IMRN International Mathematics Research Notices
1998, No. 15

Computational Classification of Numbers and Algebraic Properties

Robert M. Beals and Dinesh S. Thakur

1 Introduction

In this paper, we propose a computational classification of finite characteristic numbers (Laurent series with coefficients in a finite field), and prove that some classes have good algebraic properties. This provides tools from the theories of computation, formal languages, and formal logic for finer study of transcendence and algebraic independence questions. Using them, we place some well-known transcendental numbers occurring in number theory in the computational hierarchy.

Existence of or lack of patterns in the digit sequences of naturally occurring real numbers is a natural question. Rational numbers (and only rational numbers) have eventually periodic digit sequences. But the question has not been studied much for irrational real numbers, except for statistical studies on normality and randomness of digits for general numbers as well as special numbers such as π. Apart from the fact that no interesting patterns are found in general, the other reason for the lack of such studies is that irrationals usually are not naturally presented by their digit expansions (say decimals), at least in theoretical studies. Also, since it is hard to control carry-overs well, when we add or multiply, it is usually hard to manipulate the formulas to get good control on digit expansions of sums and products.

The situation seems to be much better for finite characteristic numbers: There are well-known strong analogies between integers, rationals, and reals on one hand, and polynomials, rational functions, and Laurent series (all with coefficients in a finite field) on the other. Again rational functions correspond to eventually periodic Laurent series expansions. Also the Laurent series representation is widely used: There are no

Received 18 May 1998. Revision received 1 July 1998
Communicated by Barry Mazur.

carry-over difficulties, and many times the expressions can be manipulated to find the expansions. There is also a remarkable result of Christol [Ch] (combined with work of Cobham) which says roughly that for Laurent series, being algebraic over the rationals corresponds exactly to digit patterns recognizable by a finite automaton, which is a very simple and weak model of a computer.

The notion of pattern is closely linked with the notion of computation: the stronger (easier) patterns can be produced by weaker (easier) machines. Most Laurent series (and real numbers) arising naturally in number theory/geometry are computable in the sense that they can be produced (see below for a more precise description of how this is done) by a Turing machine, which is the strongest theoretical model of the computer. There are in-between categories (hierarchies) of complexity studied in computer science, formal language theory, and logic; and remarkably, really diverse viewpoints have converged to the same notions. This is at least well known for the notion of computability: Recursive function theory, Church's lambda calculus, Turing machines, Post systems, generative grammars, cellular automata, etc., all lead to the same notion. Similarly, the notion of finite automata corresponds to regular languages, images of fixed points of uniform substitutions (uniform tag sequences), definability (in logic), and algebraicity (in the context of finite characteristic numbers). We give more examples in the next section.

Hence it seems natural to attempt a finer classification of transcendental Laurent series arising in number theory, by finding out where they fit in the hierarchy. Then we can use techniques of all these diverse fields for further study. For such a classification to be useful, it is necessary to relate closure properties of language classes to algebraic properties of the sets of numbers representable in the class. This is the focus of our study here. Basically, we classify numbers $\sum a_i t^i \in \mathbb{F}_q((t))$ by looking at what kind of machines can produce a_i, given i (say, positive), expanded in base q, as the input. (This is different than the study of (time or space) complexity of producing the first i terms of a power series.) Since there are only countably many computable numbers, we are really attempting a finer classification of this small class of numbers, which nonetheless contains many naturally occurring important numbers.

In the next section, we describe the computational classes we want to consider. We provide informal descriptions referring the reader to standard references for precise definitions.

After that, we prove some of the algebraic properties of the classification: We show that the class of context-free Laurent series is at least closed under addition of algebraic power series, as is the class of deterministic context-free Laurent series. Context-sensitive Laurent series form a field. Context-sensitive languages are exactly those languages recognizable by nondeterministic machines with space bounded by a linear function of the input length (which is roughly the logarithm of the input number). (Note that this standard usage of 'context-sensitive' is different than 'not context-free'.) Space complexity seems to be a very natural property of finite characteristic numbers. For any deterministic or nondeterministic space class at least linear, we show that the corresponding numbers form a field. In addition, PSPACE (i.e., polynomial space) Laurent series (and many other classes including the almost biggest class of recursive) form a field algebraically closed in Laurent series.

Making use of such good algebraic (as well as differential) closure properties of the classes, we can manipulate from numbers we are interested in to the numbers which easily yield to the standard tools of language theory, such as pumping lemmas, for instance. In the next section, we give such examples of classification for some important Laurent series such as analogues of π, e, and q-expansions of modular forms.

Finally, in the last section, we compare our results with other studies of power series in formal language theory and studies for real numbers: the results of Rice, Chomsky and Schützenberger, Mahler's classification, etc.

2 Computational classes and basic tools

In this section, we collect some basic information on computational classes and the corresponding language classes.

The most well-known hierarchy is the Chomsky hierarchy consisting of: finite automata (regular languages), pushdown automata (context-free languages), linear bounded automata (context-sensitive languages), and Turing machines (languages generated by an unrestricted grammar). See [HU], [S] for precise definitions and many equivalent descriptions. Informally, Turing machines have infinite two-directional writing tape, whereas linear bounded automata has tape size linear in the input size (but note that we can erase and use the same tape several times), and finite automata corresponds to zero or constant size working tape. For more on finite automata, especially in connection with number theory, the reader may look at [A1], [CKMR], [E], and perhaps [T3]. We will also consider some refined categories such as various space and time classes.

We give a brief summary of properties of the language classes we treat in this paper. These will help us later to establish whether a given power series belongs to the class or not. The main tools to show that something does not belong to a given class are various closure properties and the pumping lemma. We include, for completeness, some properties which we do not actually use (but which may be useful in further investigations). Unless noted otherwise, this material is from [HU], where many other closure properties that we do not discuss are proved.

Fact 1. The regular sets are closed under union, concatenation, complementation, and intersection. □

The "pumping lemmas" below are useful in proving that certain languages are not context-free. There are similar pumping lemmas for the class of regular languages and for some other language classes. (ϵ denotes the empty string.)

Fact 2. Let L be a context-free language. Then there is a constant n_0, such that for any $z \in L$ with length $|z| \geq n_0$, z is a concatenation $uvwxy$ such that $vx \neq \epsilon$, $|vwx| \leq n_0$, and for all $n \geq 0$, $uv^n wx^n y \in L$. □

Fact 3. Let L be a context-free language. Then there is a constant n_0, such that for any $z \in L$ with n_0 or more positions of z marked as distinguished, z is a concatenation $uvwxy$ such that v and x together have at least one distinguished position, vwx has at most n_0 distinguished positions, and for all $n \geq 0$, $uv^n wx^n y \in L$. □

Here the substrings v and x are said to be "pumped." Also note that this fact clearly implies the previous fact.

Context-free languages (CFL's) have many, but not all, of the closure properties of regular sets.

Fact 4. Context-free languages are closed under union, concatenation, and intersection with regular sets. Context-free languages are not closed under complementation or intersection. □

CFL's may be equivalently defined as those languages generated by context-free grammars, or those languages accepted by pushdown automata (PDA, i.e., nondeterministic machines with a finite state control and a stack). An important subclass of CFL's is the class of deterministic context-free languages (DCFL's), which are accepted by deterministic pushdown automata (DPDA or, equivalently, generated by $LR(k)$ grammars). (We note here that the class of deterministic finite automata is the same (as far as its computational power is concerned) as that of nondeterministic finite automata.) The DCFL's have different closure properties from the CFL's.

Fact 5. DCFL's are closed under complementation and intersection with a regular set. DCFL's are not closed under union or concatenation. There exist DCFL's L_1 and L_2 such that $L_1 \cap L_2$ is not a CFL (and therefore is not a DCFL). □

Now we turn our attention to space complexity. The most well-known space complexity class is the class of context-sensitive languages (CSL's) which are generated by context-sensitive grammars (the left-hand side of a production may contain more than one symbol, but the right-hand side must be at least as long). Equivalently, the CSL's are those languages which are accepted by a nondeterministic Turing machine which uses $O(n)$ tape cells on inputs of length n.

The class $\mathrm{NSPACE}(S(n))$ contains exactly those languages which are accepted by a nondeterministic machine which uses $O(S(n))$ tape cells on inputs of length n. Similarly, the class $\mathrm{DSPACE}(S(n))$ contains exactly those languages which are accepted by a deterministic machine which uses $O(S(n))$ tape cells on inputs of length n. Note that, if $S(n)$ is less than n, the input tape is regarded as read-only, and only the space on the work tape is counted.

A function S from the positive integers to the positive integers is *space constructible* if for some Turing machine M that is $S(n)$ space bounded, for all n, the machine M actually uses $S(n)$ tape cells on some input of length n. S is *fully space constructible* if, in addition, there is an M that uses $S(n)$ space on all inputs of length n. If $S(n) \geq n$ is space constructible, then S is fully space constructible. Most naturally occurring functions are fully space constructible. These include products of functions such as $\lfloor \log n \rfloor$, n, 2^n.

Clearly, deterministic space classes are closed under complementation. Neil Immerman and Robert Szelepcsényi independently proved the following (cf. [P]).

Fact 6. For $S(n) \geq \log n$, $\mathrm{NSPACE}(S(n))$ is closed under complementation. □

This fact is sometimes stated with the additional condition that $S(n)$ be fully space constructible. This condition is not needed (but makes the proof cleaner). If $L \in \mathrm{NSPACE}(S(n))$ for some $S(n) \geq \log n$ which is not fully space constructible, then there is a space constructible $S'(n) \geq \log n$ such that $L \in \mathrm{NSPACE}(S'(n))$ and for all n, $S'(n) \leq S(n)$: Simply let $S'(n)$ be the actual space bound of some nondeterministic machine accepting L in at most $S(n)$ space (and at least $\log n$ space). Since $S'(n) \geq \log n$, it is fully space constructible; so the complement of L is in $\mathrm{NSPACE}(S'(n)) \subseteq \mathrm{NSPACE}(S(n))$ as desired.

Clearly $\mathrm{DSPACE}(S(n)) \subseteq \mathrm{NSPACE}(S(n))$. It is not known precisely what relationship holds in the other direction. However, we know that for fully space constructible $S(n) \geq \log n$, $\mathrm{NSPACE}(S(n)) \subseteq \mathrm{DSPACE}(S(n)^2)$.

Fact 7. Let SPACE stand for DSPACE or NSPACE throughout. If $S_2(n)$ is space constructible and

$$\liminf_{n\to\infty} \frac{S_1(n)}{S_2(n)} \neq 0,$$

then there is a language in $\mathrm{SPACE}(S_2(n))$ which is not in $\mathrm{SPACE}(S_1(n))$. In particular, for $\epsilon > 0$ and $r \geq 0$, $\mathrm{SPACE}(n^r) \subsetneq \mathrm{SPACE}(n^{r+\epsilon})$. □

This fact is stated and proved in [HU] for DSPACE only. For NSPACE, a much more roundabout proof is used to obtain a weaker result. However, using the Immerman–

Szelepcsényi theorem, [HU] the proof for the DSPACE version of this fact can be readily adapted to prove the same result for NSPACE.

We also define the polynomial space class

$$\text{PSPACE} := \bigcup \text{DSPACE}(n^i) = \bigcup \text{NSPACE}(n^i).$$

The well-known class P (NP, respectively) of polynomial time (nondeterministic polynomial time, respectively) complexity is a (conjecturally proper) subset of PSPACE. But unlike the separation theorems above for the space classes, it is not even known (though it has been conjectured) that LOGSPACE := DSPACE(log(n)) is properly contained in P. So it is not known whether CFL is contained in LOGSPACE. On the other hand, it is easy to see that CFL does not contain LOGSPACE, because $\{a^n b^n c^n : n \geq 1\}$ is clearly in the latter, but well-known ([HU]) to be not CFL. It is known that

$$\text{DTIME}(S) \subseteq \text{NTIME}(S) \subseteq \text{DSPACE}(S) \subseteq \text{NSPACE}(S) \subseteq \bigcup_{c>0} \text{DTIME}(c^S),$$

$$\text{LOGSPACE} \subseteq \text{NSPACE}(\log n) \subseteq P \subseteq NP \subseteq \text{PSPACE} = N\,\text{PSPACE}$$

with at least one strict inclusion in each line. All the inclusions here are conjectured to be strict.

For ease of exposition, when considering space classes, we will consider machines which, on input N, compute the value of the t^N coefficient. Let L_c be the language over the alphabet $\Sigma_q := \{0, 1, \dots q-1\}$ consisting of the base q representations of those N for which the coefficient of t^N is c. Note that if all of the languages L_c, for $c \in \mathbb{F}_q$, are in NSPACE($S(n)$) or DSPACE($S(n)$), then a nondeterministic or, respectively, deterministic machine, can calculate the t^N coefficient in space $S(n)$, where n is the length of N. For a nondeterministic machine to compute a function, we mean that the machine halts with either the correct value of the function, or with an honest failure report. For any input, it must be the case that some computation path exists in which the machine halts with the correct function value.

Note that in any base, there are infinitely many representations of any number obtained by prepending zeroes. Of course, we do not want to have two different representations of the same number as elements of different L_c's. We simply require that numbers be written without leading zeroes. Since the set of strings of digits with no leading zeroes is a regular set, and all the classes we consider are closed under intersections with regular sets, we may henceforth ignore the problem of leading zeroes. That is, we simply do not care whether any strings with leading zeroes are in any of the L_c's, since they may be removed with no increase in complexity by an intersection with a regular set.

If L is a language over an alphabet Σ, we denote by $\overline{L}$ the complement $\Sigma^* \setminus L$.

3 Algebraic properties

In this section, we divide finite characteristic numbers (i.e., the Laurent series in $\mathbb{F}_q((t))$, where we fix $\mathbb{F}_q$) into computational classes and explore which classes have good algebraic properties.

Given a finite characteristic number $\sum a_i t^i$, we look at a machine which can, for each positive integer i, produce $a_i \in \mathbb{F}_q$ as output when given the base q-expansion of i as input. For example, we will say that $\sum a_i t^i$ is context-free if there is a pushdown automaton which accomplishes this task. Note that the problem of computing coefficients of a Laurent series is not a decision problem, and so does not correspond to a language membership problem, except when $q = 2$. However, we may, for a particular finite characteristic number $\alpha = \sum_i a_i t^i$, associate the languages $\{L_c \mid c \in \mathbb{F}_q\}$, where L_c consists of the base q representations of those i such that $a_i = c$. The second strategy clearly leads to the same classification for automata and space classes. This is true as well for context-free languages, because a PDA can nondeterministically guess c, and then simulate the PDA which accepts L_c, and output c if the simulation accepts.

In other words, whether we use one machine to compute coefficients or q machines to recognize coefficients, does not make a difference for these classes.

On the other hand, for DCFL's, the situation is different. It may happen that each L_c is a DCFL; yet no deterministic pushdown automaton can, on input i, compute the c such that $i \in L_c$. The difficulty is that the machines for the various languages L_c cannot be simulated simultaneously by a deterministic machine with a single stack. Of course, if $q = 2$, then there is no problem; L_0 and L_1 are merely the complements of each other, and the DPDA for L_0 can be made to output 0 instead of accepting, and output 1 instead of rejecting. However, consider the following example with $q \geq 5$: Let r, s, t, u be distinct nonzero elements of Σ_q. Let $L_1, L_2 \subseteq \Sigma_q^*$ be DCFL's such that $L_1 \cap L_2$ is not a CFL. Let $L_r = L_1 r$ (i.e., strings from L_1 with an r appended), $L_s = \overline{L_1} r$, $L_t = L_2 s$, $L_u = \overline{L_2} s$, $L_0 = \Sigma_q^*(\Sigma_q \setminus \{r, s\})$, and $L_x = \emptyset$ for all other $x \in \Sigma_q$. All of these languages are DCFL's, but we will see that no DPDA can compute c (to be consistent with the notation above, we identify here the sets $\mathbb{F}_q$ and Σ_q) given i, since such a machine could be modified to produce a PDA recognizing $L_1 \cap L_2$.

Suppose that some DPDA M computes c from i. To recognize $L_1 \cap L_2$, a PDA can essentially simulate the computation of c. To test if $w \in L_1 \cap L_2$, the machine simulates M on both wr and ws. To see how this is accomplished, note that the two simulations make the same stack moves right up to the last symbol, and that the stack moves on the last symbol are not so important: they need not be saved; only their effect on the state of the machine needs to be computed. Since $L_1 \cap L_2$ is not context-free, this is impossible.

For DCFL's, it seems reasonable to require the first (stronger in this case) condition; that a DPDA can compute the coefficient c from the exponent i.

For language class $\mathcal{C}$, the corresponding Laurent series set will be denoted by $\mathbb{F}_q((t))_{\mathcal{C}}$.

As mentioned in the introduction, the starting point of our investigation was the following result [Ch], [CKMR] of Christol.

Fact 8. Automatic Laurent series in $\mathbb{F}_q((t))$ are exactly the Laurent series algebraic over $\mathbb{F}_q(t)$, so that $\mathbb{F}_q((t))_{\mathrm{Aut}}$ is a field algebraically closed in the Laurent series. □

We digress for a moment to record a nice corollary (pointed out to us by Allouche): The series $\sum a_i t^{p^i} \in \mathbb{F}_p[[t]]$ is algebraic over $\mathbb{F}_p(t)$ if and only if $\sum a_i t^i \in \mathbb{F}_p(t)$. This is since both the statements are equivalent to the fact that a_i is an eventually periodic sequence.

In contrast to such nice properties for the automata, as we shall see, the next class in the Chomsky hierarchy of the context-free languages has only very weak algebraic properties.

Theorem 1. The class of context-free (or deterministic context-free) Laurent series is closed under addition of algebraic Laurent series, but is not closed under addition or multiplication in general. □

Proof. The first statement follows because a PDA (DPDA, respectively) can simulate a PDA (DPDA, respectively) and a finite state automaton in parallel (see 6.5 of [HU]), and add the coefficients obtained at the end.

For $q = 2$, this boils down to closure under the symmetric difference with a regular set:

$$\sum_{n \in L} t^n + \sum_{n \in R} t^n = \sum_{n \in (L \cup R) \setminus (L \cap R)} t^n.$$

Recall that we require both L and the complement $\overline{L}$ to be context-free, so the symmetric difference is a union $(L \cap \overline{R}) \cup (\overline{L} \cap R)$ of context-free languages, as is the complement of the symmetric difference.

For the negative statements, by "not being closed in general" we mean that for some q, counterexamples exist. While it seems likely that counterexamples exist for all q, we are content with proving the weaker statement.

We now consider the addition of two context-free Laurent series. It suffices to exhibit two deterministic context-free Laurent series such that the sum is not context-free. There are (see 6.4 of [HU]) deterministic context-free languages L_1, L_2, whose intersection is not context-free. Let $p \neq 2$ and $q = p^k$. Then

$$\sum_{n \in L_1} t^n + \sum_{n \in L_2} t^n = \sum_{n \in L_1 \cap L_2} 2t^n + \sum_{n \in (L_1 \cup L_2) - (L_1 \cap L_2)} t^n.$$

So, the sum is not context-free. In characteristic 2, with $q > 2$, we can modify this argument slightly: Let $\alpha \in \mathbb{F}_q$ with $\alpha \neq 0, 1$. Then the sum g given by

$$g(t) := \sum_{n \in L_1} t^n + \alpha \sum_{n \in L_2} t^n$$

has t^n coefficient $(\alpha + 1)$ exactly when $n \in L_1 \cap L_2$. For $q = 2$, something stronger is required of L_1 and L_2: their symmetric difference or its complement must not be a CFL. While it seems likely that such languages exist, the matter appears to be open.

For multiplication of context-free Laurent series, we focus on $q = 2$, although similar techniques should work in general. Consider $f = \sum t^{2^{2n} - 2^n}$. Since $1^n 0^n$ is the base 2 expansion of $2^{2n} - 2^n$, we see that f is context-free, in fact, deterministic context-free. We claim that f^3 is not context-free.

By the binomial expansion, it follows that the coefficient of t^n in f^3 is 1 if and only if $n \in L := \{3(2^{2n} - 2^n), 2(2^{2n} - 2^n) + (2^{2k} - 2^k), n \neq k\}$. We will show that L is not context-free by using Fact 3: If L is context-free, so is its intersection, say L_0, with the regular language $1^a 0^b 1^c 0^d$, with a, b, c, d positive, with $c + d$ even. By considering various carry-over possibilities, we see that L_0 consists of $z := 1^n 0^j 1^k 0^k$, with $j = n + 1 - 2k$, corresponding to the case $n \geq 2k - 1$. Now if we let k be large, and mark the last n_0 zeroes as distinguished, then x in Fact 3 must contain some zeroes on the right, but no ones. Now v can contain either digits from 0^j part or 1^k part, but not both, and in either case, the pumping takes you outside the language. ■

Next, we turn to the properties of the space classes. Let S be a function from positive integers to positive integers. We say that a finite characteristic number is in DSPACE(S) (respectively, NSPACE(S)) if the deterministic (respectively, nondeterministic) space complexity of computing the N-th coefficient is bounded above by $S(n)$, where n is the encoding length of N.

Theorem 2. Let a and b be finite characteristic numbers in DSPACE(S) (respectively, NSPACE(S)). Then $a + b$ is also in DSPACE(S) (respectively, NSPACE(S)). □

Proof. We describe the machine. On input N, of length n, the machine computes a_N using space $S(n)$. Since $a_N \in \mathbb{F}_q$, it can be stored in constant space. Then, reusing the $S(n)$ space, it computes b_N and the sum $a_N + b_N$. ■

Theorem 3. Let $S(n) \geq n$ for all n. Let a and b be finite characteristic numbers in DSPACE(S) (respectively, NSPACE(S)). Then ab is also in DSPACE(S) (respectively, NSPACE(S)). □

Proof. Let N be an input of length n. The N-th coefficient of ab is $\sum a_i b_{N-i}$. To compute this, a machine can run through all possible values of i, using space $O(n)$ to store both i

and $j = N - i$. For each pair i, j, the machine computes a_i and b_j, using space $S(n)$. This space is reused for each a_i and each b_j. The running total of the products $a_i b_j$ is kept using constant space. So the total space is $O(S(n))$. ■

Theorem 4. Let $S(n) \geq n$ for all n. Let a be a finite characteristic number in DSPACE(S) (respectively, NSPACE(S)), with $a \neq 0$. Then a^{-1} is also in DSPACE(S) (respectively, NSPACE(S)). □

Proof. We assume without loss of generality that $a \in 1 + t\mathbb{F}_q[[t]]$. Let $\alpha = 1 - a$. Let p be the characteristic, and let $\beta = 1 + \alpha + \cdots + \alpha^{p-1}$. By Theorems 2 and 3, β is in DSPACE(S) (respectively, NSPACE(S)). We have

$$a^{-1} = 1 + \alpha + \alpha^2 + \cdots = \beta\beta^p\beta^{p^2} \ldots .$$

Let $\beta = \sum_i b_i t^i$ (note $b_0 = 1$). Since p is the characteristic, $\beta^{p^k} = \sum_i (b_i)^{p^k} t^{ip^k}$. Let $\Omega = \Omega(N, n) = \{\overline{N} := (N_0, N_1, N_2, \ldots, N_{n-1}) \mid N_0 + pN_1 + \cdots + p^{n-1}N_{n-1} = N\}$, where as usual, n is the length of the encoding of N. Then the N-th coefficient of a^{-1} is

$$\sum_{\overline{N}\in\Omega} \prod_{i=0}^{n-1} (b_{N_i})^{p^i}.$$

We describe a space-efficient 'divide and conquer' strategy to compute this sum. Basically, the same space $S(n)$ is reused to compute all of the b_{N_i}, and the difficultly lies in keeping track of $N_0, N_1, \ldots, N_n$. Naively, this would take space n^2, since half of these n numbers are of length at least $n/2$. Let $f_k(m)$ be the t^m coefficient of $\beta^{(1+p+p^2+\cdots+p^{k-1})}$, i.e.,

$$f_k(m) = \sum_{\overline{N}\in\Omega(m,k)} \prod_{i=0}^{k-1} (b_{N_i})^{p^i}.$$

We use the recurrence

$$f_k(m) = \sum_{i+p^{\lfloor k/2\rfloor} j=m} f_{\lfloor k/2\rfloor}(i) f_{\lceil k/2\rceil}(j)^{p^{\lfloor k/2\rfloor}}.$$

So, to compute $f_k(m)$, we run through all possible values of i and j, recursively computing $f_{\lfloor k/2\rfloor}(i)$ and $f_{\lceil k/2\rceil}(j)$ (reusing space). This leads to an $S(n) + n\log n$ algorithm if we simply write down i, compute $f_{\lfloor k/2\rfloor}(i)$, then write down j (reusing the space where i was written), and compute $f_{\lceil k/2\rceil}(j)$, since the same $S(n)$ space is used for all computations of coefficients of β (the $\log n$ factor comes from the fact that the depth of the recursion is $\log n$, since each recursive call involves cutting k in half).

We improve on this by observing that i and j only require half the space as m to write down. This is because j simply is a number with half the digits of m, while i, which may be as long as m, must agree with m in the rightmost $\lfloor k/2\rfloor$ places. So each of i, j can be written in half the space as m, and the total space is therefore n plus the $S(n)$ that we use for computing coefficients of β. ■

Theorem 5. Let r be a Laurent series which is a root of $f(x) \in \mathbb{F}_q((t))[x]$. Suppose that the coefficients of f are in DSPACE(S) (respectively, NSPACE(S)) for some $S(n) \geq n$. Then r is in DSPACE($nS(n)$) (respectively, NSPACE($nS(n)$)). □

Proof. Let K be the field generated by the coefficients of f. By Theorems 2, 3, and 4, everything in K has space complexity $S(n)$. So we can assume without loss of generality that f is irreducible over K (otherwise consider the minimal polynomial of r), and that r is separable over K (otherwise consider r^p, r^{p^2}, etc., all of which have the same complexity).

Let $a, b \in K[x]$ be such that $af + bf' = 1$. Let $g(x) = x - b(x)f(x)$. Suppose that $\alpha \equiv r \bmod (t^i)$. Then $g(\alpha) \equiv r \bmod (t^{2i-c})$ for some constant c depending only on f, since $g(r) = r$ and $g'(r) = 0$.

Let $\alpha_0 \in \mathbb{F}_q$ be congruent to $r \bmod (t^{(c+1)})$. For $i \geq 0$, let $\alpha_{i+1} = g(\alpha_i)$. By induction, we have that $\alpha_i \equiv r \bmod (t^{2^i})$. Therefore, to compute the N-th coefficient of r, it suffices to compute the N-th coefficient of α_k for some k with $2^k > N$. Let n be the encoding length of N. Then for some $k = O(n)$, we have $2^k > N$.

Using the technique of Theorems 2 and 3, we may build a machine which, using work space $S(n)$, computes the coefficients of α_{i+1} given an oracle for the coefficients of α_i. To compute the N-th coefficient of α_k, it suffices to simulate k of these machines hooked together, which takes space $kS(n)$, which is $O(nS(n))$, as desired. ■

Theorem 6. (1) If $S(n) \geq n$, then the class of Laurent series corresponding to deterministic (nondeterministic, resp.) space class $S(n)$ form a field. In particular, context-sensitive Laurent series form a field.

(2) The Laurent series in PSPACE form a field ($\mathbb{F}_q((t))_{\text{PSPACE}}$ in our notation) algebraically closed in the field $\mathbb{F}_q((t))$ of all Laurent series. More generally, any space class of the form $\bigcup \text{DSPACE}(n^i S(n))$ or $\bigcup \text{NSPACE}(n^i S(n))$ (such as the class corresponding to exponential space or the Turing machines) has the same property. □

Proof. The first part follows from Theorems 2, 3, 4. For the second part, we also use Theorem 5: If g is in the space class for S and r is algebraically dependent on g, with dependency relation $P(g, r) = 0$, then f is just P developed as a polynomial in the second variable. ■

Remarks. (1) Theorem 5 can be used as a tool to prove algebraic independence of two given numbers by comparing their relative complexity.

(2) More generally, Fact 7 together with Theorem 5 implies, for example, that SPACE($n^{r+1+\epsilon}$) contains some element transcendental over SPACE(n^r).

(3) To compute $ia_i \in \mathbb{F}_q$, one needs to only look at the last digit of i, so all classes (including automata, context-free and space classes) are closed under derivatives. So by Theorems 3 and 4, the space classes for $S(n) \geq n$ are closed under derivative and logarithmic derivative. The same is true for automata. This is a quite useful tool to show nonmembership in a class, because logarithmic derivatives turn products into (sometimes simpler) sums, and derivatives kill all powers which are multiples of the characteristic.

(4) Closure under Hadamard products (i.e., term-wise products) for algebraic power series (so for automata by Christol's theorem) was proved in [F]. At the automata or regular languages level, it corresponds to direct product or intersection, respectively, and the property is very transparent and natural in that viewpoint. From a computational classes perspective, it is clearly valid for all space classes as well. It does not hold for CFL's: Let f_1 and f_2 be power series with only $0, 1$ coefficients. Then the Hadamard product corresponds to taking the intersection of the $L_1's$, but CFL's are not closed under intersection. (Indeed, the intersection of two DCFL's need not be context-free.)

(5) One can use these algebraic (and differential) properties to manipulate from numbers we are interested in to numbers whose Laurent series expansion yields more easily to computational analysis by various computational/ language theoretic tools such as a pumping lemma. This is illustrated in the examples in the next section.

(6) The class PSPACE has independent characterizations: A problem is in PSPACE if and only if it is describable in first-order logic with the addition of the partial fixed point operator if and only if it is describable in second-order logic with a transitive closure operator. See [I] and references therein for the terminology and many such equivalences, which also may lead to more tools.

4 Examples

In this section, we show where several naturally occurring finite characteristic numbers in number theory lie in our classification.

The theory of Drinfeld modules gives analogues of e, $2\pi i$, gamma, and zeta values. Also, one encounters periods of elliptic curves and Fourier expansions of modular forms in finite characteristic. Many of these were shown (see, e.g., [T3] for examples and references) to be nonautomatic (i.e., transcendental). Most of them are easily seen to be computable or even in PSPACE (this will be clear from the formulae below: For modular form expansions, this follows from their algebraic dependence on, say, theta and Eisenstein series). Here we attempt to pin some of them down more accurately.

First we consider $\tilde{\pi}$ which is a fundamental period of the Carlitz-Drinfeld exponential for $\mathbb{F}_q[T]$ and hence is a good analogue of $2\pi i$. It is known that $\tilde{\pi}^{q-1}$ is a Laurent series in $\mathbb{F}_q((t))$ (with $t = T^{-1}$) just as $(2\pi i)^2$ is a real number. Another simple way to get a Laurent series is to take its one unit part $\pi := T^{-q/(q-1)}\tilde{\pi}$. Note that for $q = 2$, $\pi = \tilde{\pi}/T^2$. For more on the analogies and connections with values of function field gamma function, see [T1], [T2].

Theorem 7. (1) $1/\pi$ is in LOGSPACE, π is in linear space. In particular, $\tilde{\pi}^{q-1}$ (which is just $\tilde{\pi}$ for $q = 2$) is context-sensitive. On the other hand, $1/\pi$ is not context-free.

(2) Let w be the logarithmic derivative of π with respect to T. Then w is context-sensitive, but not context-free. □

Proof. In [A2], it is shown that $1/\pi = \sum a_n (T^{-1})^n$, with $a_n = 0$ if n cannot be represented as a sum of distinct $q^j - 1$'s ($j > 0$), and with $a_n = (-1)^k$ if n is the sum of k such numbers (k is then uniquely defined). Note that this occurs if and only if for some m, the following hold:

(1) The base q representation of m has only zeroes and ones.

(2) k is the number of ones in the base q representation of m.

(3) $m - k = n$.

Since the encoding length of k is logarithmic in the encoding length of n, a machine can run through all possible values of k in LOGSPACE. For each value of k, the machine checks that $n + k$ has only zeroes and ones, and that k of the digits are one. Therefore, $1/\pi$ is in LOGSPACE. By Theorem 4, π is in linear space; i.e., it is context-sensitive. Theorem 3 and the remarks before the theorem now finish the proof of the first part of (1).

Suppose the language of n's with $a_n = -1$ is context-free. Then so is its intersection, say L, with the regular set $1^a 0^b$. Now let r be large and $f := q^r$. Then $n := (q^{r+f-1} - 1) + (q^{r+f-2} - 1) + \cdots + (q^r - 1)$ has base q expansion $1^{f-1}0^{r+1}$, and thus $n \in L$. Then by marking some 1's as distinguished, Fact 3 implies that for some $b > 0$ and c, we have $n_i := 1^{f-1+bi}0^{r+1+ci} \in L$, for all i. But this easily leads to contradiction, by looking for possibilities for m_i – (sum of digits of m_i) $= n_i$ as follows: Since $\sum_{j=0}^{l-1}(q^j - 1) < q^l - l$, the digits of m_i up to exponent $l - 1$ cannot influence the l-th digit of n_i. So m_i is forced to have the expansion of the form $1^{f-1+bi}a_{r+1+ci} \cdots a_1$, where a_j are zero or one. So in order to get the expression for n_i, with all those zeroes at the end, the $f - 1 + bi$ of the -1's occurring in $\sum(q^j - 1)$ from the 1^{f-1+bi} part have to cancel with $q^j - 1$'s from the next digits; i.e., $f - 1 + bi = \sum(q^{a_j(j-1)} - 1)$. Now Fact 3 implies that b, c are absolutely bounded, independently of r. So one can choose large enough r and i so that $q^r - r < bi < q^r - 1$. Then $q^r - 1 < f - 1 + bi < q^{r+1} - 1$. So when we express $f - 1 + bi$ as the sum of $q^j - 1$'s, one term has to be $q^r - 1 = f - 1$, but by the inequalities above, bi is not a sum of this type. This contradiction finishes the proof of (1).

Now we use the formula in [A3], [MY], namely, $w = \sum c_n (T^{-1})^{n+1}$ with $c_n = \sum_{q^j - 1 | n} 1 \in \mathbb{F}_q$. Just dividing n by $q^j - 1$'s one at a time in linear space (and reusing the space), we can see that w is context-sensitive. (Alternately, we can conclude this combining part (1) above, part (1) of Theorem 6, and Remark (3) of the last section.) Suppose it is also context-free; then its intersection with the regular set $\{q^u - 1\}$ (which consists of numbers with all digits $q-1$) is also context-free. But $c_{q^u - 1} = d(u) = \prod(u_i + 1)$, where $u = \prod p_i^{u_i}$ is the prime factorization of u. Now the subset of these $q^n - 1$'s, where $c_{q^n-1} = d(n) = 2$, is also context-free. But since all these numbers have $q - 1$ as the only digit, we can only pump it. This implies that if $d(n) = 2$, then for some $h > 0$ (pump-length), $d(n + hi)$ is also two for all i. Now for some m, the number $(nh)m + 1$ is a prime, so $d(n + h(mn^2)) = d(n)d((nh)m + 1) = 2d(n)$. This gives a contradiction if the characteristic is not two. On the other hand, if the characteristic is two, we look at the subset where $d(n)$ is odd; i.e., n is a square. By pumping, the context-free assumption implies $d(n + hi)$ are all odd, which means $n + hi$ is a square for all i, which is a contradiction again. This completes the proof. Note that we have generalized the proof of nonautomaticity of w (and hence of transcendence of π) by using the pumping lemma. ■

Next we look at an analogue of e given by the Carlitz-Drinfeld theory. Namely, $e = \sum D_i^{-1} \in \mathbb{F}_q((1/T))$, where $D_i = \prod_{0 \le j < i}(T^{q^i} - T^{q^j})$.

Theorem 8. e is in DSPACE(n) and so is context-sensitive. □

Proof. Since $D_i^{-1} \in t^{iq^i}\mathbb{F}_q[[t]]$, to compute the N-th coefficient of e, it suffices to consider D_i^{-1} for $i \le \log N$. So, using $\log n$ space (where n is the length of N), a machine can enumerate values of i such that $iq^i < \log N$. Since N is written in base q, once i is written down, using $\log n$ space for scrap work, the machine has access to the digits of $N - iq^i$. So it suffices to describe a machine which, in linear space, computes the N-th coefficient of $(t^{iq^i} D_i)^{-1}$ given N and i. Since

$$\left(t^{iq^i} D_i\right)^{-1} = \prod_{j=0}^{i} \left(1 - t^{q^i - q^j}\right)^{-1} = \prod_{j=0}^{i} \left(1 + t^{q^i - q^j} + t^{2\left(q^i - q^j\right)} + t^{3\left(q^i - q^j\right)} \cdots\right),$$

it follows that the N-th coefficient is the number of ways to write N as $\sum_{j<i} \ell_j (q^i - q^j)$ with nonnegative integers ℓ_j. Call this number $f_i(N)$; we have the recurrence

$$f_i(N) = \sum_{q^{\lfloor i/2 \rfloor} a + b = N} f_{\lceil i/2 \rceil}(a) + f_{\lfloor i/2 \rfloor - 1}(b).$$

As in the proof of Theorem 4, the machine can effect this recursion in linear space; either a or b can be written using only half as much space as n, since a is a number of half the length, and b agrees with N in the rightmost $n/2$ positions. ■

Next we look at the sequence of squares (essentially the expansion of the theta function) when $q = 2$.

Theorem 9. The set L of squares (and hence $\sum t^{n^2}$) is context-sensitive, and under the generalized Riemann hypothesis (GRH), it is even in LOGSPACE. But for $q = 2$, it is not context-free. □

Proof. L is easily recognizable (just check whether $a^2 = n$, reusing the same space for a with $a^2 \leq n$) in deterministic linear space. So, in particular, it is context-sensitive. In fact, one can do better: Under GRH, if n is not square, it is a quadratic nonresidue for a prime smaller than $c(\log n)^{2+\epsilon}$, and this can be checked in LOGSPACE: A machine need only check, for each $m < c(\log n)^3$, that n is a quadratic residue modulo m. Each such m can be written in $O(\log\log n)$ space (this space is reused for each m), which is logarithmic in the input length. Also, computing n modulo m and enumerating the quadratic residues modulo m can all be done in the same space.

Since CFL's are closed under intersections with regular sets, it suffices to show that $L_5 = L \cap \{x \in \{0,1\}^* \mid x \text{ has exactly five ones}\}$ is not context-free.

Note that L_5 contains the binary representations of elements of the following three sets: $A = \{(2^k + 3)^2 \mid k \geq 3\}$, $B = \{(3 \cdot 2^k + 1)^2 \mid k \geq 3\}$, and $C = \{(2^{2k-1} + 2^k + 1)^2 \mid k \geq 3\}$. Both A and B clearly correspond to context-free subsets of L_5, while C does not.

Most probably, L_5 contains binary representations of only finitely many odd numbers not in $A \cup B \cup C$ (among odd numbers with at most 64 bits, there are only 13 squares with exactly five one bits not in $A \cup B \cup C$; all of these have at most 26 bits). This would imply that L_5 is not context-free. However, our proof goes by another route.

Let n_0 be the pumping lemma constant for L_5, and take a string $z \in L_5$ encoding an element of C with $|z| \geq n_0$, and $|z|$ large enough that z neither begins nor ends with 1001. Then $z = uvwxy$, where $|vx| \geq 1$ and for $n \geq 0$, $uv^nwx^ny \in L_5$. Let $\sigma(n)$ be the number whose binary representation is uv^nwx^ny. Since v and x consist only of zeroes, $\sigma(n) = P(2^n)$ for some $P \in \mathbb{Z}[x]$. Since $P(2^n)$ is a square for all $n \geq 0$, it follows that $P(x)$ is a square in $\mathbb{Z}[x]$. We present the proof of this implication explained to us by Bjorn Poonen in Lemma 1 below.

Let $f(x) \in \mathbb{Z}[x]$ be such that $f(x)^2 = P(x)$. We see that P has three nonzero terms, from which it follows that f is a binomial: Let $P(x) = a^2x^{2k} + bx^\ell + c^2$ with $ac \neq 0$ (clearly the degree of P is even and the first and last coefficients are nonzero squares). Then f has degree k, and has nonzero constant term. The term of f with the least positive degree is necessarily the x^ℓ term: ℓ is the multiplicity of zero as a root of $P'(x) = 2f(x)f'(x)$, and therefore the multiplicity of zero as a root of $f'(x)$. By a similar consideration, with $x^kf(1/x)$

in the role of f, we see that the term of f with the greatest degree less than k is $\ell - k$. This is only possible if $k = \ell$ and $f(x) = ax^k + b$.

So $P(x) = a^2x^{2k} + 2acx^k + c^2$. If x is a sufficiently large power of two, then the number of ones in $P(x)$ is simply the sum of the numbers of ones in a^2, ac, and c^2. Since this sum is 5, at least one of the summands is one, so either a or c is a power of two.

Say a is a power of two; then c is not a power of two, so $2ac$ and c^2 each have at least two ones. But since the total number of ones is 5, $2ac$ and c^2 have exactly two ones. However, the only square with two ones is 9: if $r^2 - 1$ is a power of two, then both $r - 1$ and $r + 1$ are powers of two, so $r = 3$. So $c = 3$ if a is a power of two. On the other hand, if c is a power of two, then an identical argument yields $a = 3$.

However, z was chosen to be the binary representation of a sufficiently large element of C so as to neither begin nor end with the string 1001. This contradicts that $a = 3$ or $c = 3$, and finishes the proof modulo the following lemma. ■

Lemma 1 (Poonen). If $P \in \mathbb{Q}[x]$ is such that $P(4^k)$ is an integer square for all sufficiently large $k > 0$, then either $P(x)$ or $xP(x)$ is a square in $\mathbb{Q}[x]$. If in addition $P \in \mathbb{Z}[x]$, then $P(x)$ or $xP(x)$ is a square in $\mathbb{Z}[x]$. □

Proof. We may assume that $\deg P(x)$ is even (by multiplying by x if necessary). Clearly the leading coefficient of P must be positive.

Write $\sqrt{P(x)} = f(x) + e(x)$, where $f(x)$ is a polynomial in $\mathbb{R}[x]$, with positive leading coefficient, and $e(x)$ is a series over R in positive powers of $1/x$, which (by complex analysis) converges for $|x| > |\alpha|$, α being the largest complex zero of $P(x)$.

We show first that $e(x)$ is zero. If not, write $e(x) = e_r/x^r + \cdots$ with e_r nonzero. Let $d = \deg f$. Define integer constants b_i by $B(y) = \sum b_i y^i = \prod_{j=0}^{d}(y - 4^j)$ for an indeterminate y. Then grouping terms of equal degree in x shows that $\sum b_i f(4^i x)$ vanishes. Now the left-hand side of $\sum b_i \sqrt{P(4^i x)} = \sum b_i f(4^i x) + \sum b_i e(4^i x) = \sum b_i e(4^i x)$ is an integer when $x = 4^k$ for sufficiently large k, by assumption. The right-hand side is a series starting with the *nonzero* term $e_r 4^{-rk} B(4^{-r})/x^r$. For sufficiently large k, the value of such a (convergent) series at $x = 4^k$ is small but nonzero, and cannot be an integer. This contradiction shows that $e(x) = 0$.

Hence $f(x)$ is a polynomial in $\mathbb{R}[x]$. Its values at 4^k for sufficiently large k are integers, so by Lagrange interpolation, $f(x) \in Q[x]$.

If one knows moreover that $P(x)$ has integer coefficients, then so must $f(x)$. ■

We expect that the second assertion of the theorem generalizes to any q, but have not proved it. For connections with q-expansions of modular forms, we refer to [T3], [AT].

The tools seem to be successful in placing some natural numbers in Chomsky hierarchy, and one might hope to do the same for some gamma and zeta values in a function field setting. But to get a finer classification, tools need to be developed to show that certain languages are not in the given space class.

5 Complements

In this section, we comment on some related topics of interest.

(I) Formal power series have been studied in the literature on the formal language theory (see [SS] and references therein), but from the point of view which is most natural to language theory. Let us note some similarities and differences: One associates to a language its characteristic power series, which is a formal power series in noncommuting variables (alphabet), so that words in the language get coefficient one; others get zero. If you write this in exponential notation, we get the same series we consider, but the natural operations considered are different. For example, when we multiply two series, instead of algebraic multiplication, one combines exponents by concatenation rather than adding in some base. Even the commuting variables specialization is thus different. Also, the studies are not finite characteristic specific.

As an illustration, let us compare a special instance of the famous algebraicity result of Chomsky and Schützenberger for context-free languages. For alphabet $A := \{a, b\}$ and the context-free language $L := \{a^n b^n\}$, one associates the series $s := \sum a^n b^n$. It is algebraic in the sense that it satisfies the formal equation $s = asb + ab$. In our sense, of course, the corresponding series (say, with $a = 1$, $b = 0$, and $q = 2$, and with the words considered as the exponents) is transcendental, as the language is not regular.

In these studies, we also associate ([Pi]) with a language L as its generating function $\phi_L(t) := \sum c(n)t^n$, where $c(n)$ is the number of words in L of length n. It is known that if L is regular (recognizable by finite automata), then ϕ_L is a rational function. The Chomsky-Schützenberger result says that if L is context-free, unambiguous (i.e., only one way to derive from grammar), then $\phi_L(t)$ is algebraic over $Q(t)$. Again, this is quite different than our perspective.

(II) What can one say about digit expansions of real numbers in some base or of p-adic or λ-adic numbers? There is a classical result of Rice [R] which showed that recursive real numbers form a field which is algebraically closed in the field of real numbers. Though there is a lot of work done on complexity of the basic operations, it seems that no other such good class is known. The class of automatic real numbers is closed under addition and under multiplication by rationals ([L]), but it is not closed under multiplication or reciprocation ([LST]). We are currently investigating other classes.

In order not to give the wrong impression, that the finite characteristic numbers are much easier to deal with than the real or p-adic numbers, we point out that a naive analogue of famous Thue-Siegel-Roth theorems on diophantine approximations of real numbers fails quite badly (and the situation is still unclear) for finite characteristic numbers. (See [M3], [LM].) Also, from the point of view of logic and model theory, real and p-adic fields are well understood (form complete, model complete theory), whereas the Laurent series fields are still not understood. (See [C, p. 65] and [K].)

(III) On the other hand, there is a well-known classification of complex numbers z due to Mahler (see [M1], [B], [Sch]) into A, S, T, U numbers by optimizing diophantine approximation properties of values of all polynomials evaluated at z. The A numbers turn out to be algebraic numbers. The other classes also have the property that algebraically dependent (transcendental) numbers belong to the same class. But these classes (or simple modifications by taking unions) are not closed under addition or multiplication. Analogues of this classification have been studied also for the p-adic case ([M2]), finite characteristic Laurent series ([Bu]), and general Laurent series ([D]). In each case, there are four classes (subdivided often into finer classes, but the finer classes usually do not have the nice algebraic property mentioned above) with similar properties. But, except in the complex case, the A numbers are now algebraic by definition rather than by intrinsic diophantine approximation characterization. In all these situations, almost all numbers are S numbers, and Liouville numbers are U numbers. So there are uncountably many S and U numbers. In the complex case, Schmidt ([Sc]) proved the existence of (uncountably many) T-numbers.

We can also consider disjoint classes, e.g., Aut, PSPACE \ Aut, Comp \ PSPACE, and $\mathbb{F}_q((t)) \setminus$ Comp with the same algebraic dependence (and more) properties. (We can consider even the EXP class, for example.) But except for the $\mathbb{F}_q((t)) \setminus$ Comp class, our classes are countable. Also, since complex numbers form a two-dimensional space over the real numbers, it is easy to make computational classes for complex numbers (as in Mahler's classification), but the finite characteristic complex numbers form an infinite dimensional space over the finite characteristic real numbers. On the other hand, at least in Carlitz modules theory, most of the interesting numbers lie in the finite extension $\mathbb{F}_q((u))$, with $u = T^{-1/(q-1)}$.

(IV) Now let us point out some open questions which naturally arise: Can our classes be characterized by some diophantine approximation properties? Can one provide a class of interesting numbers which generate $\mathbb{F}_q((t))_{\mathcal{C}}$ or generate a field whose algebraic closure in the Laurent series is $\mathbb{F}_q((t))_{\mathcal{C}}$, say, for $\mathcal{C}$ = PSPACE or Comp? Does the well-known result of Cobham, which says that if m and n are multiplicatively independent (e.g., powers of distinct primes), then a nonperiodic sequence of integers given in base m or n cannot both be recognized by automata, generalize to other low complexity computational classes, say, LOGSPACE? (Recently, an interesting algorithm (see [BBP]) to compute the N-th digit base 16 of π, without computing all the previous digits, has been discovered. Unfortunately, its time complexity is worse than the time complexity of the fastest known algorithm to generate the first N digits, and also its space complexity is of order n^k, probably too large for Cobham type expectation.) What are the closure properties with respect to solutions of some interesting class of differential equations? What are the good time complexity or mixed complexity classes with good algebraic properties? (It is easy to see that they are closed under addition, but even for multiplication, the naive approach leads to the time exponential in the size of input.) What are the algebraic properties of, say, PSPACE for real or p-adic numbers?

Finally, we take this opportunity to correct an oversight in [T2]: In Theorem 2, the hypothesis that 'α_l are not all zero' is missing.

Acknowledgments

We thank Bjorn Poonen for providing us with the proof of Lemma 1. The second author was supported in part by National Science Foundation grant DMS 9623187.

References

[A1] J.-P. Allouche, *Automates finis en théorie des nombres*, Exposition Math. **5** (1987), 239–266.

[A2] ———, *Sur la transcendance de la série formelle* Π, Sém. Théor. Nombres Bordeaux **2** (1990), 103–117.

[A3] ———, *Transcendence of the Carlitz-Goss gamma function at rational arguments*, J. Number Theory **60** (1996), 318–328.

[AT] J.-P. Allouche and D. Thakur, *Automata and transcendence of the Tate period in finite characteristic*, to appear in Proc. Amer. Math. Soc.

[BBP] D. Bailey, P. Borwein, and S. Plouffe, *On the rapid computation of various polylogarithmic constants*, Math. Comp. **66** (1997), 903–913.

[B] A. Baker, *Transcendental Number Theory*, Cambridge University Press, London, 1975.

[Bu] P. Bundschuh, *Transzendenzmasse in Körpern formaler Laurentreihen*, J. Reine Angew. Math. **299/300** (1978), 411–432.

[C] G. Cherlin, *Model Theoretic Algebra*, Lecture Notes in Math. **521**, Springer, New York, 1976.

[Ch] G. Christol, *Ensembles presque périodiques* k*-reconnaissables*, Theoret. Comput. Sci. **9** (1979), 141–145.

[CKMR] G. Christol, et. al., *Suites algébriques*, *automates et substitutions*, Bull. Soc. Math. France **108** (1980), 401–419.

[D] E. Dubois, *On Mahler's classification in Laurent series fields*, Rocky Mountain J. Math. **26** (1996), 1003–1016.

[E] S. Eilenberg, *Automata*, *Languages*, *and Machines*, *Vol. A*, Pure Appl. Math **58**, Academic Press, New York, 1974.

[F] H. Furstenberg, *Algebraic functions over finite fields*, J. Algebra **7** (1967), 271–277.

[HU] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Ser. Comp. Sci., Addison-Wesley, Reading, Mass., 1979.

[I] N. Immerman, *Descriptive complexity: A logician's approach to computation*, Notices Amer. Math. Soc. **42** (1995), 1127–1133.

[K] F. Kuhlman, *Elementary properties of power series fields over finite fields*, preprint, 1997.

[LM] A. Lasjaunias and B. de Mathan, *Thue's theorem in positive characteristic*, J. Reine Angew. Math. **473** (1996), 195–206.

[L] S. Lehr, *Sums and rational multiples of* q*-automatic sequences are* q*-automatic*, Theoret. Comput. Sci. **108** (1993), 385–391.

[LST] S. Lehr, J. Shallit, and J. Tromp, *On the vector space of automatic reals*, Theoret. Comput. Sci. **163** (1996), 193–210.

[M1] K. Mahler, *Zur Approximation der Exponentialfunktion und des Logarithmus I*, J. Reine Angew. Math. **166** (1932), 118–136.

[M2] ———, *Über eine klassen-einteilung der p-adischen zahlen*, Mathematica Leiden **3** (1935), 177–185; Zentralblatt fur̈ math. **11** (1935), 58.

[M3] ———, *On a theorem of Liouville in fields of positive characteristic*, Canad. J. Math. **1** (1949), 397–400.

[MY] M. Mendès France and J. Yao, *Transcendence and the Carlitz-Goss gamma function*, J. Number Theory **63** (1997), 396–402.

[P] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Mass., 1994.

[Pi] N. Pippenger, *Theories of Computability*, Cambridge University Press, Cambridge, 1997.

[R] H. G. Rice, *Recursive real numbers*, Proc. Amer. Math. Soc. **5** (1954), 784–791.

[S] A. Salomaa, *Computation and Automata*, Encyclopedia Math. Appl. **25**, Cambridge University Press, Cambridge, 1985.

[SS] A. Salomaa and M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Texts Monogr. Comput. Sci., Springer-Verlag, New York, 1978.

[Sc] W. Schmidt, "T-numbers do exist" in *Symp. Mathematica IV (INDAM, Rome, 1968/69)*, Academic Press, London (1970), 3–26.

[Sch] Th. Schneider, *Introduction aux nombres transcendants*, Gauthier-Villars, Paris, 1959.

[T1] D. Thakur, *Gamma functions for function fields and Drinfeld modules*, Ann. of Math. **134** (1991), 25–64.

[T2] ———, *Transcendence of Gamma values for* $\mathbb{F}_q[T]$, Ann. of Math. **144** (1996), 181–188.

[T3] ———, "Automata and transcendence" in *Number Theory (Tiruchirapalli, 1996)*, Contemp. Math. **210**, Amer. Math. Soc., Providence, 1998, 387–399.

Beals: University of Arizona, Department of Mathematics, Tucson, Arizona 85721, USA;
beals@math.arizona.edu

Thakur: University of Arizona, Department of Mathematics, Tucson, Arizona 85721, USA;
thakur@math.arizona.edu