The Formal Language Theory Column

by

Arto Salomaa

Turku Centre for Computer Science
University of Turku
Lemminkäisenkatu 14, 20520 Turku, Finland
asalomaa@it.utu.fi

Enumeration of Formal Languages

Michael Domaratzki
Jodrey School of Computer Science
Acadia University
Wolfville, NS B4P 2R6 Canada
mike.domaratzki@acadiau.ca

Abstract

We survey recent results on the enumeration of formal languages. In particular, we consider enumeration of regular languages accepted by deterministic and nondeterministic finite automata with n states, regular languages generated by regular expressions of a fixed length, and ω-regular languages accepted by Müller automata. We also survey the uncomputability of enumeration of context-free languages and more general structures.

1 Introduction

Given a set of objects, enumeration asks "how many distinct objects are there?" Easy examples of enumeration problems are "how many binary sequences of

length n are there?" (2^n) and "how many distinct subsets of size m can we take from a set of n elements?" ($\binom{n}{m}$). A sampling of other classical topics for enumeration familiar to computer scientists include graphs ("how many non-isomorphic graphs on n vertices are there?"), trees, primitive and Lyndon words [33, A000031], and monotone Boolean functions (this enumeration problem is known as Dedekind's problem [33, A000372]). With over 100 000 sequences, the Encyclopedia of Integer Sequences [33] contains a wealth of examples related to enumeration.

The enumeration of structures in formal language theory is a topic that has been considered for almost fifty years, and the objects in formal language theory yield many interesting enumeration problems. In this survey, we consider recent results on the enumeration of formal languages. Many of these results concern enumeration of finite automata, but we also consider enumeration of regular expressions, context-free languages, and more general results.

Why enumeration? There are several compelling reasons for studying the enumeration of formal languages beyond the intrinsic research challenge. In particular, research on enumeration is closely linked to problems of random generation of automata [3], average case complexity [27] and establishing lower bounds by counting arguments (see, e.g., Domaratzki *et al.* [11, Thm. 2.5] for an example from formal language theory). Results on enumeration are useful in varied locations when, for one reason or another, the number of regular languages of a given size is required. A recent example is given Gramlich and Schnitger [14], who use bounds on the number of regular languages accepted by NFAs with n states in proving inapproximability results for finding minimal NFAs for a given DFA.

2 Enumeration and Formal Language Theory

Given an infinite set of objects $\mathcal{S}$, enumeration asks the question "how many distinct objects are there in $\mathcal{S}$ of size n"? The goal of enumeration is to express this quantity exactly as a function of n, typically in a closed form. Asymptotics for these functions are typically also of interest to researchers, for comparative purposes.

There are some important assumptions in enumeration problems. The two most crucial—especially in relation to formal language enumeration—are the *measurement* and the idea of *equivalence*. First, we must have a measure on $\mathcal{S}$ such that the number of objects of size n is finite for all n. Clearly, without a measure satisfying these requirements, asking enumeration questions doesn't make sense. When enumerating structures in formal language theory, there are often several different descriptional complexity measures available. This gives many, often unique research questions for the same structure, as we see in this survey.

Secondly, our enumeration problem asks about the number of *distinct* objects of size n. Thus, we must have a concept of which objects are and are not equivalent.

Typically, in classical areas like graph theory, the notion of equivalence means *isomorphic*: two directed graphs are equivalent if they are isomorphic. In formal language theory, we still use the notion of isomorphic—when discussing the uniqueness of minimal DFAs, for instance. However, when dealing with devices which generate or accept languages, our primary notion of equivalence is usually equality of the languages they generate or accept. Thus, we focus on this concept of equivalence in this survey: two language devices are equivalent if they accept or generate the same language.

3 Preliminaries

We assume the reader is familiar with the basic notions of formal language theory, in particular, the concepts of deterministic and nondeterministic finite automata (DFAs and NFAs), regular expressions, regular languages, context-free grammars (CFGs) and context-free languages (CFLs). See, for example, Rozenberg and Salomaa [32] for an introduction to concepts used in this survey.

We will employ a few descriptional complexity measures of regular languages below. The (deterministic) state complexity of a regular language L is the minimal number of states in any DFA accepting L. See Yu [36, 37] for surveys of results on state complexity. The nondeterministic state complexity [17] of a regular language L is, as expected, the minimal number of states in any NFA accepting L.

4 Early Results

Since nearly the inception of the study of formal languages, there has been interest in enumeration problems relating to automata. For a list of references and background, we refer the reader to Domaratzki *et al.* [9], where it is noted that the problem was considered at least as early as 1959, and in 1960, Harary listed enumeration of automata as an unsolved problem in graph enumeration. Harrison [16] wrote "A census of finite automata" in 1965, which provided enumeration results using group-theoretic means. Many other papers also attacked the enumeration of automata, including strongly-connected, initially-connected[1] and minimal

[1]Recall that a DFA is *initially-connected* if, for each state q, there is a word w such that $\delta(q_0, w) = q$, and similarly for an NFA. Initially-connected automata are also called *accessible* in the literature.

automata. Much research was independently conducted in the Soviet Union and in the West.

Most early research focuses on enumerating automata by considering them to be distinct if they are non-isomorphic, and little attention is given to the languages accepted by the automata. Some of the work on enumeration of minimal automata does begin to address the number of languages accepted by DFAs. To see this, let $f_k(n)$ be the number of pairwise non-isomorphic minimal DFAs with n states over a k-letter alphabet and $g_k(n)$ be the number of distinct regular languages accepted by DFAs with n states over a k-letter alphabet. Then we note that $g_k(n) = \sum_{i=1}^{n} f_k(i)$ [9, Prop. 1]. Thus, in what follows, we are generally looking for bounds on $g_k(n)$, but it will be sufficient to obtain bounds on $f_k(n)$.

For research on enumeration of minimal finite automata, we mention here in particular the less well-known work of Narushima [24, 25, 26], who developed new methods, namely inclusion-exclusion properties on semi-lattices, for enumeration of minimal automata. These techniques appear to have never been exploited to enumerate formal languages (in particular, the relationship between Narushima's methods and methods for enumerating initially-connected automata does not appear to have been studied) and the inclusion-exclusion principles do not appear to have ever been employed to give any asymptotic analysis of the number of minimal automata with n states.

There is also other work on minimal automata which we do not cover in this survey. The work of Korshunov [18, 19] (a survey in Russian is also available [20]) enumerates minimal automata. However, as noted in Domaratzki *et al.* [9], the automata studied by Korshunov lack a distinguished initial state. Korshunov also studies initially-connected automata (in which an initial state is given [20, Ch. 4]), however, it does not appear that the work was broadened to study initially-connected minimal automata.

5 Enumeration by State Complexity

Renewed interest in the enumeration of formal languages can be traced to the work of Nicaud which investigated average state complexity of operations on regular languages [27]. In order to examine the average case complexity of these operations, an exact characterization of all distinct automata with n states is required. Nicaud gives such a characterization for unary regular languages and, as a by-product, also gives an asymptotic enumeration of unary regular languages. Recall that $f_k(n)$ denotes the number of pairwise non-isomorphic minimal DFAs with n states over a k-letter alphabet and $g_k(n)$ denotes the number of distinct regular languages accepted by DFAs with n states over a k-letter alphabet. The following result is due to Nicaud [27]:

Theorem 1. *The function $f_1(n)$ satisfies $f_1(n) \sim n2^{n-1}$.*

This result of Nicaud was considered by Domaratzki *et al.* [9]. In particular, the asymptotic bound on $f_1(n)$ can be further refined, and an asymptotic bound on $g_1(n)$ can also be given [9]:

Corollary 2. *The following asymptotic bound holds: $g_1(n) = 2^n(n-\alpha+O(n2^{-n/2}))$ where α is a constant with approximate value* 1.382714455402*.*

The value of α in Theorem 2 is given by a sum involving the Möbius function [9]. Domaratzki *et al.* also examine the behaviour of the function $f_k(n)$ for $k \geq 2$. These results depend on the following theorem due to Liskovets [23] and, independently, Robinson [31]. Let $C_k(n)$ be the number of initially-connected DFAs (without final states) on n states over an alphabet of size k.

Theorem 3. *Let $n, k \geq 2$. The function $C_k(n)$ satisfies the following recurrence:*

$$C_k(n) = n^{nk} - \sum_{i=1}^{n-1} \binom{n-1}{i-1} C_k(i) n^{(n-i)k}. \tag{1}$$

The asymptotics of $C_k(n)$ are are given by Robinson [31]:

$$C_k(n) = n^{kn} \gamma_k^{n(1+o(1))}, \tag{2}$$

where γ_k is a constant depending only on k, the size of the alphabet. Korshunov [20, p. 50] also gives precise results in this area. Using Theorem 3, Domaratzki *et al.* [9] give asymptotic bounds on $f_k(n)$ for $k \geq 2$:

Theorem 4. *The function $f_k(n)$ is bounded below by a function which is asymptotically $(k - o(1))n2^{n-1}n^{(k-1)n}$ and bounded above by $2^n C_k(n)/(n-1)!$.*

Thus, considering the estimates of (2), the upper and lower bounds in Theorem 4 differ by a factor of $(\gamma_k e)^n$. For $k = 2$ this is approximately 2.27^n [9].

Reis *et al.* [30] have also considered enumeration of automata and in particular, initially-connected DFAs. By proposing a canonical, compact string representation for initially-connected DFAs, Reis *et al.* give an alternate formula for $C_k(n)$ [30].

Theorem 5. *The function $C_k(n)$ satisfies the following formula:*

$$C_k(n) = \sum_{b_1=1}^{k} \sum_{b_2=1}^{2k-b_1} \sum_{b_3=1}^{3k-b_1-b_2} \cdots \sum_{b_{n-1}=1}^{k(n-1)-\sum_{\ell=1}^{n-2} b_\ell} \prod_{j=1}^{n} j^{b_j-1}. \tag{3}$$

Finally, we consider the recent work by Bassino and Nicaud [1], who also study the enumeration of non-isomorphic initially-connected DFAs. Recall that the Stirling numbers of the second kind, denoted here by $S_2(n, m)$, are defined by $S_2(0, 0) = 1$, $S_2(n, 0) = 0$ for all $n \geq 1$ and, for all $n, m \geq 1$,

$$S_2(n, m) = mS_2(n - 1, m) + S_2(n - 1, m - 1).$$

The main enumerative result of Bassino and Nicaud gives bounds on $C_k(n)$ [1]:

Theorem 6. *Let $n, k \geq 1$. The following asymptotic bound holds:*

$$C_k(n) \in \Theta\left(nS_2(kn, n)\right). \tag{4}$$

We note that Theorem 6 is obtained from exact bounds on $C_k(n)$. Bassino and Nicaud also reinterpret a result of Korshunov using Stirling numbers of the second kind. Note that Theorems 3, 5 and 6 all do not account for the choices of final states. Thus, each of these quantities can be multiplied by a factor of 2^n, as is done in the upper bound of Theorem 4.

5.1 Enumeration by Nondeterministic State Complexity

Despite the long history of enumeration of finite automata, and the central importance of nondeterminism in automata theory, there does not appear to have been any consideration of the enumeration of nondeterministic finite automata or of regular languages by their nondeterministic state complexity until very recently. Estimates of this quantity have appeared in at least one instance (in 1997 by Pomerance *et al.* [29], which we note below), but the first study of the enumeration problem appears to be by Domaratzki *et al.* [9]. Let $G_k(n)$ denote the number of distinct regular languages accepted by NFAs with n states over a k-letter alphabet. We first consider the unary case [9]:

Theorem 7. *The function $G_1(n)$ satisfies the inequality $G_1(n) \geq 2^{n+(2.295-o(1))\sqrt{\frac{n}{\log n}}}$.*

Theorem 7 is given by languages which are accepted by NFAs in Chrobak normal form [4]. A non-trivial upper bound on $G_1(n)$ is given by Pomerance *et al.* [29]:

Theorem 8. *There are $O(n/(\log n))^n$ distinct unary languages accepted by NFAs with n states.*

For larger alphabets, the following bounds are known [9]:

Theorem 9. *For $k \geq 2$, we have $n2^{(k-1)n^2} \leq G_k(n) \leq (2n - 1)2^{kn^2} + 1$.*

One fact worth noting is that the upper bound in Theorem 9 does not enforce that the NFAs are initially-connected. In fact, it can be shown that there are asymptotically 2^{kn^2} initially-connected NFAs on n states over a k-letter alphabet with a fixed initial state and no final states [9, Thm. 12]. This result is derived from analyzing the recurrence analogous to (2) for NFAs.

5.2 Enumeration of Finite Languages by State Complexity

We now turn to enumeration of finite languages. Recently, finite languages have received an increasing amount of attention. The state complexity of language operations acting on finite languages is almost as well-studied as that of regular languages (the survey of Yu [37] also covers the case where the languages are finite). Further, the relationship between the state complexity and the longest word in a finite language has been recently studied [2].

Let $f'_k(n)$ denote the number of pairwise non-isomorphic DFAs with n states over a k-letter alphabet which accept finite languages. For finite unary languages, enumeration is trivial: the number of finite unary languages accepted by a DFA with n states is exactly 2^{n-1}. For larger alphabets, the problem has been studied by Domaratzki *et al.* [9], Domaratzki [7] and Liskovets [22].

For arbitrary alphabets, a lower bound may be given by an explicit construction [9, Thm. 15]:

Theorem 10. *For $k, n \geq 2$, $f'_k(n) \geq 2^{n-2}((n-1)!)^{k-1}$.*

Domaratzki [7] gives an improved lower bound on the number of finite languages accepted by DFAs with n states over a binary alphabet. In particular, the following bound is given by explicitly constructing large sets of finite languages all accepted by DFAs with n states [7]:

Theorem 11. *For all $n \geq 5$, $f'_2(n) \geq \frac{(2n-3)!}{(n-2)!} c_1^{n-2}$ for some constant $c_1 \simeq 1.0669467$.*

An upper bound on the number of finite languages accepted by DFAs with n states over a binary alphabet is possible by giving another combinatorial interpretation to the classical Genocchi numbers. The Genocchi numbers G_{2n} for $n \geq 1$ can be defined in terms of the following generating function (see Sloane [33, A001469] for further references):

$$\frac{2t}{e^t + 1} = t + \sum_{n \geq 1} (-1)^n G_{2n} \frac{t^{2n}}{(2n)!}.$$

In particular, we have the following result [6]:

Theorem 12. *For all $n \geq 2$, $f'_2(n) \leq 2^{n-2} G_{2n}$.*

Theorem 12 can be extended to alphabets of size k using an generalization of the Genocchi numbers due to Han [15].

Enumeration of finite languages has also been considered by Liskovets [22] by enumerating acyclic unlabelled DFAs. Using two approaches previously developed, Liskovets gives an exact enumeration of unlabelled DFAs accepting finite languages.

Let $a_k(n, r)$ be the recurrence defined by

$$a_k(n, r) = \sum_{t=0}^{n-1} \binom{n}{t} (-1)^{n-t-1} (t + r)^{k(n-t)} a_k(t, r)$$

for $n, r \geq 1$ and $a_k(0, r) = 1$ for all $r \geq 0$. The recurrence $a_k(n, r)$ enumerates DFAs which are called *quasi-acyclic* by Liskovets, but is primarily an auxiliary recurrence for the following result [22]:

Theorem 13. *Let $c_k(n)$ be the function defined by $c_k(1) = 1$ and*

$$\sum_{t=1}^{n} \binom{n-1}{t-1} a_k(n - t, t + 1) \cdot c_k(t) = a_k(n, 1)$$

for $n \geq 2$. Then $c_k(n)$ gives the number of labelled, initially-connected acyclic DFAs on n states over a k-letter alphabet.

As Liskovets notes, the number of *unlabelled* initially connected acyclic DFAs is given by the quantity $c_k(n)/(n-1)!$. The above bounds can be further improved by considering only DFAs with a unique so-called pre-dead state (the pre-dead state is the state for which all of its transitions enter the dead state). Though Liskovets does not give asymptotics for $c_k(n)$, numerical evidence suggests it gives a good upper bound on the number of finite languages accepted by DFAs with at most n states.

5.3 Enumeration of Finite Languages by Nondeterministic State Complexity

For enumeration of finite languages by nondeterministic state complexity, let $G'_k(n)$ denote the number of finite languages over a k-letter alphabet with nondeterministic state complexity n. We have the following result [9].

Theorem 14. *We have $G'_1(n) = 2^n$, and for all $k \geq 2$ and $n \geq 2$,*

$$2^{(k-1)n(n-1)/2} \leq G'_k(n) \leq 2^{n-1+kn(n-1)/2}.$$

5.4 Enumeration by ⊕-State Complexity

Recently, van Zijl [35] has considered enumeration problems for ⊕-DFAs and ⊕-NFAs. A *symmetric difference NFA* (or ⊕-NFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where each component is the same as a traditional NFA. However, we extend δ to a function $\delta : Q \times \Sigma^* \to 2^Q$ as follows:

$$\begin{aligned} \delta(q, \epsilon) &= \{q\} \quad \forall q \in Q \\ \delta(q, aw) &= \bigoplus_{q' \in \delta(q,a)} \delta(q', w) \quad \forall q \in Q, w \in \Sigma^*, a \in \Sigma. \end{aligned}$$

Here, $\oplus$ is the symmetric difference operation on sets: $X_1 \oplus X_2 = (X_1 \setminus X_2) \cup (X_2 \setminus X_1)$. Thus, ⊕-NFAs are obtained from traditional NFAs by extending the transition function to words by using symmetric difference instead of union. A ⊕-DFA is any DFA obtained by applying the subset construction to a ⊕-NFA.

van Zijl considers enumeration of regular languages by the number of states in the ⊕-NFA and ⊕-DFA simultaneously. This problem has been considered for traditional NFAs and DFAs by Domaratzki *et al.* [9]. Let φ be the Euler totient function. We have the following result [35, Thm. 10]:

Theorem 15. *For all $n \geq 1$, there are at least $\frac{2^n}{n}\varphi(2^n - 1)$ distinct regular languages over a binary alphabet such that each is accepted by an n-state ⊕-NFA, and the minimal ⊕-DFA for each has $2^n - 1$ states.*

6 Enumeration by Regular Expression Size

Lee and Shallit have recently investigated the enumeration of regular languages by regular expression size [21]. This follows previous work, most recently by Ellul *et al.* [13], on the study of regular expression size as a descriptional complexity measure for regular languages. The work of Ellul *et al.* [13] includes investigations of trade-offs between regular expression size and automata size and the effect of operations on regular expression size.

The study of the descriptional complexity of regular expressions requires us to be precise about our measure of the length of a regular expression. For instance, Lee and Shallit [21] and Ellul *et al.* [13] consider the following three measures:

(a) The *ordinary length* of a regular expression, that is, the number of symbols in the regular expression, including parentheses, ϵ and $\emptyset$.

(b) The *reverse polish* length, which is the length of the equivalent expression written in reverse polish (postfix) notation.

(c) The *alphabetic length*, which counts only letters from the alphabet Σ, and ignores all operators, occurrences of ϵ and $\emptyset$, and parentheses.

Ellul *et al.* [13] note that each of these lengths is linearly related to each other, provided the expressions do not contain some basic forms of redundancy (such redundancy-avoiding expressions are called *irreducible* by Ellul *et al.* [13], where we refer the reader for more details).

The techniques of Lee and Shallit are themselves worth mentioning. The first step is constructing a CFG G such that $L(G)$ generates the language of all valid regular expressions over an alphabet Σ (i.e., $L(G)$ consists of words over the alphabet $\Sigma \cup \{(,), \emptyset, \epsilon, +, *\}$, each of which is a valid regular expression). Using the Chomsky-Shützenberger Theorem, G can be translated to a system of linear equations which (implicitly) give the number of regular expressions of a given length. Lee and Shallit then use Gröbner bases to obtain a generating function for the number of regular expressions of length n. This technique enumerates all valid regular expressions, which treats regular expressions as being distinct if they differ as words generated by the grammar G.

In the following, $S_k(n)$ denotes the number of valid regular expressions of ordinary length n over a k-letter alphabet. The following result is due to Lee and Shallit [21]:

Theorem 16. *The function $S_k(n)$ satisfies $S_k(n) \sim c_k \alpha_k^n n^{-3/2}$, for some constant c_k, where $\alpha_1 = 6.1552665$ and $\alpha_2 = 7.2700161767$.*

Clearly, $S_k(n)$ is an upper bound on the number of distinct regular languages generated by a regular expression of length n over a k-letter alphabet, denoted by $R_k(n)$. By further refining the grammars used to generate regular expressions to reduce the number of repeated regular expressions, Lee and Shallit give improved upper bounds on $R_k(n)$:

Theorem 17. *The function $R_k(n)$ satisfies the upper bounds in Table 1, where the length of the regular expressions is ordinary length.*

k	1	2	3	4	5	6
$R_k(n)$	$O(2.9090^n)$	$O(4.2198^n)$	$O(5.3182^n)$	$O(6.4068^n)$	$O(7.4736^n)$	$O(8.5261^n)$

Table 1: Upper bounds on $R_k(n)$ for $1 \leq k \leq 6$.

Lee and Shallit also give upper bounds for $R_k(n)$ using reverse polish and alphabetic length, as well as establish lower bounds on $R_k(n)$ [21]:

k	1	2	3	4	5	6
$R_k(n)$	$\Omega(1.3247^n)$	$\Omega(2.7799^n)$	$\Omega(3.9582^n)$	$\Omega(5.0629^n)$	$\Omega(6.1319^n)$	$\Omega(7.1804^n)$

Table 2: Lower bounds on $R_k(n)$ for $1 \leq k \leq 6$.

Theorem 18. *The function $R_k(n)$ satisfies the lower bounds in Table 2, where the length of the regular expressions is ordinary length.*

Again, lower bounds for reverse polish and alphabetic length are also given. The bounds in Table 2 are obtained by explicitly constructing large sets of distinct regular expressions of the given length. We note that Lee and Shallit also give bounds on the number of star-free and finite languages accepted by regular expressions of a given length.

7 Enumeration of ω-regular Languages

Finite automata recognizing infinite words are a classic model of study in the field of formal language theory. For an introduction to automata on infinite words see, e.g., Pin and Perrin [28] or Thomas [34]. A one-way infinite word w over the alphabet Σ is a mapping $w : \mathbb{N} \to \Sigma$. Denote $w_i = w(i)$. We view w as a word which has a starting point w_1 and proceeds to the right $w = w_1w_2w_3w_4\cdots$. The set of all one-way infinite words is denoted Σ^ω.

One model for accepting the ω-regular languages (i.e., the sets of one-way infinite words recognized by a regular expression involving the operator X^ω) are Müller automata. A (deterministic) Müller automaton M is given by a 5-tuple $M = (Q, \Sigma, \delta, q_0, \mathcal{F})$ where Q is a finite set of states, Σ is the alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state and $\mathcal{F} \subseteq 2^Q$ is the acceptance table. For any infinite word $w \in \Sigma^\omega$, w is accepted by a Müller automaton M if, when starting in the initial state, the set of states visited by w infinitely often is an element of $\mathcal{F}$. As usual, the language accepted by M is the set of all words accepted by M.

Let $f_k^{(\omega)}(n)$ be the number of distinct ω-regular languages accepted by a deterministic Müller automata with n states over a k-letter alphabet. Domaratzki [5] has given upper and lower bounds on the number of ω-regular languages accepted by Müller automata:

Theorem 19. *For all $k \geq 2$, there exists a constant γ_k depending only on k such that for all $n \geq 3$, the following bound hold:*

$$f_k^{(\omega)}(n) \leq \frac{n^{kn}\gamma_k^{n(1+o(1))}2^{2^n-n-1}}{(n-1)!} \cdot \sum_{m=0}^{k} \binom{n}{m}.$$

Further, for all $n > k \geq 2$,

$$f_k^{(\omega)}(n) \geq 2^{2^{n-\lfloor n/k \rfloor - 1}-1}.$$

Note that the constant γ_k in Theorem 19 is the same as the constant in (2). The upper bound of Theorem 19 is interesting, since it relies on the fact that some of the 2^{2^n} possible acceptance tables, called *strongly inadmissible* acceptance tables, are not valid for any possible assignment of transition functions [5].

8 Enumeration of Context-Free Languages

Domaratzki *et al.* [10] have recently considered enumeration questions for context-free languages. The main stumbling block to counting the number of context-free languages of a given size is the fact that deciding if two context-free grammars are equivalent (i.e., generate the same language) is undecidable. However, this does not preclude that the enumeration of context-free languages of size n is computable as a function of n. But in fact, it does turn out that the function counting the number of CFLs of a given size is uncomputable.

In the following theorem [10], we restrict our attention to descriptional complexity measures that are *well-behaved*. By well-behaved, we mean that the total number of CFGs of any given size is finite. We note that, for instance, the minimal number of nonterminals in any CFG generating a CFL is not a well-behaved descriptional complexity measure, since all finite CFLs are generated by CFGs with one nonterminal.

Theorem 20. *If $c(n)$ is the number of CFLs of size n (for any well-behaved, computable descriptional complexity measure), then $c(n)$ is uncomputable.*

However, despite the fact that the number of CFLs of size n is uncomputable, we can still approximate this quantity. For instance, it can be shown that the number of CFLs generated by a CFG in Chomsky Normal Form with at most n nonterminals over a fixed sized alphabet is $2^{\Theta(n^3)}$ [10, Thm. 7].

8.1 Related Enumeration Results

Theorem 20 can be extended to give a general result on the uncomputability of enumerative functions. In what follows, let X be a recursive language, d be a computable and well-behaved descriptional complexity measure, R be an equivalence relation on X and $g_R(n)$ denote the number of equivalence classes on the elements of measure n in X. Let Σ_k, Δ_k be levels in the arithmetic hierarchy. We have the following result [10]:

Theorem 21. *For any equivalence relation R on X that is complete for Σ_k or for Δ_k, the corresponding function $g_R(n)$ is not computable.*

For instance, Domaratzki *et al.* [10] note the following applications of Theorem 21:

- The number of distinct rational relations defined by nondeterministic finite transducers with n states is uncomputable.
- The number of distinct recursively enumerable languages recognized by Turing machines of size n is uncomputable.

However, not all equivalence relations are captured by Theorem 21. We mention some interesting open problems in this area in Section 9.

9 Open Problems

Enumeration of formal languages has several areas of investigation which are open. We mention some open problems which seem particularly interesting.

We first note some asymptotic bounds that we think might be easily improved. Enumeration of regular languages by nondeterministic state complexity is a very natural problem that has not received much attention. The bounds for many of these problems are likely to be able to be improved. We mention in particular the number of unary regular languages accepted by NFAs with n states as one such open problem. The current best known upper bound is logarithmically $n \log(cn) - n \log\log(n)$ while the best known lower bound is logarithmically $n + (c - o(1)) \sqrt{\frac{n}{\log n}}$.

Enumeration of automata accepting ω-regular languages is an interesting area which has received only minimal attention. The unique mode of acceptance for Müller automata presents an interesting enumeration problem, and some results have been obtained by Domaratzki [5]. However, tight bounds have not been obtained, and enumeration of Büchi automata has not been considered. The acceptance mode of Büchi automata yield a distinct notion of equivalence and it would be interesting to give asymptotics for the number of ω-regular languages accepted by Büchi automata with n states.

Theorem 21 gives a general result for proving that several enumerative functions are uncomputable. However, the result is not applicable in all cases. For instance, the following problem is open [10]: Is the number of regular languages generated by CFGs of a fixed size computable?

Recently, measuring the descriptional complexity of regular languages by the minimal number of transitions required by an NFA to recognize a language has

received increased attention. This raises the natural question: how many regular languages can be accepted by NFAs with at most n transitions? Gramlich and Schnitger give an upper bound on the number of binary regular languages accepted by an NFA with n transitions: they show that this quantity is at most n^{8n+2} [14]. We can also adapt the result of Liskovets [23] and Robinson [31] of Theorem 3 (see also Domaratzki *et al.* [9] for the case of NFAs) for enumerating the number of labelled, initially-connected NFAs over n states with m transitions. In particular, if $T_k(n, m)$ is the number of initially-connected NFAs with n states and m transitions over a k-letter alphabet (without final states), we can easily show that $T_k(n, m)$ satisfies the following recurrence:

$$\begin{aligned} T_k(1,1) &= k, \\ T_k(n,m) &= 0 \quad \text{if } n \geq m+2 \text{ and} \\ T_k(n,m) &= \binom{kn^2}{m} - \binom{kn(n-1)}{m} - \sum_{i=1}^{n-1}\sum_{j=1}^{m}\binom{n-1}{i-1}T_k(i,j)\binom{kn(n-i)}{(m-j)}. \end{aligned}$$

However, tight asymptotic bounds for enumerating regular languages by the number of transitions are unknown.

Enumeration by other descriptional complexity measures is also an area for future research. For instance, the measure of radius [12, 8] has been implicitly studied in relation to the enumeration of finite languages [7] and as descriptional complexity measure [2]. Further, simultaneous enumeration by several descriptional complexity measures has only received some attention in the literature [9, 35]. We feel that there are many interesting avenues of research in the area of enumeration of formal languages.

Finally, we note that explicitly computing values of the functions described here is often challenging for even small values of n. As an example, we note that the values of $G_1(n)$ (the number of unary regular languages accepted by NFAs with n states) is known only for values of $n \leq 6$.

10 Conclusions

Enumeration problems in formal language theory have many applications, and also presents interesting challenges relating to our understanding of the structure of language devices, especially distinctness and minimality. The recent work surveyed here shows that results in enumeration of formal languages often yield enlightening results that further our knowledge of the theory of formal languages in general. Though these fundamental questions have been examined for many years, interesting challenges still remain.

References

[1] Bassino, F., and Nicaud, C. Enumeration and random generation of accessible automata. *Submitted for publication* (2006). Retrieved from `http://www-igm.univ-mlv.fr/~bassino/publications/tcs06.ps`, April, 2006.

[2] Câmpeanu, C., and Ho, W. The maximum state complexity for finite languages. *Journal of Automata, Languages and Combinatorics 9*, 2–3 (2004), 189–202.

[3] Champarnaud, J.-M., and Paranthoën, T. Random generation of DFAs. *Theoretical Computer Science 330*, 2 (2005), 221–235.

[4] Chrobak, M. Finite automata and unary languages. *Theoretical Computer Science 47* (1986), 149–158.

[5] Domaratzki, M. On enumeration of Müller automata. In *Developments in Language Theory: 7th International Conference* (2003), Z. Ésik and Z. Fülöp, Eds., vol. 2710 of *Lecture Notes in Computer Science*, pp. 254–265.

[6] Domaratzki, M. Combinatorial interpretations of a generalization of the Genocchi numbers. *Journal of Integer Sequences 7* (2004), Article 04.3.6.

[7] Domaratzki, M. Improved bounds on the number of automata accepting finite languages. *International Journal of Foundations of Computer Science 15*, 1 (2004), 143–161.

[8] Domaratzki, M., Ellul, K., Shallit, J., and Wang, M.-W. Non-uniqueness and radius of cyclic unary NFAs. *International Journal of Foundations of Computer Science 16*, 5 (2004), 883–896.

[9] Domaratzki, M., Kisman, D., and Shallit, J. On the number of distinct languages accepted by finite automata with n states. *Journal of Automata, Languages and Combinatorics 7*, 4 (2002), 469–486.

[10] Domaratzki, M., Okhotin, A., and Shallit, J. Enumeration of context-free languages and related structures. In *Seventh International Workshop on Descriptional Complexity of Formal Systems: Proceedings* (2005), C. Mereghetti, B. Palano, G. Pighizzini, and D. Wotschke, Eds., pp. 85–96.

[11] Domaratzki, M., Pighizzini, G., and Shallit, J. Simulating finite automata with context-free grammars. *Information Processing Letters 84* (2002), 339–344.

[12] Ellul, K. Descriptional complexity measures of regular languages. Master's thesis, University of Waterloo, 2002.

[13] Ellul, K., Krawetz, B., Shallit, J., and Wang, M.-W. Regular expressions: New results and open problems. *Journal of Automata, Languages and Combinatorics 9*, 2–3 (2004), 233–256.

[14] Gramlich, G., and Schnitger, G. Minimizing NFAs and regular expressions. In *STACS 2005* (2005), V. Diekert and B. Durand, Eds., vol. 3404 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 399–411.

[15] Han, G.-N. Escaliers évalués et nombers classiques. *Publ. IRMA Strasbourg, Actes 24e Séminaire Lotharingien* (1993), 77–85.

[16] Harrison, M. A census of finite automata. *Canadian journal of mathematics 17* (1965), 100–113.

[17] Holzer, M., and Kutrib, M. Nondeterministic descriptional complexity of regular languages. *International Journal of Foundations of Computer Science 14*, 6 (2003), 1087–1102.

[18] Korshunov, A. Asymptotic estimates of the number of finite automata. *Cybernetics 3*, 2 (1967), 12–19.

[19] Korshunov, A. A survey of certain trends in automata theory (in Russian). *Diskretnyi Analiz 25* (1974), 19–55.

[20] Korshunov, A. Enumeration of finite automata (in Russian). *Problemy Kibernetiki 34* (1978), 5–82,272.

[21] Lee, J., and Shallit, J. Enumerating regular expressions and their languages. In *Implementations and Application of Automata* (2005), M. Domaratzki, A. Okhotin, K. Salomaa, and S. Yu, Eds., vol. 3317 of *Lecture Notes in Computer Science*, pp. 2–22.

[22] Liksovets, V. Exact enumeration of acyclic deterministic automata. *Discrete Applied Mathematics 154*, 3 (2006), 537–551.

[23] Liskovets, V. The number of connected initial automata. *Cybernetics 5* (1969), 259–262.

[24] Narushima, H. Principle of inclusion-exclusion on semilattices. *Journal of Combinatorial Theory, Series A 17* (1974), 196–203.

[25] Narushima, H. A survey of enumerative combinatorial theory and a problem. *Proceedings of the Faculty of Science of Tokai University 14* (1979), 1–10.

[26] Narushima, H. Principle of inclusion-exclusion on partially ordered sets. *Discrete Mathematics 42* (1982), 243–250.

[27] Nicaud, C. Average state complexity of operations on unary automata. In *Proc. 24th Symposium, Mathematical Foundations of Computer Science 1999* (1999), M. Kutylowski, L. Pacholski, and T. Wierzbicki, Eds., vol. 1672 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 231–240.

[28] Perrin, D., and Pin, J.-E. *Infinite Words*. Elsevier, 2004.

[29] Pomerance, C., Robson, J., and Shallit, J. Automaticity II: Descriptional complexity in the unary case. *Theoretical Computer Science 180* (1997), 181–201.

[30] Reis, R., Moreira, N., and Almeida, M. On the representation of finite automata. In *Seventh International Workshop on Descriptional Complexity of Formal Systems: Proceedings* (2005), C. Mereghetti, B. Palano, G. Pighizzini, and D. Wotschke, Eds., pp. 269–276.

[31] Robinson, R. W. Counting strongly connected finite automata. In *Graph Theory with Applications to Algorithms and Computer Science* (1985), Y. Alavi, G. Chartrand, L. Lesniak, D. Lick, and C. Wall, Eds., pp. 671–685.

[32] Rozenberg, G., and Salomaa, A., Eds. *Handbook of Formal Languages*. Springer-Verlag, 1997.

[33] Sloane, N. *On-line Encyclopedia of Integer Sequences*. Available Electronically at `http://research.att.com/~njas/sequences`, 2006.

[34] Thomas, W. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed. Elsevier, 1990, pp. 133–192.

[35] van Zijl, L. Magic numbers for symmetric difference NFAs. *International Journal of Foundations of Computer Science 16*, 5 (2005), 1027–1038.

[36] Yu, S. State complexity of the regular languages. *Journal of Automata, Languages and Combinatorics 6* (2001), 221–234.

[37] Yu, S. State complexity of finite and infinite regular languages. *Bulletin of the European Association of Theoretical Computer Science 76* (2002), 142–152.