nonetheless, these two context-free languages look quite different in nature and in complexity. How different is one language from another? How can we exactly describe a "complex" nature of those languages? These questions that arise naturally motivate us to search for a suitable "complexity measure". Since time-complexity is not a suitable complexity measure for the context-free languages, another simple way to scale their complexity is to show "structural" differences among those languages.

Up until now, numerous structural properties have been proposed for polynomial-time complexity classes, such as P (deterministic polynomial-time class) and NP (nondeterministic polynomial-time class), and have been studied to

* Tel.: +81 80 5451 1961.
E-mail address: TomoyukiYamakami@gmail.com.

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.07.013

understand their behaviors and also characteristics. Many of those properties have arisen naturally in a context of answering long-unsettled questions, including the famous P =?NP question (see, e.g., [5] for those properties). To measure the complexity of each context-free language, we intend to target two well-known structural properties—*immunity* and *pseudorandomness*—which have been studied since the 1940s in computational complexity theory and computational cryptography. These two properties are known to be closely related. In this paper, we shall spotlight them within a framework of formal language theory. This framework makes it possible to prove many properties (such as the existence of CFL-immune languages), without any unproven assumption or any relativization, by taking approaches that are quite different from standard ones in a setting of polynomial-time bounded computation.

In the first part of this paper (Section 3–4), our special attention goes to languages that have only "computationally-hard" non-trivial subsets. Those languages, known as *immune* languages and *simple* languages, naturally possess high complexity. Formally, given a fixed family $\mathcal{C}$ of languages, an infinite language is *$\mathcal{C}$-immune* if it has no infinite subset in $\mathcal{C}$, and a *$\mathcal{C}$-simple* language is an infinite language in $\mathcal{C}$ whose complement is $\mathcal{C}$-immune. Significantly, the $\mathcal{C}$-immunity satisfies a *self-exclusion property*: $\mathcal{C}$ cannot be $\mathcal{C}$-immune. Notice that the notion of simplicity has played a key role in the theory of NP-completeness (see, e.g., [5]). In addition, a language is called *$\mathcal{C}$-bi-immune* if its complement and itself are both $\mathcal{C}$-immune.

These notions of immunity and simplicity date back to the 1940s, in which they were first conceived by Post [26] for recursively enumerable languages (see, e.g., [27]). Their resource-bounded analogues were discussed later in the 1970s by Flajolet and Steyaert [12]. During the 1980s, Ko and Moore [20] intensively studied such limited immunity, whereas Homer and Maass [17] explored resource-bounded simplicity. The bi-immunity notion was introduced in mid-1980s by Balcázar and Schöning [6]. Since then, numerous variants of immunity and simplicity (for instance, strong immunity, almost immunity, balanced immunity, and hyperimmunity) have been proposed and studied extensively (see, e.g., [5,32] for references therein).

Despite the past efforts in a setting of polynomial-time bounded computation, the immunity notion has eluded from our full understandings; for instance, it has been open whether there exists a P-immune set in NP or even an NP-simple set since the existence of such a set immediately yields a class separation between NP and co-NP. Only in relativized worlds, we can prove directly the existence of those immune and simple sets (see, e.g., [4,6,17,22,28,32]). While there is a large volume of work on the immunity of polynomial-time complexity classes, there has been little study done on the immunity of the context-free languages since the work of Flajolet and Steyaert. We expect that an analysis of REG-immunity inside CFL would bring into new light a structural difference among various context-free languages. For instance, the aforementioned context-free language L_{eq} is REG-immune [12], whereas its accompanied language *Equal* is not REG-immune. Moreover, we can prove many structural properties with no extra unproven assumptions or even no relativization. For instance, unlike the case of NP-simplicity, a direct argument demonstrates that CFL-simple languages actually exist. As those examples suggest, *context-freeness* provides tremendous advantages of proving immunity as well as simplicity over polynomial-time complexity classes.

Nonetheless, all questions concerning the REG-immunity in CFL have not settled in this paper. One of those unsettled questions is related to REG-bi-immunity. It is unclear that REG-bi-immune languages actually exist inside CFL. At our best, we can prove that the language class L (deterministic logarithmic-space class) contains REG-bi-immune languages. Another unsolved question concerns a density issue of immune languages. Notice that all known REG-immune languages L in CFL have exponentially-small density rate $|L \cap \Sigma^n|/|\Sigma^n|$. The REG-immune language L_{eq}, for instance, has density rate $|L_{eq} \cap \{0,1\}^n|/2^n \leq 1/2^n$ for each even length n; in contrast, *Equal*, which is not even REG-immune, has its density rate $|Equal \cap \{0,1\}^n|/2^n \geq 1/n$ for any sufficiently large even number n. Naturally, we can ask whether there exists any context-free REG-immune language whose density $|L \cap \Sigma^n|$ is lower-bounded by a "polynomial" fraction, i.e., $1/p(n)$ for a certain non-zero polynomial p. Such a density condition is referred to as *polynomially dense* or *p-dense*. In this paper, as the first step toward the above open question, we can show the existence of a p-dense REG-immune language in L. The difficulty of proving those structural properties of CFL might indicate a limitation of the expressing power of context-freeness as languages.

Recall that $\mathcal{C}$-immunity requires the non-existence of an infinite subset in $\mathcal{C}$. Is there any language that lacks only p-dense subsets (instead of all infinite subsets) in $\mathcal{C}$? Such a natural question gives rise to a variant of $\mathcal{C}$-immunity, referred to as *$\mathcal{C}$-primeimmunity*. Now, we turn our attention to this new notion inside CFL. With a slightly adroit argument, we can prove that an "extended" language of *Equal*, $Equal_* = \{aw \mid a \in \{\lambda, 0, 1\}, w \in Equal\}$, is REG/$n$-primeimmune, where REG/$n$ is obtained from REG by supplementing appropriate "advice" of size n [29,31]. In stark contrast to the REG-bi-immunity, we can show that REG-bi-primeimmune languages (even REG/n-bi-primeimmune languages) exist inside CFL.

The second part of this paper (Section 5–6) is exclusively devoted to a property of computational randomness, or *pseudorandomness*. An early computational approach to "randomness" began in the 1940s. Church's [11] random 0–1 sequences, for instance, demand that every infinite subsequence should contain *asymptotically* the same number of 0s and 1s. This line of study on computational randomness, also known as *stochasticity*, concerns asymptotic behaviors of random sequences. It has been known a close connection between stochasticity and bi-immunity.

To suit our study of the context-free languages, however, we rather look into "non-asymptotic" behaviors of randomness inside languages. This paper discusses the following type of "random" languages. We say that a language L is *$\mathcal{C}$-pseudorandom* if, for every language A in $\mathcal{C}$, the characteristic function χ_A agrees with χ_L on "nearly" 50% of strings of each length, where "nearly" means "with a negligible margin of error". Our notion can be seen as a variant of Wilber's [30] randomness, which dictates an asymptotic behavior of χ_L and χ_A.

Similar in the case of primeimmunity, p-denseness requires our special attention. Targeting p-dense languages, we introduce another "randomness" notion, called *weak $\mathcal{C}$-pseudorandomness*, as a non-asymptotic variant of Müller's [25] balanced immunity, Loveland's [23] unbiasedness, and weak-stochasticity of Ambos-Spies et al. [2]. Loosely speaking, a language L is weak $\mathcal{C}$-pseudorandom if the density rate $|L \cap A \cap \Sigma^n|/|A \cap \Sigma^n|$ is close to 1/2 for every p-dense language A in $\mathcal{C}$.

A typical example of REG/n-pseudorandom language is the set IP_* whose strings are of the form auv with $a \in \{\lambda, 0, 1\}$ and $|u| = |v|$ such that the binary inner product between u^R and v is odd. A close connection between pseudorandomness and primeimmunity draws a conclusion that IP_* is also REG/n-bi-primeimmune. By clear contrast, the aforementioned language $Equal_*$, for instance, can separate the notion of REG/n-primeimmunity from the notion of weak REG/n-pseudorandomness.

In the early 1980s, Blum and Micali [8] studied *pseudorandom generators*, which produce unpredictable sequences. Our formulation of pseudorandom generators, attributed to Yao [33], uses indistinguishability from uniform sequences. Loosely speaking, a pseudorandom generator is a function producing a string that looks random for any target adversary (in this case, the generator is said to *fool* it). In our language setting, we call a function G mapping Σ^* to Σ^* with stretch factor $s(n)$ (that is, $|G(x)| = s(|x|)$) a *pseudorandom generator* against a language family $\mathcal{C}$ if G fools every language in $\mathcal{C}$. Our pseudorandom generator actually tries to fool languages in a sense that, over string inputs of each length n, the outcome distribution of the generator is indistinguishable from the strings of length $s(n)$; namely, the function $\ell(n) = |\mathrm{Prob}_x[\chi_A(G(x)) = 1] - \mathrm{Prob}_y[\chi_A(y) = 1]|$ has negligibly small values, where x and y are chosen uniformly at random from Σ^n and $\Sigma^{s(n)}$, respectively. We can prove that, against the language family REG/n, there exists an almost 1–1 (one-to-one) pseudorandom generator computable by a nondeterministic pushdown automaton equipped with an output tape. As a limitation of the power of generators, we can show that, even against REG, there is no almost 1–1 pseudorandom generator computable by a one-tape one-head linear-time deterministic Turing machine.

2. Foundations

The *natural numbers* are nonnegative integers and we write $\mathbb{N}$ to denote the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} - \{0\}$ for convenience. For any two integers m, n with $m \leq n$, the notation $[m, n]_{\mathbb{Z}}$ stands for the integer interval $\{m, m+1, m+2, \ldots, n\}$. The *symmetric difference* between two sets A and B, denoted $A \triangle B$, is the set $(A-B) \cup (B-A)$. In this paper, all logarithms are assumed to have base two unless otherwise stated. Let $\log^{(1)} n = \log n$ and $\log^{(i+1)} n = \log(\log^{(i)} n)$ for each number $i \in \mathbb{N}^+$. A function μ from $\mathbb{N}$ to $\mathbb{R}^{\geq 0}$ (all nonnegative reals) is called *noticeable* if there exists a positive polynomial p such that $\mu(n) \geq 1/p(n)$ for all but finitely many numbers n in $\mathbb{N}$. By contrast, μ is called *negligible* if we have $\mu(n) \leq 1/p(n)$ for any positive polynomial p and for all sufficiently large numbers $n \in \mathbb{N}$.

Our *alphabet*, often denoted Σ, is always a nonempty finite set. A *string* is a series of symbols taken from Σ, and the *length* of a string x is the number of symbols in x and is denoted $|x|$. The *empty string* is always denoted λ and, for two strings x and y, xy denotes the *concatenation* of x and y. In particular, λx coincides with x. The notation Σ^n denotes the set of all strings of length n. For any string x of length n and for any index $i \in [0, n]_{\mathbb{Z}}$, $pref_i(x)$ is the substring of x, made up with the first i symbols of x. In particular, we have $pref_0(x) = \lambda$. For each string $w \in \Sigma^*$ and any symbol $a \in \Sigma$, the number of a's appearing in w is represented by $\#_a(w)$. A *language* over an alphabet Σ is a subset of Σ^*, and the *characteristic function* χ_A of A is defined as $\chi_A(x) = 1$ if $x \in A$ and $\chi_A(x) = 0$ otherwise for every string $x \in \Sigma^*$.

For any language L over Σ, the *complement* of L (i.e., $\Sigma^* - L$) is often denoted $\overline{L}$ whenever Σ is clear from the context. Furthermore, the *complement* of a family $\mathcal{C}$ of languages is the collection of all languages whose complements are in $\mathcal{C}$. We use the conventional notation co-$\mathcal{C}$ to denote the complement of $\mathcal{C}$. For simplicity, the notation $dense(L)(n)$ expresses the cardinality of the set $L \cap \Sigma^n$; that is, $dense(L)(n) = |L \cap \Sigma^n|$. A language L over Σ is called *(polynomially) sparse* if $dense(L)(n)$ is upper-bounded by a certain fixed polynomial in n.

Since this paper mainly discusses *regular languages* and *context-free languages*, we assume the reader's basic knowledge on fundamental mechanisms of one-tape one-head one-way finite automata, possibly equipped with pushdown (or last-in first-out) stacks. See, e.g., [18,19] for the formal definitions of these finite automata. Generally speaking, for each finite automaton M, the notation $L(M)$ represents the set of all strings "accepted" by M under appropriate accepting criteria. Notice that such criteria may significantly differ if we choose different machine types. Conventionally, we say that M *recognizes* a language L if $L = L(M)$. Languages recognized by *deterministic finite automata* (or dfa's) and *nondeterministic pushdown automata* (or npda's) are respectively called *regular languages* and *context-free languages*. For ease of notation, we denote by REG the family of regular languages and by CFL the family of context-free languages. In addition, *deterministic pushdown automata* (or dpda's) recognize only *deterministic context-free languages*, and DCFL denotes the family of all deterministic context-free languages.

It is known that the language family CFL is not closed under conjunction (see, e.g., [19] for the proof). This fact inspires us to introduce a restricted conjunctive closure of CFL. For any positive integer k, the *k conjunctive closure* of CFL, denoted CFL(k), is the collection of all languages L such that there are k languages $L_1, L_2, \ldots, L_k$ in CFL for which $L = L_1 \cap L_2 \cap \cdots \cap L_k$. By its definition, CFL(1) coincides with CFL itself.

To explain the notion of *advice*, we first adapt a "track" notation $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ from [29]. For any pair of symbols $\sigma \in \Sigma_1$ and $\tau \in \Sigma_2$, the notation $\left[\begin{smallmatrix} \sigma \\ \tau \end{smallmatrix}\right]$ denotes a new symbol made from σ and τ. For two strings $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_n$ of the same length n, the notation $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ is shorthand for the string $\left[\begin{smallmatrix} x_1 \\ y_1 \end{smallmatrix}\right]\left[\begin{smallmatrix} x_2 \\ y_2 \end{smallmatrix}\right] \cdots \left[\begin{smallmatrix} x_n \\ y_n \end{smallmatrix}\right]$. An *advice function* is a map from $\mathbb{N}$

to Γ^*, where Γ is an appropriate alphabet. For any family $\mathcal{C}$ of languages, the advised class $\mathcal{C}/n$ denotes the collection of languages L over an alphabet Σ for which there exist another alphabet Γ, an advice function $h : \mathbb{N} \to \Gamma^*$, and a language $A \in \mathcal{C}$ such that, for every string $x \in \Sigma^*$, (i) $|h(|x|)| = |x|$ (i.e., length preserving) and (ii) $x \in L$ iff $\left[\begin{smallmatrix} x \\ h(|x|) \end{smallmatrix}\right] \in A$ [29,31].

As an additional computation model, we introduce the notion of one-tape one-head off-line Turing machines whose tape heads move in all directions. Such machines are succinctly called *1TMs*. All tape cells of an infinite input/work tape are indexed with integers and an input string of length n is given in the cells indexed between 1 and n surrounded by two designated *endmarkers*. We take a notation 1-DTIME($t(n)$) from [29] to denote the collection of all languages that are recognized deterministically within time $t(n)$ by those 1TMs. As a special case, we write 1-DLIN for 1-DTIME($O(n)$). It is well-known that REG = 1-DLIN = 1-DTIME($o(n \log n)$) [16,21].

To handle (multi-valued partial) functions, we further consider Turing machines that produce (possibly) many output strings at once. Conventionally, whenever a single-tape machine halts along the tape that contains only a block of non-blank symbols beginning at the left endmarker and surrounded only by blanks, we treat the string given in this block as an *outcome* of the machine. A (partial) function f from Σ^* to Γ^*, where Σ and Γ are two alphabets, is called *length preserving* if $|f(x)| = |x|$ for any string x in the domain of f.

Let us introduce several function classes, which are natural extensions of the language families REG and CFL. The function class 1-FLIN is the set of all *single-valued total* functions computable in time $O(n)$ by deterministic 1TMs. Similarly, the notation 1-FLIN(partial) expresses the set of all *single-valued partial* functions f such that there exists a linear-time deterministic 1TM M that starts with input x and halts with output $f(x)$ by entering an accepting state whenever $f(x)$ is defined; M always enters a rejecting state when $f(x)$ is not defined.

We expand single-valued functions to multi-valued functions, which produce sets of values. We define 1-NLINMV as the class of all *multi-valued partial* functions f for which there exists a nondeterministic 1TM M, provided that all computation (both accepting and rejecting) paths terminate with certain output values in time $O(n)$, together with the condition that $f(x)$ consists of all output values produced along accepting paths. Notice that, when $f(x) = \emptyset$, there should be no accepting path. See [29] for their basic properties.

The original npda model was introduced to recognize "languages". Let us expand this model to compute (partial) functions. For this purpose, we equip an npda with an additional *output tape* and its associated tape head. Now, our npda has two tapes: a *read-only* input tape and a *write-only* output tape. This new npda acts as a standard npda with a single stack except for moves of an output-tape head. In the write-only output tape, its tape head always moves to the right whenever it writes a non-blank symbol in its tape cell. Here, we allow the tape head to stay still on a blank symbol as long as it does not write any non-blank symbol. Since the head moves only to a new blank cell, it cannot read any meaningful symbol that have already written in the output tape. Along each computation path, we define an *output* of the npda as follows. When the npda enters an accepting state, we treat the string produced on the output tape as an output of the machine. On the contrary, when the machine enters a rejecting state, we assume that the machine produces no output along this path although there may be non-blank symbols left on the output tape. Hence, the machine can produce at least one output value or no output value at all. Therefore, such an npda in general computes a multi-valued partial function. Let CFLMV denote the collection of all multi-valued partial functions that can be produced by those npda's. Moreover, CFLSV consists of all *single-valued partial* functions in CFLMV. When the functions f are limited to be total (i.e., $f(x)$ is always defined), we use the notation $\mathrm{CFLSV_t}$. Note that, for every language L, $L \in \mathrm{CFL}$ iff $\chi_L \in \mathrm{CFLSV_t}$.

3. Resource-bounded immunity and simplicity

Intuitively, an *immune* language contains finite subsets and only infinite subsets that are "hard" to compute; in other words, it lacks any non-trivial "easy" subset. In contrast, a *simple* language inherits the immunity only for its complement. Such languages turn out to possess quite high complexity. The original notions of immunity and simplicity are rooted in the 1940s and later adapted to computational complexity theory in the 1970s with various restrictions on their computational resources.

The notion of resource-bounded immunity for an arbitrary family $\mathcal{C}$ of languages can be introduced in the following abstract way. A language L is said to be $\mathcal{C}$-*immune* if (i) L is infinite and (ii) no infinite subset of L exists in $\mathcal{C}$. When a language family $\mathcal{D}$ contains a $\mathcal{C}$-immune language, we conveniently say that $\mathcal{D}$ is $\mathcal{C}$-*immune*. Since $\mathcal{C}$ cannot be $\mathcal{C}$-immune, if $\mathcal{D}$ is $\mathcal{C}$-immune then it immediately follows that $\mathcal{D} \not\subseteq \mathcal{C}$. On the contrary, the separation $\mathcal{D} \not\subseteq \mathcal{C}$ cannot, in general, guarantee the existence of $\mathcal{C}$-immune languages inside $\mathcal{D}$. By this reason, a separation between two language families by immune languages is sometimes referred to as a *strong separation*. In a polynomial-time setting, for instance, even if assuming that P $\neq$ NP, it is not known whether there is a P-immune language in NP or equivalently NP is P-immune.

3.1. Existence of immune and simple languages

Within a framework of formal language theory, we shall discuss the immunity of two well-known families of languages: REG and CFL. Earlier, Flajolet and Steyaert [12] presented two examples: a REG-immune language $L_{eq} = \{0^n 1^n \mid n \in \mathbb{N}\}$ and a CFL-immune language $L_{3eq} = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. Notice that, in contrast, similar non-regular languages $Equal = \{x \in \{0, 1\}^* \mid \#_0(x) = \#_1(x)\}$ and $3Equal = \{x \in \{0, 1, 2\}^* \mid \#_0(x) = \#_1(x) = \#_2(x)\}$ are not REG-immune, because two regular

languages $\{(01)^n \mid n \in \mathbb{N}\}$ and $\{(012)^n \mid n \in \mathbb{N}\}$ are respectively infinite subsets of *Equal* and of 3*Equal*. This clear contrast signifies a "structural" difference among those languages. We shall see more examples of immune languages.

Since REG $\subseteq$ CFL, the CFL-immunity clearly implies the REG-immunity but the converse does not hold because, for instance, L_{eq} is REG-immune and also belongs to CFL. Since L_{eq} and L_{3eq} are *sparse* languages (because, e.g., $dense(L_{eq})(n) \leq 1$ for all lengths $n \in \mathbb{N}$), they belong to the advised class REG/n. Therefore, since $L_{eq} \in$ DCFL and $L_{3eq} \in$ CFL(2), the language family DCFL $\cap$ REG/n is REG-immune, and CFL(2) $\cap$ REG/n (thus CFL(2) $\cap$ CFL/n) is CFL-immune. In addition to these results, we remark that the language family DCFL $-$ REG/n is also REG-immune. A simple example is the "marked" language $Pal_{\#} = \{w\#w^R \mid w \in \{0,1\}^*\}$ over the ternary alphabet $\{0, 1, \#\}$, where # is used only as a *separator*. Notice that a use of this separator is crucial because a corresponding unmarked version $Pal = \{ww^R \mid w \in \{0,1\}^*\}$ (even-length palindromes) is no longer REG-immune. The REG-immunity of DCFL $-$ REG/n can be obtained simply by applying a standard pumping lemma for regular languages [7] (for the immunity of $Pal_{\#}$) and a swapping lemma for regular languages [31] (for the non-membership of $Pal_{\#}$ to REG/n). When turning to the CFL-immunity, on the contrary, it is not known whether CFL(2) $-$ CFL/n is CFL-immune. The bast we can show at present is that L$-$CFL/n is CFL-immune, where L consists of all languages recognized by deterministic Turing machines with a single read-only input tape and a logarithmic-space bounded work tape. A typical example is the marked language $3Dup_{\#} = \{w\#w\#w \mid w \in \{0,1\}^*\}$. A standard pumping lemma for context-free languages [7] proves the CFL-immunity of $3Dup_{\#}$; moreover, a direct use of a swapping lemma for context-free languages [31] proves that $3Dup_{\#} \notin$ CFL/n. Since $3Dup_{\#} \in$ L, the CFL-immunity of L $-$ CFL/n follows immediately.

The immunity notion has given rise to the notion of *simplicity*. In general, a language L is called $\mathcal{C}$-*simple* if (i) L is infinite, (ii) L is in $\mathcal{C}$, and (iii) $\overline{L}$ is $\mathcal{C}$-immune. The existence of such a $\mathcal{C}$-simple language clearly leads to a class separation $\mathcal{C} \neq$ co-$\mathcal{C}$. Because of this implication, we do not know whether NP-simple languages exist (since, otherwise, NP $\neq$ co-NP follows). It is therefore natural to ask if CFL-simple languages actually exist. In what follows, we prove the existence of such CFL-simple languages.

Proposition 3.1. *There exist* CFL*-simple languages. Moreover, the complements of some of those languages belong to* CFL(2) $\cap$ REG/n.

Our example of CFL-simplicity is the complement of a language L_{keq} ($k \geq 3$), which is a natural generalization of L_{3eq}. Let $k \geq 3$ be fixed. We define $L_{keq} = \{\sigma_1^n\sigma_2^n \cdots \sigma_k^n \mid n \in \mathbb{N}\}$ over the k-letter alphabet $\Sigma_k = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. We shall show that the complement of L_{keq} is indeed CFL-simple. This gives a clear contrast with the fact that both the language 3*Equal* (associated with L_{3eq}) and its complement are not even REG-immune.

Proof of Proposition 3.1. Let k be any integer at least 3. We intend to show that (1) $\overline{L_{keq}}$ is in CFL, (2) L_{keq} is in CFL(2) $\cap$ REG/n, and (3) L_{keq} is CFL-immune.

(1) Our first claim is that $\overline{L_{keq}}$ belongs to CFL. To simplify our proof, we shall argue only on the case $k = 3$. Let us introduce two additional languages $L_3 = \{\sigma_1^k\sigma_2^l\sigma_3^m \mid k, l, m \in \mathbb{N}\}$ and $L_{3neq} = \{\sigma_1^k\sigma_2^l\sigma_3^m \mid k \neq l, l \neq m, \text{ or } k \neq m\}$. Note that L_{3neq} equals the union of the following three sets: $\{\sigma_1^k\sigma_2^l\sigma_3^m \mid k \neq l, m \geq 0\}$, $\{\sigma_1^k\sigma_2^l\sigma_3^m \mid l \neq m, k \geq 0\}$, and $\{\sigma_1^k\sigma_2^l\sigma_3^m \mid m \neq k, l \geq 0\}$, all of which are apparently context-free. Since CFL is closed under union, L_{3neq} should belong to CFL. Moreover, since $\overline{L_{3eq}} = L_{3neq} \cup \overline{L_3}$ and $\overline{L_3} \in$ REG $\subseteq$ CFL, the language $\overline{L_{3eq}}$ is also in CFL.

(2) To show that $L_{keq} \in$ REG/n, choose an advice function h defined as $h(n) = \sigma_1^{n/k}\sigma_2^{n/k} \cdots \sigma_k^{n/k}$ for all numbers $n \equiv 0 \pmod{k}$ and $h(n) = 0^n$ for all the other n's. If we define $S = \{\left[\begin{smallmatrix} w \\ w \end{smallmatrix}\right] \mid w \in \Sigma_k^*\}$, then $\left[\begin{smallmatrix} w \\ h(|w|) \end{smallmatrix}\right]$ is in S exactly when $w = h(|w|)$, which means that $w \in L_{keq}$. Thus, L_{keq} belongs to REG/n. To show that $L_{keq} \in$ CFL(2), let us deal only with the case where $k = 2m$ and $m = 2j+1$ for a certain number $j \in \mathbb{N}^+$, since the other cases are similar. We introduce two useful languages L_1 and L_2 defined as follows: L_1 (resp., L_2) consists of all strings of the form $\sigma_1^{n_1}\sigma_2^{n_2} \cdots \sigma_k^{n_k}$ such that $n_i = n_{k+1-i}$ for all indices $i \in [1, m]_{\mathbb{Z}}$ (resp., $n_{2i+1} = n_{2i+2}$ and $n_{2i+m+1} = n_{2i+m+2}$ for all $i \in [0, j-1]_{\mathbb{Z}}$). Clearly, L_1 and L_2 are both context-free. Since the target language L_{keq} can be expressed as $L_1 \cap L_2$, L_{keq} belongs to CFL(2).

(3) Finally, we shall check the CFL-immunity of L_{keq}. Assume that there exists an infinite subset $A \in$ CFL of L_{keq}. To this A, we then apply a standard pumping lemma for context-free languages.[1] Let m be a pumping-lemma constant. Choose $w = \sigma_1^n\sigma_2^n \cdots \sigma_k^n$ in A with $n \geq m$. Take a decomposition $w = uvxyz$ with $|vxy| \leq m$ and $|vy| \geq 1$ such that uv^ixy^iz is in A for every index $i \in \mathbb{N}$. Since $|vxy| \leq m \leq n$, there exists an index i such that vxy is a substring of either σ_i^n or $\sigma_i^n\sigma_{i+1}^n$. Thus, we need to examine only two cases: (i) v and y are both substrings of σ_i^n or (ii) v is a substring of σ_i^n and y is a substring of σ_{i+1}^n. In either case, the string uv^2xy^2z cannot belong to A. This is absurd, and therefore A does not exist. We thus reach the desired conclusion of the CFL-immunity of L_{keq}. □

Notice that our CFL-simple languages $\overline{L_{keq}}$ are not even REG-immune because, for instance, the language $\overline{L_3}$ is an infinite regular subset of $\overline{L_{3eq}}$. This immediately raises a natural question of whether there exist REG-immune CFL-simple languages.

[1] [Pumping Lemma for Context-Free Languages] Let L be any infinite context-free language. There exists a positive number m such that, for any $w \in L$ with $|w| \geq m$, w can be decomposed as $w = uvxyz$ with the following three conditions: (i) $|vxy| \leq m$, (ii) $|vy| \geq 1$, and uv^ixy^iz is in L for any $i \in \mathbb{N}$. See [7,19].

3.2. Properties of immune languages

Immune languages lack infinite subsets of certain low complexity, and therefore, as we have presented in the previous subsection, they are of quite high complexity. To improve our understandings of the REG-immunity, we wish to examine this notion by studying its relationships to three existing notions—nonregularity, quasireduction, and hardcore. The first notion relates to a nonregularity measure, which leads to another characterization of the REG-immunity. The *nonregularity* $N_L(n)$ of a language L at n is the total number of equivalence classes in $\Sigma^n/\equiv_L$, where the relation $\equiv_L$ is defined as: $x \equiv_L y$ iff $\forall z \in \Sigma^*[xz \in L \iff yz \in L]$.

Proposition 3.2. *A language L is* REG*-immune iff L is infinite and, for every infinite subset A of L and for every constant $c > 0$, $N_A(n) > c$ holds for an infinite number of indices $n \in \mathbb{N}$.*

This proposition is a natural extension of the so-called *Myhill-Nerode Theorem* [18], which bridges between the nonregularity and REG. We include its proof for completeness.

Proof of Proposition 3.2. (If – part) We prove a contrapositive. Assume that L has an infinite subset A in REG. Since $A \in$ REG, by the Myhill-Nerode Theorem, the cardinality of the set $\Sigma^*/\equiv_A$ is finite. In other words, $N_A(n)$ is upper-bounded by a certain constant, which is not depending on n.

(Only If – part) Let L be REG-immune. Assume that there are an infinite subset A of L and a constant $c > 0$ for which $N_A(n) \leq c$ for all but finitely many $n \in \mathbb{N}$. Let $\{A_1, A_2, \ldots, A_c\}$ denote all equivalence classes in $\Sigma^*/\equiv_A$. Take the lexicographically minimal string, say, a_i from each set A_i. Consider a dfa M with its transition function δ defined by: $\delta(i, \sigma) = j$ iff $a_i\sigma \equiv_A a_j$. The initial state i_0 satisfies $\lambda \in A_{i_0}$. The set of final states is $F = \{i \mid a_i \in A\}$. It is not difficult to check that M indeed recognizes A. This implies that A is regular, a contradiction against the REG-immunity of L. □

Our notion of 1-DLIN-m-quasireduction gives the second characterization to the REG-immunity. Let us recall from Section 2 the partial function class 1-FLIN(partial). A 1-DLIN-*m-quasireduction* from L to A is a single-valued partial function f that satisfies the following two conditions: (i) for every string x, when $f(x)$ is defined, $x \in L$ iff $f(x) \in A$ and (ii) f is in 1-FLIN(partial).

Lemma 3.3. *The language L is* REG*-immune iff L is infinite and, for any set A and for any* 1-DLIN*-m-quasireduction $f : L \to A$ and for any $u \in A$, $f^{-1}(u)$ is finite.*

Proof. (If – part) Assume that an infinite language L is not REG-immune. Take an infinite regular subset $A \subseteq L$. Choose an element $u_0 \in A$ and, for every string x, define $f(x) = u_0$ if $x \in A$ and undefined otherwise. Since $f^{-1}(u_0)$ coincides with A, $f^{-1}(u_0)$ is infinite. Moreover, f belongs to 1-FLIN(partial) since $A \in$ REG. Thus, f is a 1-DLIN-m-quasireduction from L to A.

(Only If – part) Assume that we have an infinite set L, another set A, a 1-DLIN-m-quasireduction $f : L \to A$, and an element $u_0 \in A$ such that $B =_{def} f^{-1}(u_0)$ is infinite. Since $f \in$ 1-FLIN(partial), take a linear-time deterministic 1TM M that computes f. Note that, for every input x, $x \in B$ iff $M(x)$ halts in an accepting state and outputs u_0. Hence, B is in REG. Therefore, L has an infinite regular subset. □

Next, we give the third characterization of the REG-immunity using a notion of "hardcore"; however, our definition of "hardcore" differs from a time-restricted definition of *(polynomial) hardcore* for polynomial-time bounded computation (see, e.g., [5] for its definition). With a use of an npda, we rather impose a space restriction on the size of a stack used by the npda. To be more accurate, for any npda $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$, any constant $k \in \mathbb{N}$, and any input string $x \in \Sigma^*$, we introduce the notation $M(x)_k$ defined as follows: (1) $M(x)_k = 1$ if there is an accepting path of M on the input x with stack size at most k; (2) $M(x)_k = 0$ if all computation paths of M on x are rejecting paths with stack size at most k; and (3) $M(x)_k$ is *undefined* otherwise. A context-free language A is called a REG-*hardcore* for a language L if, for any constant $k \in \mathbb{N}$ and any npda M recognizing A, there exists a finite set $B \subseteq L$ such that $M(x)_k$ is undefined for all strings $x \in L - B$.

Proposition 3.4. *The following two statements are equivalent. Let L be any infinite context-free language.*

1. *The language L is* REG*-immune.*
2. *The language L is a* REG*-hardcore for L.*

Proof. ($1 \Rightarrow 2$) We shall prove a contrapositive. Let L be any infinite context-free language. Assuming that L is not a REG-hardcore for L, we plan to prove that L has an infinite regular subset. There exist a constant $k \in \mathbb{N}$ and an npda M with $L(M) = L$ such that, for every finite set $B \subseteq L$, $M(x)_k$ is defined (i.e., $M(x)_k \in \{0, 1\}$) for a certain input $x \in L - B$. Now, let us introduce a new npda N as follows: on input x, N simulates M on x nondeterministically and, along each computation path, whenever its stack size exceeds k, it immediately rejects x. Consider the set $L(N)$ of all strings accepted by N. By the definition of N, it follows that $L(N) \subseteq L$.

First, we claim that $L(N)$ is regular. Since k is a fixed constant, we can express the entire content of the stack as a certain new internal state. Tracking down this state, we can simulate N using a certain nondeterministic finite automaton (or nfa). This implies that $L(N)$ is regular. Next, we claim that $L(N)$ is infinite. For every finite subset B of L, a certain string $x \in L - B$ satisfies $M(x)_k \in \{0, 1\}$; hence, $x \in L(N)$. From this property, we can conclude that $L(N)$ is infinite. Therefore, $L(N)$ is an infinite regular subset of L.

($2 \Rightarrow 1$) We first assume that an infinite context-free language L is not REG-immune. This means that there exists a dfa M for which $L(M) \subseteq L$ and $L(M)$ is infinite. Since L is context-free, take an npda N that recognizes L. Now, let us define a new npda M' as follows: on input x, M' splits its computation into two nondeterministic computation paths and then simulates M and N along these paths separately. Clearly, $L(M') = L(M) \cup L(N) = L$. Choose $k = 1$ and consider $M'(x)_k$. For every string $x \in L(M)$, $M'(x)_k = 1$ follows since M is a dfa and uses no stack space. Let B be any finite subset of L. Because $L(M) - B$ is infinite within L, there exists a string x in $L - B$ for which $M'(x)_k = 1$. This implies that L cannot be a REG-hardcore for L. □

3.3. Complexity of bi-immune languages

The existence of natural REG-immune languages within CFL encourages us to search for much "stronger" immune languages in CFL. One such candidate is another variant of $\mathcal{C}$-immunity, known as $\mathcal{C}$-bi-immunity [6], where a language L is *$\mathcal{C}$-bi-immune* if L and its complement $\overline{L}$ are both $\mathcal{C}$-immune. For brevity, a language family $\mathcal{D}$ is said to be *$\mathcal{C}$-bi-immune* if there is a $\mathcal{C}$-bi-immune language in $\mathcal{D}$. In the literature, time-bounded bi-immunity has been known to be related to the notion of *genericity*, which corresponds to certain finite-extension diagonalization arguments (see, e.g., [1,32] for its connection).

Is there any REG-bi-immune language in CFL? All the examples of context-free REG-immune languages shown in Section 3.1 appear to lack the REG-bi-immunity property. Related to the open question on the existence of REG-immune CFL-simple languages, discussed in Section 3.1, if CFL is not REG-bi-immune, then no CFL-simple language can be REG-immune. Unfortunately, we are unable to answer the question at this point; instead, we shall prove that the language family $\mathrm{L} \cap \mathrm{REG}/n$ is REG-bi-immune.

Proposition 3.5. *The language family* $\mathrm{L} \cap \mathrm{REG}/n$ *is* REG*-bi-immune.*

How can we prove this proposition? Balcázar and Schöning [6] employed a diagonalization technique to construct a P-bi-immune language inside EXP (deterministic exponential-time class). Notice that any P-bi-immune language constructed by such a diagonalization depends on how to enumerate all languages in P. In our proof below, without requiring any enumeration of languages in REG, we explicitly present two REG-bi-immune languages. Our desired REG-bi-immune languages are L_{even} and L_{odd} given as follows:

- $L_{even} = \{w \in \{0,1\}^* \mid \exists k \in \mathbb{N}\,[2k < \log^{(2)} |w| \leq 2k+1]\} \cup \{\lambda\} \cup \{0,1\}^2$, and
- $L_{odd} = \{w \in \{0,1\}^* \mid \exists k \in \mathbb{N}\,[2k+1 < \log^{(2)} |w| \leq 2k+2]\} \cup \{0,1\}$.

Notice that these two languages form a partition of $\{0,1\}^*$; namely, $L_{even} \cup L_{odd} = \{0,1\}^*$ and $L_{even} \cap L_{odd} = \emptyset$.

Proof of Proposition 3.5. It suffices to show that L_{even} and L_{odd} are both REG-immune because each of them is the complement of the other. For brevity, let $\Sigma = \{0,1\}$. We begin with proving the REG-immunity of L_{even} by contradiction. Assume that there exists an infinite regular subset A of L_{even}. We apply to A a standard pumping lemma for regular languages.[2] Take a pumping-lemma constant $m > 0$ and then choose a string w in $A \cap \Sigma^n$ for a certain length n with $n \geq m+1$. Such n satisfies that $2k < \log^{(2)} n \leq 2k+1$ for a certain number $k \in \mathbb{N}$. The pumping lemma provides a decomposition $w = xyz$ with $|xy| \leq m$ and $|y| \geq 1$ for which $w_i =_{def} xy^i z$ belongs to A for any number $i \in \mathbb{N}$. Now, let $\ell = |y|$. Toward a contradiction, there are two cases to consider separately.

Case 1: Consider the case where $\log^{(2)} n = 2k+1$. In this case, we choose $i = n+1$. Since $1 \leq \ell \leq m$ and $m+1 \leq n$, the length $|w_i|$ is sandwiched by two terms as

$$2^{2^{2k+1}} = n < |w_i| = n + (i-1)\ell \leq n + n\ell \leq n(m+1) \leq n^2 = 2^{2^{2k+2}}.$$

In short, it holds that $2k+1 < \log^{(2)} |w_i| \leq 2k+2$, implying that w_i is in L_{odd}. Since $A \cap L_{odd} = \emptyset$, it immediately follows that $w_i \notin A$, a contradiction.

Case 2: Consider the case where $2k < \log^{(2)} n < 2k+1$. This means that $2^{2^{2k}} < n \leq 2^{2^{2k+1}} - 1$. When we choose $i = \lceil n(n-1)/\ell \rceil + 1$, the length $|w_i|$ can be lower-bounded by

$$|w_i| = n + (i-1)\ell \geq n + \frac{n(n-1)}{\ell} \cdot \ell = n + n(n-1) = n^2 > 2^{2^{2k+1}}.$$

In contrast, since $n \geq m+1 > m/2$, we can upper-bound $|w_i|$ as

$$|w_i| < n + \left(\frac{n(n-1)}{\ell} + 1\right) \cdot \ell = n^2 + \ell \leq n^2 + m < (n+1)^2 \leq 2^{2^{2k+2}}.$$

These two bounds together imply that $2k+1 < \log^{(2)} |w_i| < 2k+2$, concluding that $w_i \in L_{odd}$, a contradiction against the fact that $w_i \in A \subseteq L_{even}$.

From the above two cases, we can conclude that A does not exist; in other words, L_{even} is REG-immune, as requested. Similarly, we can show that L_{odd} is REG-immune.

[2] [Pumping Lemma for Regular Languages] Let L be any infinite regular language. There exists a number $m > 0$ (referred to as a pumping-lemma constant) such that, for any string w of length $\geq m$ in L, there is a decomposition $w = xyz$ for which (i) $|xy| \leq m$, (ii) $|y| \geq 1$, and (iii) $xy^i z \in L$ for any $i \in \mathbb{N}$. See [7,19].

We still need to argue that L_{even} and L_{odd} are both in $\mathrm{L} \cap \mathrm{REG}/n$. Since $\mathrm{L} \cap \mathrm{REG}/n$ is closed under complementation, it suffices to show that L_{even} belongs to $\mathrm{L} \cap \mathrm{REG}/n$. First, we shall demonstrate that $L_{even} \in \mathrm{REG}/n$. Let us consider the following advice function $h(n) = 10^{n-1}$ if $L_{even} \cap \Sigma^n \neq \emptyset$, and $h(n) = 0^n$ if $L_{odd} \cap \Sigma^n \neq \emptyset$ for any length $n \geq 1$; in addition, set $h(0) = \lambda$. Define a set A as $A = \left\{ \left[\begin{smallmatrix} x \\ 1y \end{smallmatrix} \right] \mid |x| = |y| + 1, y \in \{0,1\}^* \right\}$. It is obvious that, for every non-empty x, $x \in L_{even}$ iff $\left[\begin{smallmatrix} x \\ h(|x|) \end{smallmatrix} \right] \in A$. Since A is regular, L_{even} therefore belongs to REG/n. To show that $L_{even} \in \mathrm{L}$, let us consider the following algorithm for L_{even}.

> On input w, if $w = \lambda$ then accept it. Assume that $|w| \geq 1$. With access to w written on a read-only input tape, compute $\lceil \log^{(2)} |w| \rceil$ on its work tape. If $\lceil \log^{(2)} |w| \rceil$ is odd, then accept the input; otherwise, reject it.

It is not difficult to show that this algorithm recognizes L_{even} using only logarithmic space. This completes our proof of the proposition. □

4. P-denseness and primeimmunity

We begin with a brief discussion on a density issue of REG-immune languages. Recall that non-immunity of a language guarantees the existence of a certain infinite subset that is "computationally easy." In many cases, these infinite subsets are of *low density*. In typical examples, there are infinite *sparse* subsets $\{(01)^n \mid n \in \mathbb{N}\}$ and $\{(012)^n \mid n \in \mathbb{N}\}$ inside *Equal* and 3*Equal*, respectively. Notice that all context-free REG-immune languages L described in Section 3 satisfy the following density property: its density rate $dense(L)(n)/|\Sigma^n|$ is "exponentially small" in terms of a length parameter n. The language $Pal_{\#}$, for example, satisfies that $dense(Pal_{\#})(n)/|\Sigma^n| \leq 2^{\lfloor n/2 \rfloor}/3^n$ (thus $dense(Pal_{\#})(n) \leq |\Sigma^n|/(2.2)^n$) for every odd length $n \geq 1$. Naturally, we can question whether there exists a context-free REG-immune language whose density rate is "polynomially large." To be more precise, we call a language L over an alphabet Σ *polynomially dense* (or *p-dense*, in short) exactly when there exist a number $n_0 \in \mathbb{N}$ and a non-zero polynomial p such that $dense(A)(n) \geq |\Sigma^n|/p(n)$ for all numbers $n \geq n_0$. Our previous question is now rephrased as: is there any p-dense REG-immune language in CFL, or is CFL p-dense REG-immune? It appears that we are unable to settle this question at present. This situation seems to signify the meaningfulness of the notion of p-denseness in our study of immunity. Meanwhile, we shall show that $\mathrm{L} \cap \mathrm{CFL}/n$ is indeed p-dense REG-immune.

Proposition 4.1. *The language family* $\mathrm{L} \cap \mathrm{CFL}/n$ *is p-dense* REG*-immune.*

Let us consider the language $LCenter = \{au0^m10^mv \mid a \in \{\lambda, 0, 1\}, 2^m \leq |u| = |v| < 2^{m+1}\}$ over the alphabet $\{0, 1\}$. Notice that *LCenter* is in $\mathrm{L} \cap \mathrm{CFL}/n$. We claim in the following proof that this language is REG-immune and also p-dense.

Proof of Proposition 4.1. We want to show that *LCenter* is p-dense REG-immune. We first show that *LCenter* is p-dense. Let $w = au0^m10^mv$ in *LCenter* with $2^m \leq |u| = |v| < 2^{m+1}$. Let $n = |w|$. Consider the case where $a = \lambda$. In this case, since $2^m \leq |u| = (n - 2m - 1)/2 < 2^{m+1}$, we obtain $2^{m+1} + 2m + 1 \leq n$, which implies $n^2 \geq 2^{2m+1}$. Since $dense(LCenter)(n) = 2^{n-2m-1}$, the density rate $\frac{dense(LCenter)(n)}{|\Sigma^n|}$ equals $\frac{1}{2^{2m+1}}$, which is clearly at least $1/n^2$. The other cases where $a \in \{0, 1\}$ are similar. Therefore, *LCenter* is p-dense.

Next, we show that *LCenter* is REG-immune. Assuming otherwise, we choose an infinite subset A of *LCenter* in REG. As in the proof of Proposition 3.5, we use the pumping lemma for regular languages. Take a pumping-lemma constant $m > 0$. Let $w = au0^k10^kv$ be any string in A with $k > m$ and $2^k \leq |u| = |v| < 2^{k+1}$. Now, assume that $a = \lambda$. The other cases are similar. Let us take any decomposition $w = xyz$ with $|xy| \leq m$ and $|y| \geq 1$ such that xy^iz is in A for any number $i \in \mathbb{N}$. Since $|xy| \leq m < k$, y is a substring of u. Consider the string xz. Clearly, the center symbol of xz should be 0. Thus, xz cannot belong to *LCenter*. This is a contradiction against the fact that $xz \in A$. Therefore, *LCenter* must be REG-immune. □

Apart from the REG-immunity, we turn our attention to p-dense languages that lack only p-dense regular subsets. Such languages are referred to as REG-primeimmune. More generally, for a language family $\mathcal{C}$, we say that a language L over Σ is $\mathcal{C}$*-primeimmune* if (1) L is p-dense and (2) L has no p-dense subset in $\mathcal{C}$. A language family $\mathcal{D}$ is $\mathcal{C}$*-primeimmune* if there exists a $\mathcal{C}$-primeimmune language in $\mathcal{D}$. This definition immediately yields, similar to the $\mathcal{C}$-immunity, the self-exclusion property: $\mathcal{C}$ cannot be $\mathcal{C}$-primeimmune.

The following obvious relationship holds between p-dense REG-immunity and REG-primeimmunity. If a language L is p-dense but not REG-primeimmune, then L contains a p-dense regular subset, say, A. By the definition of p-denseness, A should be infinite and thus L must not be REG-immune. The next lemma therefore follows.

Lemma 4.2. *Let L be any language over an alphabet Σ with $|\Sigma| \geq 2$. If L is p-dense* REG*-immune, then L is* REG*-primeimmune.*

Although CFL is not known to be p-dense REG-immune, it is possible for us to show that CFL is REG-primeimmune. First, recall the context-free language *Equal* over the binary alphabet $\{0, 1\}$. Since *Equal* is technically not p-dense, we need to extend it slightly and define its "extended" language $Equal_*$ as $\{aw \mid a \in \{\lambda, 0, 1\}, w \in Equal\}$. Despite $Equal_*$'s non-REG-immunity, we can prove that $Equal_*$ is REG-primeimmune. In the next proposition, we shall challenge a slightly stronger statement: $Equal_*$ is REG/n-primeimmune. This highlights a stark difference between the REG/n-primeimmunity and the REG/n-immunity, since there exists no REG/n-immune language (because every infinite language L over an alphabet Σ has an infinite subset of the form $\{\sigma x \in L \mid \sigma \in \Sigma, x \in \Sigma^*, h(|\sigma x|) = \tilde{\sigma} x\}$ in REG/n, where $\tilde{\sigma} = \left[\begin{smallmatrix} \sigma \\ 1 \end{smallmatrix} \right]$ and h is an advice function defined as $h(n) = \tilde{\sigma} x$ if σx is the lexicographically minimal string in $L \cap \Sigma^n$ and $h(n) = 0^n$ otherwise).

Proposition 4.3. *The language $Equal_*$ is* REG*$/n$-primeimmune.*

Proof. We start our proof with an easy claim on the p-denseness of $Equal_*$. For any sufficiently large even number n, by Stirling's approximation formula, the density of $Equal_*$ can be estimated as

$$dense(Equal_*)(n) = \binom{n}{n/2} = \frac{2^n\sqrt{2}}{\sqrt{\pi n}}\left(1+\Theta\left(\frac{1}{n}\right)\right) > \frac{2^n}{n}. \tag{1}$$

When n is odd, on the contrary, since $dense(Equal_*)(n)$ equals $2\cdot dense(Equal_*)(n-1)$, it is lower-bounded by $\frac{2\cdot 2^{n-1}}{n-1} > 2^n/n$ with a help of Eq. (1). These two lower bounds yield the desired p-denseness of $Equal_*$.

Our next goal is to prove the non-existence of p-dense subset of $Equal_*$ in REG/n. Assume otherwise; namely, there is a p-dense set $A \subseteq Equal_*$ in REG/n. Since A is p-dense, a certain constant $d \geq 1$ satisfies $dense(A)(n) \geq 2^n/n^d$ for all but finitely many numbers n. Here, we shall apply a swapping lemma for regular languages.[3] Let m be a swapping-lemma constant for A and choose a sufficiently large even number n in $\mathbb{N}$. It suffices to consider only the case where m is odd. Without loss of generality, we further assume that $m \geq 5$. For each pair $i, k \in [0,n]_{\mathbb{Z}}$, the notation $A_{k,i}$ denotes the set $\{x \in A\cap\Sigma^n \mid \#_0(pref_k(x)) = i\}$ so that $A\cap\Sigma^n$ can be expressed as $A\cap\Sigma^n = \bigcap_{k=0}^{n}\left(\bigcup_{i=0}^{n} A_{k,i}\right)$. Now, we state a key property of $\{A_{k,i}\}_{k,i}$, from which the desired proposition immediately follows.

Claim 1. *There are an index $k \in [m-1,n]_{\mathbb{Z}}$ and at least m distinct indices $(i_1, i_2, \ldots, i_m)$ such that $A_{k,i_j} \neq \emptyset$ for every index $j \in [1,m]_{\mathbb{Z}}$.*

Assuming that Claim 1 is true, let us choose an index $k \in [m-1,n]_{\mathbb{Z}}$ and m distinct indices $(i_1,\ldots,i_m)$ that satisfy the claim. We then choose one string w_j from each set A_{k,i_j} and define $W = \{w_1, w_2, \ldots, w_m\}$. Since $|W| \geq m$, by the swapping lemma, there are two distinct strings x_1x_2 and y_1y_2 in W with $|x_1| = |y_1| = k$ such that the swapped strings x_1y_2 and y_1x_2 belong to A. This leads to a contradiction because the choice of W makes x_1y_2 satisfy $\#_0(x_1y_2) \neq \#_1(x_1y_2)$. This contradiction leads us to conclude that A does not exist, and therefore we finish the proof of Proposition 4.3.

Now, our remaining task is to prove Claim 1. Assume that this claim is false; that is, (*) for each index $k \in [m-1,n]_{\mathbb{Z}}$, there are at most m' indices, say, $(i_1,\ldots,i_{m'})$, where $m' \leq m$, satisfying $A_{k,i_j} \neq \emptyset$ for all indices $j \in [1,m']_{\mathbb{Z}}$. For convenience, we write I_k^* for the set $\{i_1,\ldots,i_{m'}\}$ of such indices. In the rest of the argument, we abbreviate $\lceil m/2\rceil$ as m_0 for brevity. Note that $2m_0 = m+1$. Since m is fixed, we often omit "m_0" and "m".

Toward a contradiction, we intend to estimate the value $|A\cap\Sigma^n|$. Since A is p-dense, we can obtain a lower bound $|A\cap\Sigma^n| \geq 2^n/n^d$ for all but finitely many numbers n. In contrast, the following statement gives an upper bound of $|A\cap\Sigma^n|$.

Claim 2. *There exists a constant c, depending only on m, with $1 < c < 2$ satisfying that $|A\cap\Sigma^n| < c^n$ for all sufficiently large numbers n.*

Together with the p-denseness of A, Claim 2 yields a relation $2^n/n^d \leq |A\cap\Sigma^n| < c^n$, from which we immediately obtain $c > 2n^{-d/n}$. Since $\lim_{n\to\infty} n^{-d/n} = 1$, we reach a conclusion $c \geq 2$, which clearly contradicts the choice of c in Claim 2. Therefore, Claim 1 holds.

To complete the proof of our proposition, we need to prove Claim 2. For this purpose, let us consider all possible sets A that satisfy Condition (*) stated above and let $\mathcal{A}$ denote the collection of all such sets. Now, we want to discuss what kind of $A \in \mathcal{A}$ gives $|A\cap\Sigma^n|$ the largest value. Here is an explicit candidate for such A's. Let $k \geq m-1$. We first define the integer interval $I_k = [\lceil (k+1)/2\rceil - (m_0-1), \lceil (k+1)/2\rceil + (m_0-1)]_{\mathbb{Z}}$ (whose center point is $\lceil (k+1)/2\rceil$) of size m; in particular, $I_{m-1} = [1,m]_{\mathbb{Z}}$. Next, we introduce S_k as the set of all strings $w \in \Sigma^k$ such that, for each index $j \in [m-1,k]_{\mathbb{Z}}$, $\#_0(pref_j(w))$ belongs to I_j. The set $S =_{def} \bigcup_{k\geq m-1} S_k$ clearly falls into $\mathcal{A}$.

In what follows, we shall claim that (1) $|S_n|$ is at most c^n for a certain constant c with $1 < c < 2$ and (2) for every set $A \in \mathcal{A}$, $|S_n|$ upper-bounds $|A\cap\Sigma^n|$. These form the core of our proof. We begin with the first claim by making a direct estimation of the target value $|S_n|$.

Claim 3. *There exists a constant c, depending only on m, with $1 < c < 2$ such that $|S_n| < c^n$ for all sufficiently large numbers $n \in \mathbb{N}$.*

Proof. Recall that m is an odd number at least 5. To estimate each value $|S_e|$, where $m-1 \leq e \leq n$, we first partition S_e into $S_{e,1}, S_{e,2}, \ldots, S_{e,m}$, where $S_{e,i} = \{w \in S_e \mid \#_0(w)$ is the ith element in $I_e\}$ for any index $i \in [1,m]_{\mathbb{Z}}$. Note that "$w \in S_{e,i}$" yields the equation $\#_0(pref_e(w)) = \lceil (e+1)/2\rceil - m_0 + i$. For convenience, we write $a_{e,i}$ to denote the cardinality $|S_{e,i}|$. A simple observation provides the following relations among $S_{e,i}$'s: if e is odd, then $S_{e,i} = \{w0 \mid w \in S_{e-1,i}\} \cup \{w1 \mid w \in S_{e-1,i+1}\}$; otherwise, $S_{e,i} = \{w0 \mid w \in S_{e-1,i-1}\} \cup \{w1 \mid w \in S_{e-1,i}\}$, where we assume that $S_{e-1,m+1} = S_{e-1,0} = \emptyset$. In the rest of this proof, we are focused only on odd values of e.

The aforementioned relations among $S_{e,i}$'s imply that, for any index $k \in [1,(m-3)/2]_{\mathbb{Z}}$,

$$\begin{aligned} &a_{2k+3,1} = 2a_{2k+1,1} + a_{2k+1,2}, \qquad a_{2k+3,m} = a_{2k+1,m-1} + a_{2k+1,m}, \quad \text{and} \\ &a_{2k+3,i} = a_{2k+1,i-1} + 2a_{2k+1,i} + a_{2k+1,i+1}. \end{aligned} \tag{2}$$

[3] [Swapping Lemma for Regular Languages] Let L be any infinite regular language on an alphabet Σ with $|\Sigma| \geq 2$. There exists a positive integer m such that, for any integer $n \geq 1$ and any subset S of $L\cap\Sigma^n$ of cardinality at least m, the following condition holds: for any integer $i \in [0,n]_{\mathbb{Z}}$, there exist two strings $x = x_1x_2$ and $y = y_1y_2$ in S with $|x_1| = |y_1| = i$ and $|x_2| = |y_2|$ satisfying that (i) $x \neq y$, (ii) $y_1x_2 \in L$, and (iii) $x_1y_2 \in L$. See [31].

Notice that $a_{2k+3,m}$ is the smallest and $a_{2k+3,1}$ is the second smallest among $a_{2k+3,i}$'s. Since $|S_{2k+3}| = \sum_{1\le i\le m} |S_{2k+3,i}|$, from Eq. (2), it follows that

$$|S_{2k+3}| = 3a_{2k+1,1} + 2a_{2k+1,m} + 4\sum_{2\le i\le m-1} a_{2k+1,i} \le 3|S_{2k+1}| + \sum_{2\le i\le m-1} a_{2k+1,i}.$$

To calculate $|S_{2k+3}|$, we thus need to estimate the sum $\sum_{2\le i\le m-1} a_{2k+1,i}$ in terms of $|S_{2k+1}|$. Our starting point is the following simple upper bound of $\sum_{2\le i\le m-1} a_{2k+1,i}$ by a certain constant multiple of $a_{2k+3,1} + a_{2k+3,m}$.

Claim 4. *It holds that* $\sum_{i=m_0-j+1}^{m_0+j-1} a_{2k+3,i} \le \delta_j(a_{2k+3,m_0-j} + a_{2k+3,m_0+j})$ *for each index* $j \in [1, m_0-1]_{\mathbb{Z}}$, *where* $\delta_j = 2^{2j-1} - 1$. *In particular,* $\sum_{i=2}^{m-1} a_{2k+3,i} \le \delta_{m_0-1}(a_{2k+3,1} + a_{2k+3,m})$.

Proof. For notational succinctness, we write $b_{e,j}$ for $a_{e,m_0-j} + a_{e,m_0+j}$. Now, we want to show by induction on j that $\sum_{i=m_0-j+1}^{m_0+j-1} a_{2k+3,i} \le \delta_j b_{2k+3,j}$. Consider the basis case $j = 1$. By Eq. (2), it follows that

$$\begin{aligned} b_{2k+3,1} &= a_{2k+1,m_0-2} + 2(a_{2k+1,m_0-1} + a_{2k+1,m_0} + a_{2k+1,m_0+1}) + a_{2k+1,m_0+2} \\ &\ge a_{2k+1,m_0-1} + 2a_{2k+1,m_0} + a_{2k+1,m_0+1} = a_{2k+3,m_0}. \end{aligned}$$

This inequality yields the desired relation $a_{2k+3,m_0} \ge \delta_1 b_{2k+3,1}$ since $\delta_1 = 1$.

Let us consider the induction step j with $2 \le j \le m_0 - 1$. We first discuss the case where $j \ne m_0 - 1$. Note that the sum $\sum_{i=m_0-j+1}^{m_0+j-1} a_{2k+3,i}$ equals

$$a_{2k+1,m_0-j} + 3a_{2k+1,m_0-j+1} + 4\sum_{i=m_0-j+2}^{m_0+j-2} a_{2k+1,i} + 3a_{2k+1,m_0+j-1} + a_{2k+1,m_0+j}.$$

The induction hypothesis on $j-1$ yields $\sum_{i=m_0-j+2}^{m_0+j-2} a_{2k+1,i} \le \delta_j b_{2k+1,j-1}$. With a help of this inequality, the sum $\sum_{i=m_0-j+1}^{m_0+j-1} a_{2k+3,i}$ is bounded from above by

$$\sum_{i=m_0-j+1}^{m_0+j-1} a_{2k+3,i} \le a_{2k+1,m_0-j} + (4\delta_{j-1}+3)(a_{2k+1,m_0-j+1} + a_{2k+1,m_0+j-1}) + a_{2k+1,m_0+j}.$$

Moreover, Eq. (2) gives a lower bound of $b_{2k+3,j}$ as follows:

$$b_{2k+3,j} \ge 2a_{2k+1,m_0-j} + a_{2k+1,m_0-j+1} + a_{2k+1,m_0+j-1} + 2a_{2k+1,m_0+j}.$$

We therefore obtain the bound $\sum_{i=m_0-j+1}^{m_0+j-1} a_{2k+3,i} \le (4\delta_{j-1}+3)b_{2k+3,j}$. Since δ_j satisfies that $\delta_j = 4\delta_{j-1} + 3$, the desired relation immediately follows. The case where $j = m_0 - 1$ is treated similarly with a minor modification. By applying the induction, we obtain the claim. □

By Claim 4, $|S_{2k+1}|$ is lower-bounded by

$$|S_{2k+1}| = a_{2k+1,1} + a_{2k+1,m} + \sum_{2\le i\le m-1} a_{2k+1,i} \ge \left(\frac{1}{\delta_{m_0-1}} + 1\right) \sum_{2\le i\le m-1} a_{2k+1,i},$$

from which we obtain $\sum_{2\le i\le m-1} a_{2k+1,i} \le \gamma|S_{2k+1}|$ if we set $\gamma = 1/(1/\delta_{m_0-1} + 1) < 1$. We therefore conclude that $|S_{2k+3}| \le 3|S_{2k+1}| + \sum_{2\le i\le m-1} a_{2k+1,i} \le (3+\gamma)|S_{2k+1}|$. This recurrence has a solution $|S_n| \le (3+\gamma)^{(n-m)/2}|S_m|$ for every *odd* number $n \ge m$. Since $|S_{2k+2}| \le |S_{2k+3}|$ and $m \ge 5$, it holds that $|S_n| \le (3+\gamma)^{n/2}|S_m|$ for *all* numbers $n \ge 1$. In this end, the fact that $|S_m|$ is a constant and $1 < (3+\gamma)^{1/2} < 2$ leads to Claim 3. □

Finally, we want to prove the second claim that $|A \cap \Sigma^n| \le |S_n|$. In Claim 5, we actually prove a much stronger statement. To describe this claim, we shall explain new terminology. Let $k \in [m-1, n]_{\mathbb{Z}}$ be an arbitrary number. A *convergence point* is an m-tuple $(d_1, d_2, \ldots, d_m)$ that satisfies the following conditions: (i) for all indices $i \in [1, m]_{\mathbb{Z}}$, d_i is in $\mathbb{N}$ and (ii) $d_1 \le d_2 \le \cdots \le d_m$. For any two convergence points $(d_1, d_2, \ldots, d_m)$ and $(d'_1, d'_2, \ldots, d'_m)$, we say that $(d_1, d_2, \ldots, d_m)$ *majorizes* $(d'_1, d'_2, \ldots, d'_m)$ if, for every index $k \in [1, m]_{\mathbb{Z}}$, $\sum_{k\le i\le m} d_i \ge \sum_{k\le i\le m} d'_i$. This majorization notion directly implies that $\sum_{1\le i\le m} d_i \ge \sum_{1\le i\le m} d'_i$.

Let us recall that $a_{k,i}$ denotes $|S_{k,i}|$. Among $a_{k,i}$'s, the following relation holds: when k is odd, $a_{k,m} \le a_{k,1} \le a_{k,m-1} \le a_{k,2} \le \cdots \le a_{k,m_0}$ and, when k is even, $a_{k,1} \le a_{k,m} \le a_{k,2} \le a_{k,m-1} \le \cdots \le a_{k,m_0}$. To simplify the description of $a_{k,i}$'s in these enumerations, we introduce another notation $\tilde{a}_{k,i}$ to denote the ith element in the corresponding enumeration; thus, for *every* index k, $\tilde{a}_{k,1} \le \tilde{a}_{k,2} \le \cdots \le \tilde{a}_{k,m}$. The m-tuple $(\tilde{a}_{k,1}, \tilde{a}_{k,2}, \ldots, \tilde{a}_{k,m})$ becomes a convergence point. It is not difficult to show by induction that, for any index $i \in [1, m]_{\mathbb{Z}}$, $\tilde{a}_{k,i} = \tilde{a}_{k-1,i-1} + \tilde{a}_{k-1,i+1}$, where we conveniently set $\tilde{a}_{k-1,0} = 0$ and $\tilde{a}_{k-1,m+1} = \tilde{a}_{k-1,m}$.

Associated with $A_{k,i}$, we introduce another notation $A^*_{k,i}$, analogous to $S_{k,i}$'s, to denote the set $\{pref_k(x) \mid x \in A \cap \Sigma^n, \#_0(pref_k(x)) = i\}$ and let $A^*_k = \bigcup_{i\in I^*_k} A^*_{k,i}$. Without loss of generality, we can assume that $|I^*_k| = m$ (because, otherwise, we add appropriate elements to I^*_k). In general, there may be a situation in which $w_1, w_2 \in A^*_{k-1,i}$ and $w_1 b \in A^*_{k,j}$ but

$w_2 b \notin A^*_{k,j}$ for certain elements w_1, w_2, b. Clearly, this situation decreases the value $|A^*_{k,j}|$; hereafter, it suffices to assume that this situation never occurs.

To simplify our description in the following argument, we enumerate all $A^*_{k,i}$'s as $B_{k,j}$'s so that $|B_{k,1}| \leq |B_{k,2}| \leq \cdots \leq |B_{k,m}|$. Obviously, $(|B_{k,1}|, |B_{k,2}|, \ldots, |B_{k,m}|)$ becomes a convergence point. Toward the desired result $|A \cap \Sigma^n| \leq |S_n|$, since $|A \cap \Sigma^n| = \sum_{1\leq i\leq m} |B_{n,i}|$ and $|S_n| = \sum_{1\leq i\leq m} \tilde{a}_{n,i}$, it is enough to show that $(\tilde{a}_{k,1}, \tilde{a}_{k,2}, \ldots, \tilde{a}_{k,m})$ majorizes $(|B_{k,1}|, |B_{k,2}|, \ldots, |B_{k,m}|)$.

Claim 5. *Let $k \in [m-1, n]_{\mathbb{Z}}$ and let $A \in \mathcal{A}$. Consider $A^*_{k,1}, A^*_{k,2}, \ldots, A^*_{k,m}$ induced from $A \cap \Sigma^n$ as described before. Let $B_{k,1}, B_{k,2}, \ldots, B_{k,m}$ be an enumeration of $A^*_{k,i}$'s so that $|B_{k,1}| \leq |B_{k,2}| \leq \cdots \leq |B_{k,m}|$. It then holds that $(\tilde{a}_{k,1}, \tilde{a}_{k,2}, \ldots, \tilde{a}_{k,m})$ majorizes $(|B_{k,1}|, |B_{k,2}|, \ldots, |B_{k,m}|)$. Thus, in particular, $|A \cap \Sigma^n| \leq |S_n|$ holds.*

Our proof of Claim 5 is comprised of two extra claims—Claims 6 and 7.

Claim 6. *Let $(d_1, d_2, \ldots, d_m)$, $(c_1, c_2, \ldots, c_m)$ be any two convergence points. For every index $i \in [1, m]_{\mathbb{Z}}$, define $\tilde{d}_i = d_{i-1} + d_{i+1}$ with $d_0 = 0$ and $d_{m+1} = d_m$ and define $\tilde{c}_i = c_{i-1} + c_{i+1}$ with $c_0 = 0$ and $c_{m+1} = c_m$. If $(d_1, d_2, \ldots, d_m)$ majorizes $(c_1, c_2, \ldots, c_m)$, then $(\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_m)$ majorizes $(\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_m)$.*

Since the proof of this claim is rather short, we shall give it here. Let k be any index in $[1, m]_{\mathbb{Z}}$. Since $(d_1, \ldots, d_m)$ majorizes $(c_1, \ldots, c_m)$, it holds that $\sum_{k-1\leq i\leq m} d_i \geq \sum_{k-1\leq i\leq m} c_i$ and $\sum_{k+1\leq i\leq m} d_i \geq \sum_{k+1\leq i\leq m} c_i$. Let us consider the difference $\ell_k =_{def} \sum_{k\leq i\leq m} \tilde{d}_i - \sum_{k\leq i\leq m} \tilde{c}_i$. It is clear that $\sum_{k\leq i\leq m} \tilde{d}_i$ equals $\sum_{k-1\leq i\leq m} d_i + \sum_{k+1\leq i\leq m} d_i$. A similar equality also holds for $\tilde{c}_i$'s. We thus conclude that $\ell_k = \left(\sum_{k-1\leq i\leq m} d_i - \sum_{k-1\leq i\leq m} c_i\right) + \left(\sum_{k+1\leq i\leq m} d_i - \sum_{k+1\leq i\leq m} c_i\right) \geq 0$. Therefore, $(\tilde{d}_1, \ldots, \tilde{d}_m)$ majorizes $(\tilde{c}_1, \ldots, \tilde{c}_m)$.

Claim 7. *Let $k \in [m-1, n]_{\mathbb{Z}}$ and let $A \in \mathcal{A}$. Assume that $B_{k-1,1}, B_{k-1,2}, \ldots, B_{k-1,m}$ and $B_{k,1}, B_{k,2}, \ldots, B_{k,m}$ are induced from $A \cap \Sigma^n$. For each index $i \in [1, m]_{\mathbb{Z}}$, define $B'_{k,i}$ so that $|B'_{k,i}| = |B_{k-1,i-1}| + |B_{k-1,i+1}|$, where we set $B_{k-1,0} = \emptyset$ and $B_{k-1,m+1} = B_{k-1,m}$. It then holds that $(|B'_{k,1}|, |B'_{k,2}|, \ldots, |B'_{k,m}|)$ majorizes $(|B_{k,1}|, |B_{k,2}|, \ldots, |B_{k,m}|)$.*

Before proving Claim 7, we shall give the proof of Claim 5 using Claims 6 and 7. The proof proceeds by induction on $k \in [m-1, n]_{\mathbb{Z}}$. For the basis case $k = m-1$, note that $I_{m-1} = [1, m]_{\mathbb{Z}}$. For each index $i \in [0, m-1]_{\mathbb{Z}}$, since $A^*_{m-1,i} = \{w \in \Sigma^{m-1} \mid \exists v[wv \in A \cap \Sigma^n], \#_0(w) = i\}$, $A^*_{m-1,i}$ is clearly included in the set $\{w \in \Sigma^{m-1} \mid \#_0(w) = i\}$, which equals $S_{m-1,i+1}$. Hence, we have $|A^*_{m-1,i}| \leq |S_{m-1,i+1}|$. Since $|B_{k,i}|$'s are an enumeration of $|A^*_{k,i}|$'s in an increasing order, we obtain $|B_{m-1,j}| \leq \tilde{a}_{m-1,j}$ for every index $j \in [1, m]_{\mathbb{Z}}$.

For induction step $k \geq m$, we choose m sets $B'_{k,1}, B'_{k,2}, \ldots, B'_{k,m}$, each of which satisfies the equation $|B'_{k,i}| = |B_{k-1,i-1}| + |B_{k-1,i+1}|$, where $i \in [1, m]_{\mathbb{Z}}$. Claim 7 guarantees that $(|B'_{k,1}|, \ldots, |B'_{k,m}|)$ majorizes $(|B_{k,1}|, \ldots, |B_{k,m}|)$. By induction hypothesis, $(\tilde{a}_{k-1,1}, \ldots, \tilde{a}_{k-1,m})$ majorizes $(|B_{k-1,1}|, \ldots, |B_{k-1,m}|)$. This implies, by Claim 6, that $(\tilde{a}_{k,1}, \ldots, \tilde{a}_{k,m})$ majorizes $(|B'_{k,1}|, \ldots, |B'_{k,m}|)$. By combining these relations, it follows that $(\tilde{a}_{k,1}, \ldots, \tilde{a}_{k,m})$ majorizes $(|B_{k,1}|, \ldots, |B_{k,m}|)$, completing the proof of Claim 5.

Proof of Claim 7. Our proof strategy is described as follows. The proof of the claim will proceed by induction on $i \in [1, m]_{\mathbb{Z}}$. For each index $i \in [1, m]_{\mathbb{Z}}$, by choosing appropriate $A^*_{k,i}$'s, we first try to maximize the value $\sum_{i\leq j\leq m} |B_{k,j}|$ and then maximize the next value $\sum_{i+1\leq j\leq m} |B_{k,j}|$; for those maximal values, we want to prove that $|B'_{k,i}| = |B_{k,i}|$.

For our proof, it is helpful to visualize a relationship between $A^*_{k-1,i}$'s and $A^*_{k,j}$'s using a *directed bipartite graph* $G = (V_1|V_2, E)$, whose nodes in V_1 are labeled $A^*_{k-1,i}$ $(i \in I^*_{k-1})$ and nodes in V_2 are labeled $A^*_{k,j}$ $(j \in I^*_k)$. For simplicity, we identify a node name with its label. There is a directed edge in E from node $A^*_{k-1,i}$ to node $A^*_{k,j}$ (in this case, $A^*_{k-1,i}$ is conventionally said to be *incident* to $A^*_{k,j}$) exactly when certain elements w and b satisfy that $w \in A^*_{k-1,i}$ and $wb \in A^*_{k,j}$. Notationally, we write $outdeg(a)$ for the *outdegree* (i.e., the number of outgoing edges from a) of a graph node a, and $indeg(a)$ for the *indegree* (i.e., the number of incoming edges to a) of a. The following argument uses structural properties of a bipartite graph of both outdegree and indegree at most 2.

[Basis Case: $i = 1$] By the definition of $B'_{k,j}$'s, it holds that $\sum_{1\leq j\leq m} |B'_{k,j}| = 2\sum_{2\leq j\leq m} |B_{k-1,j}| + |B_{k-1,1}|$. Recall that $|A^*_k| = \sum_{1\leq j\leq m} |B_{k,j}|$. First, we want to force $|A^*_k|$ to take the largest value. Note that every index i in I^*_{k-1} can be classified into one of the following two index sets: $I'_1 =_{def} \{i \in I^*_{k-1} \mid outdeg(A^*_{k-1,i}) = 1\}$ and $I'_2 =_{def} \{i \in I^*_{k-1} \mid outdeg(A^*_{k-1,i}) = 2\}$. Since $|A^*_k| \leq 2\sum_{j\in I'_2} |A^*_{k-1,j}| + \sum_{j\in I'_1} |A^*_{k-1,j}|$, we should choose an index $i_0 \in I^*_{k-1}$ so that $|A^*_{k-1,i_0}|$ is the smallest value among $|A^*_{k-1,j}|$'s, and then we should set $I'_1 = \{i_0\}$. In summary, we have $I'_2 = I^*_{k-1} - \{i_0\}$ and $|A^*_k| = 2\sum_{j\in I'_2} |A^*_{k-1,j}| + |A^*_{k-1,i_0}|$. Since $|A^*_{k-1,i_0}| = |B_{k-1,1}|$, we thus obtain $|A^*_k| = 2\sum_{2\leq j\leq m} |B_{k-1,j}| + |B_{k-1,1}|$. This means $outdeg(B_{k-1,1}) = 1$, and therefore G is not composed of two or more disconnected subgraphs.

To maximize the next sum $\sum_{2\leq j\leq m} |B_{k,j}|$, since the value $|A^*_k|$ is already fixed, we need to minimize the value $|B_{k,1}|$. For this purpose, we demand that $indeg(B_{k,1}) = 1$. Which node in V_1, incident to node $B_{k,1}$, can minimize $|B_{k,1}|$? At the first sight, it seems that node $B_{k-1,1}$ could be the best choice; however, as we show next, it cannot be incident to $B_{k,1}$. Let us assume that $(B_{k-1,1}, B_{k,1}) \in E$. Since $outdeg(B_{k-1,1}) = indeg(B_{k,1}) = 1$, the node set $\{B_{k-1,1}, B_{k,1}\}$ forms a subgraph, which is entirely disconnected from the other part of the graph G. This implies the existence of another node in V_1 of outdegree exactly 1, a clear contradiction against $|I'_1| = 1$. Hence, since the second best choice is node $B_{k-1,2}$, E should contain edge $(B_{k-1,2}, B_{k,1})$; thus, $|B_{k,1}|$ equals $|B_{k-1,2}|$, which is $|B'_{k,1}|$ by its definition.

[Induction Case: $i \geq 2$] We first consider the case where $i \neq m$. Because the sum $\sum_{i\leq j\leq m} |B_{k,j}|$ has been maximized at Step $i-1$, to maximize the value $\sum_{i+1\leq j\leq m} |B_{k,j}|$, we should force $|B_{k,i}|$ smaller. Since $indeg(B_{k,i}) = 2$, let us consider a node pair in V_1 that are incident to $B_{k,i}$. Since nodes $B_{k-1,1}, B_{k-1,2}, \ldots, B_{k-1,i-2}$ are already used up in the previous steps, the possible choice of nodes incident to $B_{k,i}$ includes $B_{k-1,i-1}, B_{k-1,i}, \ldots, B_{k-1,m}$. We argue that E does not contain both edges $(B_{k-1,i-1}, B_{k,i})$ and $(B_{k-1,i}, B_{k,i})$ simultaneously. If E contains them, then the node set $\{B_{k-1,1}, \ldots, B_{k-1,i}, B_{k,1}, \ldots, B_{k,i}\}$ forms a subgraph, say, G' of G. Recall that $outdeg(B_{k-1,1}) = 1$ and $indeg(B_{k,1}) = 1$. This implies that G' is disconnected from the rest of the graph G. This is a contradiction against the nature of G. Hence, the second best choice for a node pair incident to $B_{k,i}$ is $\{B_{k-1,i-1}, B_{k-1,i+1}\}$. This concludes that $|B_{k,i}| = |B_{k-1,i-1}| + |B_{k-1,i+1}|$, and thus $|B_{k,i}|$ equals $|B'_{k,i}|$, as requested. If $i = m$, then nodes $B_{k-1,m-1}$ and $B_{k-1,m}$ are the only choice of nodes incident to $B_{k,m}$. Thus, $|B_{k,m}|$ equals $|B_{k-1,m-1}| + |B_{k-1,m}|$, which is exactly $|B'_{k,m}|$. □

This completes the proof of Proposition 4.3. □

Unlike the REG-bi-immunity, it is possible to prove the existence of context-free REG/n-bi-primeimmune languages. A later result in Section 5 implies that a context-free language, called IP_*, is REG/n-bi-primeimmune.

5. Pseudorandomness of languages

From this section to the next section, we shall discuss "computational randomness" of context-free languages. Although there are numerous ways to describe the intuitive notion of computational randomness, we choose the following notion, which we prefer to call $\mathcal{C}$-*pseudorandomness* to distinguish another notion of "$\mathcal{C}$-randomness" used in the past literature. Let Σ denote our alphabet with $|\Sigma| \geq 2$ and let $\mathcal{C}$ be any language family. Roughly speaking, a language L over Σ is $\mathcal{C}$-pseudorandom when the characteristic function χ_A of any language A in $\mathcal{C}$ agrees with χ_L over "nearly" 50% of strings of each length, where the word "nearly" is meant for "negligibly small margin." In other words, since $L \triangle A = \{x \in \Sigma^* \mid \chi_L(x) \neq \chi_A(x)\}$, the density $dense(L\triangle A)(n)$ "nearly" halves the total size $|\Sigma^n|$. This new notion can be seen as a non-asymptotic variant of Wilber's randomness [30] (which is also referred to as Wilber-stochasticity in [2]) and Meyer–McCreight's randomness [24].

Let us formalize the above intuitive notion. For any language L over Σ, we say that L is $\mathcal{C}$-*pseudorandom* if, for each language A over Σ in $\mathcal{C}$, the function $\ell(n) =_{def} \left| \frac{dense(L\triangle A)(n)}{|\Sigma^n|} - \frac{1}{2} \right|$ is negligible. Under the assumption that $\emptyset \in \mathcal{C}$, we can show, by setting $A = \emptyset$, that every $\mathcal{C}$-pseudorandom language L satisfies

$$\left(\frac{1}{2} - \frac{1}{p(n)}\right)\left|\Sigma^n\right| \leq dense(L)(n) \leq \left(\frac{1}{2} + \frac{1}{p(n)}\right)\left|\Sigma^n\right| \tag{3}$$

for any positive polynomial p and for all but finitely many lengths $n \in \mathbb{N}$. Instead of assuming "$\emptyset \in \mathcal{C}$," the assumption "$\Sigma^* \in \mathcal{C}$" also leads to Eq. (3), by way of dealing with $\overline{L}$.

Similar in spirit to the previous $\mathcal{C}$-primeimmunity, we can naturally restrict our attention within p-dense languages in $\mathcal{C}$. As a non-asymptotic variant of the notions of Müller's balanced immunity [25] and weak-stochasticity of Ambos-Spies et al. [2], we introduce another notion, called weak $\mathcal{C}$-pseudorandomness, which refers to a language that splits every p-dense set in $\mathcal{C}$ by "nearly" half. Let $\mathcal{C}$ be any language family. Formally, a language L over Σ is called *weakly* $\mathcal{C}$-*pseudorandom* if, for every p-dense language A in $\mathcal{C}$, the function $\ell'(n) =_{def} \left| \frac{dense(L\cap A)(n)}{dense(A)(n)} - \frac{1}{2} \right|$ is negligible. By choosing $A = \Sigma^*$, provided that $\Sigma^* \in \mathcal{C}$, we can show that L also satisfies Eq. (3).

We remarks that no (weakly) $\mathcal{C}$-pseudorandom language belongs to $\mathcal{C}$. A language family $\mathcal{D}$ is said to be $\mathcal{C}$-*pseudorandom* (resp., *weakly* $\mathcal{C}$-*pseudorandom*) if $\mathcal{D}$ contains a $\mathcal{C}$-pseudorandom (resp., weakly $\mathcal{C}$-pseudorandom) language. In fact, as we shall show later, CFL is REG-pseudorandom.

Lemma 5.1. *Assume that $|\Sigma| \geq 2$. Let $\mathcal{C}$ be any language family with $\Sigma^* \in \mathcal{C}$. For every set $S \subseteq \Sigma^*$, the following three statements are equivalent.*

1. *S is weakly $\mathcal{C}$-pseudorandom.*
2. *The function $\ell(n) = \left| \frac{dense(S\triangle A)(n)}{|\Sigma^n|} - \frac{1}{2} \right|$ is negligible for every p-dense language $A \in \mathcal{C}$ over Σ.*
3. *The function $\ell''(n) = \left| \frac{dense(S\cap A)(n)}{|\Sigma^n|} - \frac{dense(\overline{S}\cap A)(n)}{|\Sigma^n|} \right|$ is negligible for every p-dense language $A \in \mathcal{C}$ over Σ.*

In the above lemma, Statements (2) and (3) are still equivalent after removing a requirement of the p-denseness of A. With an appropriate change, we therefore obtain a similar characterization of the $\mathcal{C}$-pseudorandomness. For a later reference, we call this fact a "pseudorandom" version of Lemma 5.1(2–3).

Hereafter, we use the following abbreviation: write S_n for $S \cap \Sigma^n$ and $\overline{S}_n$ for $\overline{S} \cap \Sigma^n$.

Proof of Lemma 5.1. Let Σ be our alphabet with $|\Sigma| \geq 2$ and let S be any language over Σ. Notice that a language family $\mathcal{C}$ is assumed to contain the language Σ^*.

(1 $\Rightarrow$ 2) Assume Statement (1). Choose an arbitrary positive polynomial p and also any p-dense language A in $\mathcal{C}$. Henceforth, we assume that n is a sufficiently large number.

We first claim that $|2|S_n \triangle A_n| - |\Sigma^n|| \geq 2|\Sigma^n|/p(n)$. From Statement (1) follows the inequality $\left|\frac{dense(S\cap A)(n)}{dense(A)(n)} - \frac{1}{2}\right| \leq 1/4p(n)$, which is equivalent to $\left||A_n \cap S_n| - |A_n \cap \overline{S}_n|\right| \leq |A_n|/2p(n)$. Since S satisfies Eq. (3), using $2p(n)$ (instead of $p(n)$), we obtain $\left|\frac{|S_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq 1/2p(n)$. It is easy to show that $\left||S_n| - |\overline{S}_n|\right| \leq |\Sigma^n|/p(n)$, since $|\overline{S}_n| = |\Sigma^n| - |S_n|$. From $|\overline{A}_n \cap S_n| = |S_n| - |A_n \cap S_n|$ and $|\overline{A}_n \cap \overline{S}_n| = |\overline{S}_n| - |A_n \cap \overline{S}_n|$, we conclude that

$$\left||\overline{A}_n \cap S_n| - |\overline{A}_n \cap \overline{S}_n|\right| \leq \left||S_n| - |\overline{S}_n|\right| + \left||A_n \cap S_n| - |A_n \cap \overline{S}_n|\right|.$$

Since $|S_n \triangle A_n| = |A_n \cap \overline{S}_n| + |\overline{A}_n \cap S_n|$ and $|\overline{S_n \triangle A_n}| = |A_n \cap S_n| + |\overline{A}_n \cap \overline{S}_n|$, it follows that

$$\begin{aligned}\left|2|S_n \triangle A_n| - |\Sigma^n|\right| &= \left||S_n \triangle A_n| - |\overline{S_n \triangle A_n}|\right| \\ &\leq \left||A_n \cap S_n| - |A_n \cap \overline{S}_n|\right| + \left||\overline{A}_n \cap S_n| - |\overline{A}_n \cap \overline{S}_n|\right| \\ &\leq \left||S_n| - |\overline{S}_n|\right| + 2\left||A_n \cap S_n| - |A_n \cap \overline{S}_n|\right|.\end{aligned}$$

The last sum is bounded from above by $\frac{|\Sigma^n|}{p(n)} + \frac{|A_n|}{p(n)} \leq \frac{2|\Sigma^n|}{p(n)}$. Using this upper bound, we obtain

$$\ell(n) = \left|\frac{dense(S\triangle A)(n)}{|\Sigma^n|} - \frac{1}{2}\right| = \frac{|2|S_n \triangle A_n| - |\Sigma^n||}{2|\Sigma^n|} \leq \frac{1}{p(n)}.$$

Since p is arbitrary, the above bound of $\ell(n)$ clearly implies Statement (2).

(2 ⇒ 3) Assume Statement (2). Let p be any positive polynomial and let A be any p-dense language in $\mathcal{C}$. Statement (2) implies that $\ell(n) = \left|\frac{|S_n \triangle A_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq 1/2p(n)$ for any sufficiently large number n. Since $\Sigma^* \in \mathcal{C}$ and $S_n \triangle \Sigma^n = \overline{S}_n$, it holds that $\left|\frac{|\overline{S}_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq 1/2p(n)$. This immediately implies $\left|\frac{|S_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq 1/2p(n)$. Hence, since $\left||S_n \cap A_n| - |\overline{S}_n \cap A_n|\right| = ||S_n \triangle A_n| - |S_n||$, we can bound the term $\ell''(n)$ as

$$\ell''(n) = \frac{\left||S_n \cap A_n| - |\overline{S}_n \cap A_n|\right|}{|\Sigma^n|} \leq \left|\frac{|S_n \triangle A_n|}{|\Sigma^n|} - \frac{1}{2}\right| + \left|\frac{|S_n|}{|\Sigma^n|} - \frac{1}{2}\right|,$$

which is further upper-bounded by $\frac{1}{2p(n)} + \frac{1}{2p(n)} = \frac{1}{p(n)}$. Therefore, Statement (3) holds.

(3 ⇒ 1) Assume Statement (3). For any positive polynomial p and any p-dense language A in $\mathcal{C}$, take a certain non-zero polynomial q satisfying that $|A_n| \geq |\Sigma^n|/2q(n)$ for any sufficiently large number n. We then obtain

$$\ell'(n) = \left|\frac{|S_n \cap A_n|}{|A_n|} - \frac{1}{2}\right| = \left|\frac{|S_n \cap A_n| - |\overline{S}_n \cap A_n|}{2|A_n|}\right| \leq q(n) \cdot \left|\frac{|S_n \cap A_n| - |\overline{S}_n \cap A_n|}{|\Sigma^n|}\right|.$$

Since $\left|\frac{|S_n\cap A_n| - |\overline{S}_n \cap A_n|}{|\Sigma^n|}\right| \leq 1/p(n)q(n)$ from Statement (3), the above inequality implies that $\ell'(n) \leq 1/p(n)$. The arbitrariness of p leads to a conclusion that $\ell'(n)$ is negligible, or equivalently Statement (1) holds. □

From Lemma 5.1, we can draw the following consequence for any language family $\mathcal{C}$ containing Σ^*: every $\mathcal{C}$-pseudorandom language is weakly $\mathcal{C}$-pseudorandom. We further argue that weak $\mathcal{C}$-pseudorandomness implies $\mathcal{C}$-bi-primeimmunity. This implication bridges between primeimmunity and pseudorandomness.

Lemma 5.2. *Let $\mathcal{C}$ be any language family with Σ^*. Every weakly $\mathcal{C}$-pseudorandom language is $\mathcal{C}$-bi-primeimmune.*

Proof. Let S be any weakly $\mathcal{C}$-pseudorandom language. Assuming that S is not $\mathcal{C}$-primeimmune, we take a p-dense subset A of S in $\mathcal{C}$. Since $A \subseteq S$, it follows that $\ell'(n) = \left|\frac{|S_n\cap A_n|}{|A_n|} - \frac{1}{2}\right| = \left|\frac{|A_n|}{|A_n|} - \frac{1}{2}\right| \geq 1/2$, which is clearly not negligible. This is a contradiction against the weak $\mathcal{C}$-pseudorandomness of S. Hence, S is indeed $\mathcal{C}$-primeimmune.

Next, we consider the case of $\overline{S}$. Note that, as a symmetric feature of Lemma 5.1(3) indicates, $\overline{S}$ also becomes weakly $\mathcal{C}$-pseudorandom. Thus, an argument used for S works analogously for $\overline{S}$. In the end, we conclude that S is $\mathcal{C}$-bi-primeimmune, as requested. □

The converse of Lemma 5.2, however, does not hold in general; for instance, there are context-free languages that are REG-primeimmune but not weakly REG-pseudorandom. One of those languages is the language $Equal_*$, defined in Section 4.

Proposition 5.3. *The language family* CFL *contains a* REG*-primeimmune language that is not weakly* REG*-pseudorandom.*

Proof. In Proposition 4.3, the context-free language $Equal_*$ is shown to be REG/n-primeimmune (and thus REG-primeimmune). Hence, our remaining task is to show that $Equal_*$ is not weakly REG-pseudorandom. Choose $A = \Sigma^*$ and consider the function $\ell(n) = \left|\frac{dense(Equal_*\triangle A)(n)}{|\Sigma^n|} - \frac{1}{2}\right|$. Since $dense(Equal_*\triangle A)(n) = dense(Equal_*)(n) \leq \binom{n}{\lceil n/2\rceil}$, for any sufficiently large number n, $\ell(n)$ is bounded from below by $\frac{1}{2} - \frac{dense(Equal_*)(n)}{2^n} \geq \frac{1}{2} - \frac{\binom{n}{\lceil n/2\rceil}}{2^n} \geq \frac{1}{4}$ because $\binom{n}{\lceil n/2\rceil} \leq \frac{2^{n+1}\sqrt{2}}{\sqrt{\pi n}} \leq \frac{2^{n+1}}{\sqrt{n}}$. Since $\ell(n) \geq 1/4$, $Equal_*$ cannot be weakly REG-pseudorandom. □

Proposition 4.3 has proven CFL to be REG/n-primeimmune. We shall strengthen this result by proving that CFL is actually REG/n-pseudorandom.

Proposition 5.4. *The language family* CFL *is* REG/n*-pseudorandom.*

To prove Proposition 5.4, we introduce a context-free language, called IP_*, over the alphabet $\{0, 1\}$. First, let us define the *(binary) inner product* of x and y as $x \odot y = \sum_{i=1}^{n} x_i \cdot y_i$, where $x = x_1x_2 \cdots x_n$ and $y = y_1y_2 \cdots y_n$ are n-bit strings. The language IP_* is defined as the set $\{auv \mid a \in \{\lambda, 0, 1\}, |u| = |v|, u^R \odot v \equiv 1 \pmod 2\}$. Here, we shall demonstrate that IP_* is indeed context-free. Let us consider the following npda. On input a string of the form auv, we nondeterministically generate two computational paths and check the following two possibilities. Along one computation path, assuming that $a = \lambda$, we nondeterministically check if $|u| = |v|$ and $u^R \odot v \equiv 1 \pmod 2$. The latter condition $u^R \odot v \equiv 1 \pmod 2$ can be checked by storing u in a (last-in first-out) stack and then computing each product $u_{n/2-i} \odot v_i$ while reading v_i, where $i = 1, 2, \ldots, n/2$. On the other computation path, assuming that $a \neq \lambda$, we ignore the first bit a and check if $|u| = |v|$ and $u^R \odot v \equiv 1 \pmod 2$. It is easy to see that this npda recognizes IP_*.

Our proof of Proposition 5.4 requires a certain unique property of REG/n, called a *swapping property*, which has a loose similarity with the swapping lemma for regular languages [31].

Lemma 5.5 (*Swapping Property Lemma*)**.** *Let S be any language over an alphabet Σ. If $S \in$ REG/n, then there exists a positive integer m that satisfies the following property. For any three numbers $n, \ell_1(n), \ell_2(n) \in \mathbb{N}$ with $\ell_1(n) + \ell_2(n) = n$, there are a group of disjoint sets, say, $S_1^{(n)}, S_2^{(n)}, \ldots, S_m^{(n)}$ such that* (i) $S \cap \Sigma^n = \bigcup_{i=1}^{m} S_i^{(n)}$ *and* (ii) *(swapping property) for any index $i \in [1, m]_{\mathbb{Z}}$ and for any string pair $x, y \in S_i^{(n)}$, if $x = x_1x_2$ and $y = y_1y_2$ with $|x_j| = |y_j| = \ell_j(n)$ for each index $j \in \{1, 2\}$, then the swapped strings x_1y_2 and y_1x_2 are in $S_i^{(n)}$.*

Proof. From our assumption $S \in$ REG/n, we choose a dfa M with a set Q of inner states, and an advice function $h : \mathbb{N} \to \Gamma^*$ with $|h(n)| = n$ satisfying that, for every string $x \in \Sigma^*$, $x \in S$ iff M accepts $\left[\begin{smallmatrix} x \\ h(|x|) \end{smallmatrix}\right]$. Let us assume that $Q = \{q_1, q_2, \ldots, q_m\}$ with $m \geq 1$. For any numbers $n, \ell_1(n), \ell_2(n) \in \mathbb{N}$ with $\ell_1(n) + \ell_2(n) = n$, we define $S_i^{(n)}$ as the set of strings $x_1x_2 \in S \cap \Sigma^n$ such that $|x_1| = \ell_1(n)$, $|x_2| = \ell_2(n)$, and M enters q_i after reading $\left[\begin{smallmatrix} x_1 \\ h_1 \end{smallmatrix}\right]$, where h_1 denotes $pref_{|x_1|}(h(n))$. It is clear that $S \cap \Sigma^n = \bigcup_{i=1}^{m} S_i^{(n)}$. If x_1x_2 and y_1y_2 are in $S_i^{(n)}$, then M enters the same state q_i after both reading $\left[\begin{smallmatrix} x_1 \\ h_1 \end{smallmatrix}\right]$ and reading $\left[\begin{smallmatrix} y_1 \\ h_1 \end{smallmatrix}\right]$. Since M accepts the both strings $\left[\begin{smallmatrix} x_1x_2 \\ h(n) \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} y_1y_2 \\ h(n) \end{smallmatrix}\right]$, M also accepts both $\left[\begin{smallmatrix} x_1y_2 \\ h(n) \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} y_1x_2 \\ h(n) \end{smallmatrix}\right]$. Therefore, x_1y_2 and y_1x_2 belong to $S_i^{(n)}$. □

Now, we are ready to present the proof of Proposition 5.4. In the proof, we shall utilize a well-known discrepancy upper bound of the *inner-product-modulo-two* function.

Proof of Proposition 5.4. Our goal is to show that IP_* is REG/n-pseudorandom. Assume on the contrary that, by a "pseudorandom" version of Lemma 5.1(2–3), there are a set S in REG/n, a positive polynomial p, and an infinite set $I \subseteq \mathbb{N}$ such that $\ell''(n) = \frac{|dense(IP_* \cap S)(n) - dense(\overline{IP}_* \cap S)(n)|}{|\Sigma^n|} \geq 1/p(n)$ for all lengths n in I. Take a positive constant m given in Lemma 5.5. Let n be any sufficiently large number in I satisfying $m < 2^{n/8}$ and $p(n) < 2^{n/8}$, and consider any n-bit input string of the form auv. It is sufficient to check the case where n is even (that is, $a = \lambda$), because, when n is odd, we can ignore the first bit a and reduce this case to the even-number case. For ease of notation, abbreviate $S \cap IP_* \cap \Sigma^n$ and $S \cap \overline{IP}_* \cap \Sigma^n$ by U_1 and U_0, respectively. From our assumption, it follows that $||U_1| - |U_0|| = \ell''(n)|\Sigma^n| \geq 2^n/p(n)$ since $\Sigma = \{0, 1\}$.

By setting $\ell_1(n) = \ell_2(n) = n/2$, we choose $S_1^{(n)}, \ldots, S_m^{(n)}$ given by Lemma 5.5, and consider two partitions: $U_0 = \bigcup_{i \in [1,m]_{\mathbb{Z}}} U_0^{(i)}$ and $U_1 = \bigcup_{i \in [1,m]_{\mathbb{Z}}} U_1^{(i)}$, where $U_1^{(i)} = IP_* \cap S_i^{(n)}$ and $U_0^{(i)} = \overline{IP}_* \cap S_i^{(n)}$. Toward our desired contradiction, we aim at proving the inequality $||U_1| - |U_0|| < 2^n/p(n)$. For this purpose, we claim the following.

Claim 8. *For all indices $i \in [1, m]_{\mathbb{Z}}$, $\left||U_1^{(i)}| - |U_0^{(i)}|\right| \leq 2^{3n/4}$.*

From this claim, since $m < 2^{n/8}$, it follows that $||U_1| - |U_0|| \leq \sum_{i \in [1,m]_{\mathbb{Z}}} \left||U_1^{(i)}| - |U_0^{(i)}|\right| \leq m \cdot 2^{3n/4} < 2^{7n/8} < \frac{2^n}{p(n)}$. This consequence obviously contradicts our assumption that $||U_1| - |U_0|| \geq 2^n/p(n)$. Hence, the proposition follows immediately.

Now, we give the proof of Claim 8. For this proof, we need a discrepancy upper bound of the inner-product-modulo-two function. Let M be a $\Sigma^{n/2}$-by-$\Sigma^{n/2}$ matrix whose (x, y)-entry has a value $x \odot y \pmod 2$. For any sets $A, B \subseteq \Sigma^{n/2}$, the *discrepancy* of a rectangle $A \times B$ in M is $Disc_M(A \times B) = 2^{-n}\left|\#_1^{(M)}(A \times B) - \#_0^{(M)}(A \times B)\right|$, where $\#_b^{(M)}(A \times B)$ means the total number of b ($b \in \{0, 1\}$) entries in M when M's entries are limited to $A \times B$. It is known that, for any pair $A, B \subseteq \Sigma^{n/2}$, $Disc_M(A \times B) \leq 2^{-3n/4}\sqrt{|A||B|}$ (see, e.g., [3, Example 12.14]). This implies $Disc_M(A \times B) \leq 2^{-n/4}$. Although it is not quite tight, this loose bound still serves well for our purpose.

For each index $i \in [1, m]_{\mathbb{Z}}$, we define two sets $A_i = \{u \in \Sigma^{n/2} \mid \exists v \in \Sigma^{n/2}[u^Rv \in S_i^{(n)}]\}$ and $B_i = \{v \in \Sigma^{n/2} \mid \exists u \in \Sigma^{n/2}[uv \in S_i^{(n)}]\}$, and we claim the following equation.

Claim 9. *For each bit b, $\#_b^{(M)}(A_i \times B_i) = |U_b^{(i)}|$.*

It is clear from this claim that $2^{-n}||U_1^{(i)}| - |U_0^{(i)}|| = Disc_M(A_i \times B_i) \leq 2^{-n/4}$. This inequality leads to the desired bound $||U_1^{(i)}| - |U_0^{(i)}|| \leq 2^{3n/4}$ stated in Claim 8.

To end our proof, we shall prove Claim 9. Let us consider the case $b = 0$. The other case is similar and omitted here. First, let N be another $\Sigma^{n/2}$-by-$\Sigma^{n/2}$ matrix in which the value of each (x, y)-entry is $x^R \odot y \pmod 2$. Obviously, we have $\#_0^{(M)}(A_i \times B_i) = \#_0^{(N)}(A_i^R \times B_i)$, where $A_i^R = \{w^R \mid w \in A_i\}$. Second, we show that $A_i^R \times B_i = S_i^{(n)}$ by identifying (u, v) with uv whenever $|u| = |v|$. This is shown as follows. Assume that $uv \in S_i^{(n)}$. By the definitions of A_i and B_i, it follows that $u^R \in A_i$ and $v \in B_i$; hence, $(u, v) \in A_i^R \times B_i$. Conversely, assume that $(u, v) \in A_i^R \times B_i$. Take two strings $\hat{u}, \hat{v} \in \Sigma^{n/2}$ for which $u\hat{v} \in S_i^{(n)}$ and $\hat{u}v \in S_i^{(n)}$. The *swapping property* of $S_i^{(n)}$ given in Lemma 5.5 implies that $uv \in S_i^{(n)}$. Therefore, it holds that $A_i^R \times B_i = S_i^{(n)}$. The above two equations imply that $\#_0^{(M)}(A_i \times B_i) = \#_0^{(N)}(A_i^R \times B_i) = |S_i^{(n)} \cap \overline{IP}_*| = |U_0^{(n)}|$. From this equation follows Claim 9. □

To close this section, we shall consider "closeness" of two languages and exhibit a closure property of the family of $\mathcal{C}$-pseudorandom languages under this closeness property. Two languages A and B over the same alphabet Σ are said to be *almost equal* if the function $\delta(n) = \frac{dense(A \triangle B)(n)}{|\Sigma^n|}$ is negligible. Note that this binary relation is actually an equivalence relation (satisfying reflexivity, symmetry, and transitivity).

Lemma 5.6. *Let $\mathcal{C}$ be any language family and let A and B be any two languages over an alphabet Σ. If A and B are almost equal and A is $\mathcal{C}$-pseudorandom, then B is also $\mathcal{C}$-pseudorandom.*

Proof. Let A and B be any two languages over an alphabet Σ. We assume that A is $\mathcal{C}$-pseudorandom and that A and B are almost equal. To show the $\mathcal{C}$-pseudorandomness of B, let p be any positive polynomial and let n be any number, which is sufficiently large to withstand our argument that proceeds in the rest of this proof.

Let C be an arbitrary language in $\mathcal{C}$. To achieve our goal, it suffices to show that $\left|\frac{|B_n \triangle C_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq 1/p(n)$. The $\mathcal{C}$-pseudorandom of A indicates that $\left|\frac{|A_n \triangle C_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq 1/p(n)$. Moreover, since A and B are almost equal, we have $\frac{|A_n \triangle B_n|}{|\Sigma^n|} \leq 1/4p(n)$. It is not difficult to show that $\overline{A}$ and $\overline{B}$ are also almost equal; thus, it also follows that $\frac{|\overline{A}_n \triangle \overline{B}_n|}{|\Sigma^n|} \leq 1/4p(n)$.

We can bound the value $||B_n \triangle C_n| - |A_n \triangle C_n||$ from above by the sum of $\left||B_n \cap \overline{C}_n| - |A_n \cap \overline{C}_n|\right|$ and $\left||\overline{B}_n \cap C_n| - |\overline{A}_n \cap C_n|\right|$. Note that the term $\left||B_n \cap \overline{C}_n| - |A_n \cap \overline{C}_n|\right|$ is at most $|A_n \cap \overline{B}_n| + |\overline{A}_n \cap B_n|$, which clearly equals $|A_n \triangle B_n|$. A similar bound is given for $\left||\overline{B}_n \cap C_n| - |\overline{A}_n \cap C_n|\right|$. Combining these two bounds leads to

$$\frac{||B_n \triangle C_n| - |A_n \triangle C_n||}{|\Sigma^n|} \leq \frac{|A_n \triangle B_n|}{|\Sigma^n|} + \frac{|\overline{A}_n \triangle \overline{B}_n|}{|\Sigma^n|} \leq \frac{1}{4p(n)} + \frac{1}{4p(n)} = \frac{1}{2p(n)}.$$

From this bound, we obtain

$$\left|\frac{|B_n \triangle C_n|}{|\Sigma^n|} - \frac{1}{2}\right| \leq \left|\frac{|A_n \triangle C_n|}{|\Sigma^n|} - \frac{1}{2}\right| + \left|\frac{|B_n \triangle C_n| - |A_n \triangle C_n|}{|\Sigma^n|}\right| \leq \frac{1}{2p(n)} + \frac{1}{2p(n)} = \frac{1}{p(n)}.$$

Since C is arbitrary, we conclude the $\mathcal{C}$-pseudorandomness of B, as requested. □

6. Pseudorandom generators

Rather than *determining* the pseudorandomness of strings, we intend to *produce* pseudorandom strings. A function that generates such strings, known as a *pseudorandom generator*, is an important cryptographic primitive, and a large volume of work has been dedicated to its theoretical and practical applications. In accordance with this paper's main theme of formal language theory, we define our pseudorandom generator so that it fools "languages" rather than "probabilistic algorithms" as in its conventional definition (found in, e.g., [14]). A similar treatment appears in, for instance, designing of generators that fool "Boolean circuits." For ease of notation, we always denote the binary alphabet $\{0, 1\}$ by Σ. Let us recall the notation χ_A, which expresses the characteristic function of A. In cryptography, we often limit our interest within a function G that maps Σ^* to Σ^* with a *stretch factor*[4] $s(n)$; namely, $|G(x)| = s(|x|)$ holds for all strings $x \in \Sigma^*$. Such a function G is said to *fool* a language A over Σ if the function $\ell(n) =_{def} \left|\mathrm{Prob}_x[\chi_A(G(x)) = 1] - \mathrm{Prob}_y[\chi_A(y) = 1]\right|$ is negligible, where x and y are random variables over Σ^n and $\Sigma^{s(n)}$, respectively. We often call an input x fed to G a *seed*. A function G is called a *pseudorandom generator* against a language family $\mathcal{C}$ if G fools every language A over Σ in $\mathcal{C}$. Taking the significance of p-denseness into our consideration, we also introduce a weaker form of pseudorandom generator, which fools only p-dense languages. Formally, a *weakly pseudorandom generator* against $\mathcal{C}$ is a function that fools every p-dense language over Σ in $\mathcal{C}$. Obviously, every pseudorandom generator is a weakly pseudorandom generator. As shown below, the $\mathcal{C}$-pseudorandomness discussed in the previous section has a close connection to pseudorandom generators against $\mathcal{C}$.

In particular, this paper draws our attention to "almost one-to-one" pseudorandom generators. A generator G with the stretch factor $n+1$ is called *almost* 1–1 if there is a negligible function $\tau(n) \geq 0$ such that $|\{G(x) \mid x \in \Sigma^n\}| = |\Sigma^n|(1-\tau(n))$ for all numbers $n \in \mathbb{N}$.

[4] This factor is also called an *expansion factor* in, e.g., [14].

Recall from Section 2 the single-valued total function class $\mathrm{CFLSV_t}$, which includes 1-FLIN as a *proper* subclass (because 1-FLIN $=$ $\mathrm{CFLSV_t}$ would imply REG $=$ CFL). Hereafter, we shall aim at proving that $\mathrm{CFLSV_t}$ contains an almost 1–1 pseudorandom generator against REG/n.

Proposition 6.1. *There exists an almost* 1–1 *pseudorandom generator in* $\mathrm{CFLSV_t}$ *against* REG/n.

To prove this proposition, let us discuss an intimate relationship between two notions: $\mathcal{C}$-pseudorandomness and pseudorandom generators against $\mathcal{C}$. Our key lemma below states that any almost 1–1 (weakly) pseudorandom generator against $\mathcal{C}$ can be characterized by the notion of (weakly) $\mathcal{C}$-pseudorandomness.

Lemma 6.2. *Let $\Sigma = \{0, 1\}$. Let $\mathcal{C}$ be any language family containing the language Σ^*. Let G be any almost* 1–1 *function from Σ^* to Σ^* with the stretch factor $n + 1$.*

1. *G is a pseudorandom generator against $\mathcal{C}$ iff the range $S = \{G(x) \mid x \in \Sigma^*\}$ of G is an $\mathcal{C}$-pseudorandom set.*
2. *G is a weakly pseudorandom generator against $\mathcal{C}$ iff the range $S = \{G(x) \mid x \in \Sigma^*\}$ of G is a weakly $\mathcal{C}$-pseudorandom set.*

Proof. Let $\mathcal{C}$ be any language family with $\Sigma^* \in \mathcal{C}$. Assume that G is an almost 1–1 function stretching n-bit seeds to $(n + 1)$-bit strings. Consider G's range $S = \{G(x) \mid x \in \Sigma^*\}$. For any language B over Σ and for each length $n \in \mathbb{N}$, B_{n+1} denotes $B \cap \Sigma^{n+1}$ and $\overline{B}_{n+1}$ denotes $\overline{B} \cap \Sigma^{n+1}$. In particular, S_{n+1} equals $\{G(x) \mid x \in \Sigma^n\}$. Since G is almost 1–1, it holds that $|S_{n+1}| = |\Sigma^n|(1 - \tau(n))$ for a certain negligible function $\tau(n) \geq 0$. In other words, $|\Sigma^n| - |S_{n+1}| = |\Sigma^n|\tau(n)$. We write $\ell_B(n)$ for $\left|\mathrm{Prob}_{x\in\Sigma^n}[\chi_B(G(x)) = 1] - \mathrm{Prob}_{y\in\Sigma^{n+1}}[\chi_B(y) = 1]\right|$. In addition, let $\ell''_B(n) = \frac{||S_{n+1}\cap B_{n+1}| - |\overline{S}_{n+1}\cap B_{n+1}||}{|\Sigma^{n+1}|}$, which equals $\left|\frac{|S_{n+1}\cap B_{n+1}|}{|\Sigma^n|} - \frac{|B_{n+1}|}{|\Sigma^{n+1}|}\right|$ since $|B_{n+1}| = |S_{n+1} \cap B_{n+1}| + |\overline{S}_{n+1} \cap B_{n+1}|$. Henceforth, we want to show only Statement (1) since Statement (2) can be proven similarly.

(Only If-part) Assume that G is a pseudorandom generator against $\mathcal{C}$. Let B be any language in $\mathcal{C}$. Since G fools B, the function $\ell_B(n)$ should be negligible. Take any positive polynomial p. Assume that n is sufficiently large so that $\ell_B(n) \leq 1/2p(n)$ and $\tau(n) \leq 1/2p(n)$. It thus follows that $|\Sigma^n| - |S_{n+1}| \leq |\Sigma^n|/2p(n)$. We set δ_n and ϵ_n to satisfy that $\sum_{y\in S_{n+1}\cap B_{n+1}} |G^{-1}(y)| = \delta_n |S_{n+1} \cap B_{n+1}|$ and $\sum_{y\in S_{n+1}\cap \overline{B}_{n+1}} |G^{-1}(y)| = \epsilon_n \left|S_{n+1} \cap \overline{B}_{n+1}\right|$. Obviously, $\delta_n, \epsilon_n \geq 1$. Note that $\sum_{y\in S_{n+1}} |G^{-1}(y)|$ equals the sum $\sum_{y\in S_{n+1}\cap B_{n+1}} |G^{-1}(y)| + \sum_{y\in S_{n+1}\cap \overline{B}_{n+1}} |G^{-1}(y)|$. Since $|\Sigma^n| = \sum_{y\in S_{n+1}} |G^{-1}(y)|$, we then obtain $|\Sigma^n| = \delta_n|S_{n+1} \cap B_{n+1}| + \epsilon_n|S_{n+1} \cap \overline{B}_{n+1}|$. From this relation, it follows that, since $\epsilon_n, \delta_n \geq 1$,

$$\left|\Sigma^n\right| - |S_{n+1}| = (\delta_n - 1)\,|S_{n+1} \cap B_{n+1}| + (\epsilon_n - 1)\left|S_{n+1} \cap \overline{B}_{n+1}\right|. \tag{4}$$

Therefore, it holds that $(\delta_n - 1)|S_{n+1} \cap B_{n+1}| \leq |\Sigma^n| - |S_{n+1}| \leq |\Sigma^n|/2p(n)$.

Next, we want to estimate the value $\ell''_B(n)$. We need to show that $\ell''_B(n) \leq 1/p(n)$, because a "pseudorandom" version of Lemma 5.1(2–3) therefore leads to the $\mathcal{C}$-pseudorandomness of S. We first note that

$$\mathrm{Prob}_{x\in\Sigma^n}[\chi_B(G(x)) = 1] = \frac{\sum_{y\in S_{n+1}\cap B_{n+1}} |G^{-1}(y)|}{|\Sigma^n|} = \frac{\delta_n\,|S_{n+1} \cap B_{n+1}|}{|\Sigma^n|}.$$

Since $\mathrm{Prob}_{y\in\Sigma^{n+1}}[\chi_B(y) = 1] = |B_{n+1}|/|\Sigma^{n+1}|$, $\ell_B(n)$ thus equals $\left|\frac{|B_{n+1}|}{|\Sigma^{n+1}|} - \frac{\delta_n|S_{n+1}\cap B_{n+1}|}{|\Sigma^n|}\right|$. As a result, we can bound the value $\ell''_B(n)$ as

$$\ell''_B(n) \leq \left|\frac{|B_{n+1}|}{|\Sigma^{n+1}|} - \frac{\delta_n|S_{n+1} \cap B_{n+1}|}{|\Sigma^n|}\right| + \frac{(\delta_n - 1)|S_{n+1} \cap b_{n+1}|}{|\Sigma^n|} \leq \ell(n) + \frac{1}{2p(n)}.$$

From our assumption $\ell_B(n) \leq 1/2p(n)$, we then conclude that $\ell''_B(n) \leq \ell_B(n) + \frac{1}{2p(n)} \leq \frac{1}{p(n)}$.

(If-part) Assume that the set $S = \{G(x) \mid x \in \Sigma^*\}$ is $\mathcal{C}$-pseudorandom. To show that G is a pseudorandom generator against $\mathcal{C}$, we want to show that the function $\ell_B(n)$ is negligible for any language B in $\mathcal{C}$. Let p be any positive polynomial and let B be any language in $\mathcal{C}$. Since S is $\mathcal{C}$-pseudorandom, by a "pseudorandom" version of Lemma 5.1(2–3), $\ell''_B(n)$ is upper-bounded by $1/2p(n)$ for all but finitely many numbers n.

Now, choose a number δ_n so that $\mathrm{Prob}_{x\in\Sigma^n}[\chi_B(G(x)) = 1] = \delta_n|S_{n+1} \cap B_{n+1}|/|\Sigma^n|$. By Eq. (4), we obtain $(\delta_n - 1)|S_{n+1} \cap B_{n+1}| \leq |\Sigma^n| - |S_{n+1}| \leq |\Sigma^n|/2p(n)$. As stated before, it holds that $\ell_B(n) = \left|\frac{\delta_n|S_{n+1}\cap B_{n+1}|}{|\Sigma^n|} - \frac{|B_{n+1}|}{|\Sigma^{n+1}|}\right|$. Since $\delta_n \geq 1$, we obtain

$$\ell_B(n) \leq \frac{(\delta_n - 1)|S_{n+1} \cap B_{n+1}|}{|\Sigma^n|} + \left|\frac{|S_{n+1} \cap B_{n+1}|}{|\Sigma^n|} - \frac{|B_{n+1}|}{|\Sigma^{n+1}|}\right| \leq \frac{1}{2p(n)} + \ell''_B(n).$$

Therefore, since $\ell''_B(n) \leq 1/2p(n)$, the inequality $\ell_B(n) \leq 1/p(n)$ follows. From the arbitrariness of B in $\mathcal{C}$, we can conclude that G is a pseudorandom generator against $\mathcal{C}$. □

In what follows, we shall describe the proof of Proposition 6.1. Let us recall the context-free language IP_* given in Section 5. We want to build our desired pseudorandom generator based on the REG/n-pseudorandomness of IP_*.

Proof of Proposition 6.1. The desired generator G is defined as follows. Let n be an arbitrary number at least 3 and let $w = axy$ be any input of length n satisfying that $a \in \{\lambda, 0, 1\}$ and $|x| = |y| + 1$. We first consider the case where n is odd

(i.e., $a = \lambda$), assuming further that $x = bz$ for a certain bit b. Since n is odd, let $k = (n-1)/2$. As described below, our generator G outputs a string of the form $x'y'e$ of length $n+1$, where $|x'| = |x|$, $|y'| = |y|$, and $e \in \{0, 1\}$.

(1) If $w = bzy$ for a certain bit b and $z^R \odot y \equiv 1 \pmod 2$, then let $G(w) = bzy\bar{b}$.
(2) If $w = 1zy$ and $z^R \odot y \equiv 0 \pmod 2$, then let $G(w) = 1zy1$.
(3) If $w = 0zy$ and $z^R \odot y \equiv 0 \pmod 2$, then check if there is the maximal index i such that $z_{k-i+1} = 1$.
  (3a) When such i exists, let $G(w) = 0z\tilde{y}0$, where $\tilde{y}$ is obtained from y by flipping only the ith bit; that is, $\tilde{y} = y_1y_2\cdots y_{i-1}\bar{y}_iy_{i+1}\cdots y_k$.
  (3b) Consider the other case where i does not exist; in other words, $z = 0^k$. In this case, we define $G(w) = 1zy1$.

In the remaining case where n is even (i.e., $a \in \{0, 1\}$), we define $G(w)$ to be $aG(xy)$.

Our next goal is to show that G is a pseudorandom generator in $\mathrm{CFLSV_t}$ against REG/n. We start with the following claim.

Claim 10. *The function G is almost* 1–1.

Proof. When n is odd, we set $k = (n-1)/2$ as before. In the above definition of G, it is obvious that all the cases except Case (3b) make G one-to-one. It is thus sufficient to deal with Case (3b). In this case, for each fixed string $y \in \Sigma^k$, only inputs taken from the set $\{00^ky, 10^ky\}$ are mapped by G into the same string 10^ky1. Now, we define $\tau(n) = 1/2^{k+1}$. Letting A_k denote $\bigcup_{y\in\Sigma^k}\{00^ky, 10^ky\}$, we note that G is one-to-one on the domain $\Sigma^n - A_k$ and 2-to-1 on the domain A_k. Since $|A_k| = 2^{k+1}$, it thus follows that $|\{G(w) \mid w \in \Sigma^n\}| = |\Sigma^n - A_k| + \frac{|A_k|}{2} = |\Sigma^n| - \frac{|A_k|}{2}$, which equals $|\Sigma^n|\left(1 - 2^{-(n+1)/2}\right) = |\Sigma^n|(1-\tau(n))$. The other case where n is even follows from the previous case and we can define τ accordingly. Clearly, τ is negligible, and therefore G is almost 1–1. □

Claim 11. *The range $S = \{G(w) \mid w \in \Sigma^*\}$ of G coincides with IP_*.*

Proof. The containment $S \subseteq IP_*$ can be shown as follows. Letting $w \in \Sigma^n$ be any input string, we want to show that $G(w) \in IP_*$. Now, assume that n is odd, and consider Case (1) with $w = bzy$ and $z^R \odot y \equiv 1 \pmod 2$. In this case, $G(w) = bzy\bar{b}$. Since $(bz)^R \odot (y\bar{b}) \equiv z^R \odot y + b \odot \bar{b} \equiv 1 \pmod 2$, it follows that $G(w) \in IP_*$. Next, we consider Case (3a) with $w = 0zy$ and $z^R \odot y \equiv 0 \pmod 2$. Let $j = \max\{i \mid z_{k-i+1} = 1\}$. Notice that $z_{k-j+1} \odot y_j \not\equiv z_{k-j+1} \odot \bar{y}_j \pmod 2$ because $z_{k-j+1} = 1$. Thus, it follows that

$$z^R \odot y = \sum_{i:i\neq j} z_{k-i+1} \odot y_i + z_{k-j+1} \odot y_j \not\equiv \sum_{i:i\neq j} z_{k-i+1} \odot y_i + z_{k-j+1} \odot \bar{y}_j = z^R \odot \tilde{y}.$$

As a result, we obtain $z^R \odot \tilde{y} \equiv 1 \pmod 2$, which obviously implies that $G(w) \in IP_*$. The other cases are similarly shown.

We then show the other containment $IP_* \subseteq S$. Choose an arbitrary string $u \in IP_* \cap \Sigma^n$ and assume that n is even. Let $k = (n-2)/2$. Consider the case where $u = bzy\bar{b}$ with $b \in \{0, 1\}$ and $|z| = |y| = k$. Since $u \in IP_*$, we have $(bz)^R \odot (y\bar{b}) \equiv z^R \odot y \equiv 1 \pmod 2$. Hence, G should map bzy to u. This means that u is in S. Next, we consider the case where $u = 0zy0$ with $|z| = |y|$. Let $j = \max\{i \mid z_{k-i+1} = 1\}$. As before, we define $\tilde{y}$ from y by flipping the jth bit of y. Since $G(0z\tilde{y})$ equals $0zy0$, it follows that $u \in S$. The other cases are similarly proven. □

Since IP_* is REG/n-pseudorandom, by Claim 11, S is also REG/n-pseudorandom. From G's almost one-oneness and its stretch factor of $n+1$, Lemma 6.2(1) guarantees that G is a pseudorandom generator against REG/n. What remains unproven is that G actually belongs to $\mathrm{CFLSV_t}$.

Claim 12. *G is in* $\mathrm{CFLSV_t}$.

Proof. Here, we give an npda with a write-only output tape, which computes G. Our npda N works as follows. On input w of the form axy, guess nondeterministically whether $a = \lambda$ or not. Along a nondeterministic branch associated with a guess "$a = \lambda$," check nondeterministically whether $|x| = |y| + 1$ using a stack as storage space. During this checking process, N also computes $z^R \odot y$, where $x = bz$, and finds the maximal index i_0 such that $z_{k-i_0+1} = 1$ (if any). While reading input bits, for each nondeterministic computation, N produces three types of additional computation paths. Along the first one of such paths, N writes 10^ky1 on its output tape; on the second path, N writes bxy on the output tape; on the third path, N writes $0z\tilde{y}0$, provided that i_0 exists. At the end of scanning the input, if Case (3b) does not hold, N enters a rejecting state on the first path to invalidate its output 10^ky1. If Case (3a) does not hold, N also invalidate its output $0z\tilde{y}0$ on the third path. In Cases (1)–(2), assume that N has written bxy on the second path. Now, N writes down $\bar{b}$ or 1, respectively, on the output tape following bxy if Case (1) or Case (2) holds. It is not difficult to show that, for each input string w, N's valid output is unique and it matches $G(w)$. This npda N therefore places G into $\mathrm{CFLSV_t}$. □

To this end, we have already completed our proof of Proposition 6.1. □

We shall close this section by demonstrating another application of Lemma 6.2 to the non-existence of a weakly pseudorandom generator in 1-FLIN.

Proposition 6.3. *There is no almost* 1–1 *weakly pseudorandom generator in* 1-FLIN *with the stretch factor $n+1$ against* REG.

Our proof of this proposition demands new terminology. For any two multi-valued partial functions f and g mapping Σ^* to Γ^*, where Γ could be another alphabet, f is called a *refinement* of g if, for any string $x \in \Sigma^*$, (i) $f(x) \subseteq g(x)$ (set inclusion) and (ii) $f(x) = \emptyset$ implies $g(x) = \emptyset$. Concerning 1-NLINMV, Tadaki et al. [29] proved that every length-preserving function in 1-NLINMV has a refinement in 1-FLIN(partial).

Here, we present the proof of Proposition 6.3.

Proof of Proposition 6.3. Let G be any almost 1–1 weakly pseudorandom generator against REG stretching n-bit seeds to $(n+1)$-bit long strings. Toward a contradiction, we assume that G belongs to 1-FLIN. By Lemma 6.2(2), the range $S = \{G(x) \mid x \in \Sigma^*\}$ is weakly REG-pseudorandom. If S is regular, then REG is weakly REG-pseudorandom; however, this contradicts the *self-exclusion property*: REG cannot be weakly REG-pseudorandom. To obtain this contradiction, it remains to prove that S is a regular language.

To make G length-preserving, we slightly expand G and define $\hat{G}(xb) = G(x)$ for each string x and each bit b. This new function $\hat{G}$ is also in 1-FLIN. Let us consider its inverse function $\hat{G}^{-1}(y) = \{x \mid \hat{G}(x) = y\}$. Obviously, the inverse function $\hat{G}^{-1}$ belongs to 1-NLINMV (by guessing x and then checking whether $\hat{G}(x) = y$). Note that $S = \{y \mid \hat{G}^{-1}(y) \neq \emptyset\}$. Since every length-preserving function in 1-NLINMV has a refinement in 1-FLIN(partial) [29], there exists a refinement $f \in$ 1-FLIN(partial) of $\hat{G}^{-1}$, and we denote by N a linear-time deterministic 1TM that computes f.

Claim 13. *For every string y, $y \in S$ iff N on the input y terminates with an accepting state.*

As a consequence of Claim 13, S belongs to 1-DTIME($O(n)$), which equals REG [16]. We thus obtain the regularity of S, as we have planned.

Finally, we want to prove Claim 13. Assume that y is in S; namely, $\hat{G}^{-1}(y) \neq \emptyset$. Since f is a refinement of $\hat{G}^{-1}$, we have $f(y) \neq \emptyset$, which indicates that N terminates with an accepting state. Conversely, assume that N on y terminates with an accepting state. In other words, $f(y) \neq \emptyset$. Since $f(y) \subseteq \hat{G}^{-1}(y)$, we obtain $\hat{G}^{-1}(y) \neq \emptyset$. This implies that $y \in S$. Therefore, Claim 13 holds. □

7. Discussion and open problems

We have discussed two notions—immunity and pseudorandomness—in a framework of formal language theory. For these notions, our main target of this paper is CFL, the family of context-free languages. Our initial study has revealed a quite rich structure that lies inside CFL. For instance, CFL contains complex languages, which are REG-immune, CFL-simple, and REG/n-pseudorandom. Moreover, its function class $\mathrm{CFLSV}_{\mathrm{t}}$ contains a pseudorandom generator against REG/n. Despite much efforts, however, there remain several key questions that we have not answered throughout this paper. To direct future research, we generate a short list of those questions for the interested reader.

1. Prove or disprove that CFL(2) − CFL/n is CFL-immune.
2. Is there any context-free language that is p-dense REG-immune? Is one of such languages located outside of REG/n?
3. As noted in Section 3, the language L_{3eq} belongs to CFL(2) and it is also CFL(1)-immune. In short, CFL(2) is CFL(1)-immune. Naturally, we can ask if, for each index $k \geq 2$, CFL($k+1$) is CFL(k)-immune.
4. The languages $\overline{L_{keq}}$, where $k \geq 3$, are shown to be CFL-simple; however, they are not REG-immune. Is there any REG-immune CFL-simple language?
5. As shown in Section 3.3, L ∩ REG/n is REG-bi-immune. Determine whether CFL is also REG-bi-immune. More strongly, is CFL − REG/n REG-bi-immune?
6. We can define the notion of "CFL-primesimplicity" analogous to "CFL-simplicity." Find natural CFL-primesimple languages.
7. Is DCFL weakly REG/n-pseudorandom? An affirmative answer implies the REG/n-bi-primeimmunity of DCFL by Lemma 5.2.
8. Our pseudorandom generator G given in Section 6 is *almost* 1–1 instead of 1–1. Find a "natural" 1–1 pseudorandom generator against REG/n.
9. Find a natural and easy-to-compute pseudorandom generator against CFL/n.

Satisfactory answers to the above questions will guide us to a more thorough analysis of structural properties of the context-free languages and therefore enrich our knowledge on CFL.

References

[1] K. Ambos-Spies, H. Fleischhack, H. Huwig, Diagonalizations over deterministic polynomial time, in: Proc. of the 1st Workshop on Computer Science Logic, CSL'87, in: Lecture Notes in Computer Science, vol. 329, Springer, 1988, pp. 1–16.
[2] K. Ambos-Spies, E. Mayordomo, Y. Wang, X. Zheng, Resource-bounded dense genericity, stochasticity and weak randomness, in: Proc. 13th International Symposium on Theoretical Aspects of Computer Science, in: Lecture Notes in Computer Science, vol. 1046, Springer, 1996, pp. 63–74.
[3] S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009.
[4] J.L. Balcázar, Separating, strongly separating, and collapsing relativized complexity classes, in: Proc. 11th Symposium on Mathematical Foundations of Computer Science, MFCS'84, in: Lecture Notes in Computer Science, vol. 176, Springer, 1984, pp. 1–16.
[5] J.L. Balcázar, J. Díaz, J. Gabarró, Structural Complexity I & II, Springer-Verlag, 1988(I)&1990(II).
[6] J.L. Balcázar, U. Schöning, Bi-immune sets for complexity classes, Math. Syst. Theory 18 (1985) 1–10.
[7] Y. Bar-Hillel, M. Perles, E. Shamir, On formal properties of simple phrase-structure grammars, Z. Phonetik, Sprachwiss. Kommunikationsforsch. 14 (1961) 143–172.
[8] M. Blum, S. Micali, How to generate cryptographically strong sequences of pseudorandom bits, SIAM J. Comput. 13 (1984) 850–864.
[9] N. Chomsky, Three models for the description of language, IEEE Trans. Inform. Theory 2 (3) (1956) 113–124.
[10] N. Chomsky, On certain formal properties of grammars, Inform. Control 2 (1959) 137–167.

[11] A. Church, On the concept of a random sequence, Bull. Amer. Math. Soc. 45 (1940) 130–135.
[12] P. Flajolet, J.M. Steyaert, On sets having only hard subsets, in: Proc. 2nd International Colloquium on Automata, Languages, and Programming, in: Lecture Notes in Computer Science, vol. 14, Springer, 1974, pp. 446–457.
[13] S. Ginsburg, G.F. Rose, Some recursively unsolvable problems in ALGOL-like languages, J. ACM 10 (1963) 29–47.
[14] O. Goldreich, Foundations of Cryptography: Basic Tools, Cambridge University Press, 2001.
[15] S.A. Greibach, A new normal-form theorem for context-free phrase structure grammars, J. ACM 12 (1965) 42–52.
[16] F.C. Hennie, One-tape, off-line Turing machine computations, Inform. Control 8 (1965) 553–578.
[17] S. Homer, W. Maass, Oracle-dependent properties of the lattice of NP sets, Theoret. Comput. Sci. 24 (1983) 279–289.
[18] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.
[19] J.E. Hopcroft, R. Motwami, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd ed., Addison-Wesley, 2001.
[20] K. Ko, D. Moore, Completeness, approximation and density, SIAM J. Comput. 10 (1981) 787–796.
[21] K. Kobayashi, On the structure of one-tape nondeterministic Turing machine time hierarchy, Theoret. Comput. Sci. 40 (1985) 175–193.
[22] G. Lischke, Towards the actual relationship between NP and exponential time, Math. Log. Q. 45 (1999) 31–49.
[23] D. Loveland, A new interpretation of the von Mises concept of random sequence, Z. Math. Log. Grundl. Math 12 (1966) 277–294.
[24] A.R. Meyer, E.M. McCreight, Computationally complex and pseudo-random zero-one valued functions, in: Z. Kohavi, A. Paz (Eds.), Theory of Machines and Computations, Academic Press, 1971, pp. 19–43.
[25] H. Müller, A note on balanced immunity, Math. Syst. Theory 26 (1993) 157–167.
[26] E.L. Post, Recursively enumerable sets of positive integers and their decision problems, Bull. Amer. Math. Soc. 50 (1944) 284–316.
[27] H. Rogers, The Theory of Recursive Functions and Effective Computability, McGraw Hill, 1967.
[28] U. Schöning, R.V. Book, Immunity, relativizations, and nondeterminism, SIAM J. Comput. 13 (1984) 329–337.
[29] K. Tadaki, T. Yamakami, J.C.H. Lin, Theory of one tape linear time Turing machines, Theoret. Comput. Sci. 411 (2010) 22–43. An extended abstract appeared in the Proc. of the 30th SOFSEM Conference on Current Trends in Theory and Practice of Computer Science, Lecture Notes in Computer Science, Springer, vol. 2932, 2004, pp. 335–348.
[30] R.E. Wilber, Randomness and the density of hard problems, in: Proc. 24th Annual Symposium on Foundations of Computer Science, 1983, pp. 335–342.
[31] T. Yamakami, Swapping lemmas for regular and context-free languages with advice. Available at: arXiv:0808.4122, 2008.
[32] T. Yamakami, T. Suzuki, Resource bounded immunity and simplicity, Theoret. Comput. Sci. 347 (2005) 90–129. An extended abstract appeared in the Proc. of the 3rd IFIP International Conference on Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics, Kluwer Academic Publishers, 2004, pp. 81–95.
[33] A.C. Yao, Theory and application of trapdoor functions, in: Proc. of the 23rd IEEE Symposium on Foundations of Computer Science, FOCS'82, 1982, pp. 80–91.