Sara Kropf

Asymptotic Analysis of Sequences Defined by Automata

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin der Technischen Wissenschaften

Alpen-Adria-Universität Klagenfurt

Fakultät für Technische Wissenschaften

Betreuer und Erstgutachter
Univ.-Prof. Dr. Clemens Heuberger
Alpen-Adria-Universität Klagenfurt
Institut für Mathematik

Zweitgutachter
Prof. Dr. Stephan Wagner
Stellenbosch University
Department of Mathematical Sciences

Klagenfurt am Wörthersee, Juni 2015

"The shortest path between two truths in the real domain passes through the complex domain."
Jacques Hadamard, 1865–1963

Abstract. In cryptography, scalar multiplication with large numbers is a frequently used operation. It is usually implemented using a digit representation of the scalar. The choice of the representation influences the running time of this scalar multiplication. Thus it is one possibility to optimize cryptographic algorithms and increase the security level.

More specifically, the running time of the multiplication depends on certain parameters of the digit representation. Such parameters can be the sum of digits, or the number of non-zero digits. To give precise estimates of the running time of an algorithm, a precise asymptotic analysis of these parameters of the chosen digit representation is of interest.

For syntactically defined digit representations, these parameters can be computed as the output sum of finite state machines. In this thesis, the output sum of an arbitrary finite state machine is analyzed asymptotically. This includes expectation, variance, covariance and central limit theorem for a sequence defined as the output sum of a finite state machine. For the moments, not only the main terms are provided, but also the existence of a continuous, nowhere differentiable, periodic function as the second order term is proved and a formula for the coefficients of its Fourier series is given.

This setting of sequences which can be defined as the output sum of finite state machines includes many well-studied sequences, like q-additive functions, digital sequences and the sum-of-digits function and the Hamming weight of syntactically defined digit representations. Thus, the results in this work generalize well known facts about the asymptotic behavior of all these sequences. Also, sequences defined by certain recursions fit into this setting.

The output sum of a finite state machine is in general asymptotically normally distributed. However, this is not guaranteed: It may happen that the output sum degenerates. In this work, this case is characterized algebraically and combinatorially. It turns out that this degeneration only happens in trivial cases and the conditions can be checked easily by inspecting a representation of the finite state machine as a graph.

These results are applied to asymptotically analyze several sequences in this thesis. This includes the sequence of carries occurring when performing addition of two numbers given in a syntactically defined digit representation. Furthermore, the number of iterations of von Neumann's addition algorithm is analyzed by using automata. However, this number of iterations is not asymptotically normally distributed, but converges to a double exponential distribution.

All the steps to obtain an asymptotic analysis of a given sequence, starting with explicitly constructing the corresponding finite state machine and stopping finally with the computation of the Fourier coefficients, can be performed automatically in the mathematical software system SageMath. To do this, the new finite state machine package, developed in the framework of this thesis, is presented. In the meanwhile, this package is included in the mathematical software system SageMath. This package was developed to conveniently work with automata and transducers. Thus, all results of this thesis are implemented as methods of this finite state machine package and the examples are computed by using them accordingly.

Kurzfassung. In der Kryptographie ist Skalarmultiplikation mit großen Zahlen eine häufig verwendete Operation. Diese wird oft unter Verwendung von Ziffernentwicklungen implementiert. Die Wahl der Ziffernentwicklung beeinflusst die Laufzeit der Skalarmultiplikation. Deshalb bietet sie eine Möglichkeit, den kryptographischen Algorithmus zu optimieren und dadurch das Sicherheitsniveau zu erhöhen.

Genauer gesagt wird die Laufzeit der Skalarmultiplikation von der Größe von bestimmten Parametern der Ziffernentwicklung beeinflusst. Beispiele für solche Parameter sind die Ziffernsumme und das Hamminggewicht. Um genaue Angaben zur Laufzeit eines Algorithmus zu geben, ist eine genaue asymptotische Analyse dieser Parameter notwendig.

Solche Parameter von syntaktisch definierten Ziffernentwicklungen können als Outputsumme von endlichen Zustandsautomaten berechnet werden. In dieser Dissertation werden solche Outputsummen asymptotisch analysiert. Die Resultate geben Auskunft über den Erwartungswert, die Varianz und die Kovarianz von Folgen, welche als Outputsumme von Automaten definiert sind. Auch zentrale Grenzwertsätze werden bewiesen. Dabei sind bei den Momenten nicht nur die Hauptterme von Interesse, sondern auch die Terme zweiter Ordnung, die im Allgemeinen durch stetige, nirgends differenzierbare, periodische Funktionen gegeben sind. Die Fourierkoeffizienten dieser Funktionen werden ebenfalls bestimmt.

Beispiele für Folgen, welche als Outputsumme von endlichen Automaten definiert werden können, sind q-additive Funktionen, Block-q-additive Funktionen, sowie Ziffernsumme und Hamminggewicht von syntaktisch definierten Ziffernentwicklungen. Deshalb verallgemeinern die Resultate dieser Arbeit viele bekannte Tatsachen über das asymptotische Verhalten dieser Folgen. Weiters können spezielle Rekursionen auch von Automaten berechnet und damit asymptotisch analysiert werden.

Im Allgemeinen ist die Outputsumme eines Automaten asymptotisch normal verteilt. Allerdings ist das nicht immer der Fall: Es kann auch der Fall eintreten, dass die Outputsumme eine degenerierte Zufallsvariable ist. Dieser Fall wird in dieser Dissertation algebraisch und kombinatorisch charakterisiert. Dabei stellt sich heraus, dass die Bedingungen für diesen Fall anhand der Darstellung des Automaten als Graph einfach überprüft werden können.

Die Resultate dieser Arbeit werden auf mehrere Folgen angewandt, um neue asymptotische Ergebnisse zu erhalten. Eine dieser Folgen besteht aus den Überträgen, die im Laufe einer schriftlichen Addition auftreten. Dabei sind beide Summanden in einer syntaktisch definierten Ziffernentwicklung gegeben. Auch die Anzahl der Iterationen der Von-Neumann-Addition wird mit Hilfe von Automaten asymptotisch analysiert. Das ist ein Beispiel einer asymptotisch nicht normal verteilten Folge: Die Anzahl der Iterationen konvergiert zu einer Gumbel-Verteilung.

Die einzelnen Schritte der asymptotischen Analyse können automatisiert werden: Von der konkreten Konstruktion der Automaten bis zur Berechnung der Fourierkoeffizienten können die einzelnen Teile im mathematischen Softwaresystem SageMath durchgeführt werden. Dazu wird in dieser Arbeit das Automatenpaket präsentiert, welches im Rahmen dieser Dissertation implementiert wurde. Mittlerweile ist dieses Paket ein integraler Bestandteil von SageMath. Die Idee dieses Pakets ist es, praktische Funktionen zu bieten, um mit Automaten zu arbeiten. Deshalb sind auch alle Resultate dieser Dissertation Teil dieses Pakets, und illustrierenden Beispiele wurden mit Hilfe dieses Pakets berechnet.

Eidesstattliche Erklärung.

Ich versichere an Eides statt, dass ich

- die eingereichte wissenschaftliche Arbeit selbstständig verfasst und andere als die angegebenen Hilfsmittel nicht benutzt habe,
- die während des Arbeitsvorganges von dritter Seite erfahrene Unterstützung, einschließlich signifikanter Betreuungshinweise, vollständig offengelegt habe,
- die Inhalte, die ich aus Werken Dritter oder eigenen Werken wortwörtlich oder sinngemäß übernommen habe, in geeigneter Form gekennzeichnet und den Ursprung der Information durch möglichst exakte Quellenangaben (z.B. in Fußnoten) ersichtlich gemacht habe,
- die Arbeit bisher weder im Inland noch im Ausland einer Prüfungsbehörde vorgelegt habe und
- zur Plagiatskontrolle eine digitale Version der Arbeit eingereicht habe, die mit der gedruckten Version übereinstimmt.

Ich bin mir bewusst, dass eine tatsachenwidrige Erklärung rechtliche Folgen haben wird.

Ort, Datum Unterschrift

Acknowledgment. While writing this thesis, I was financially supported by the Austrian Science Fund (FWF): P 24644-N26. This thesis was partially written in the framework of the Karl Popper Kolleg "Modeling-Simulation-Optimization" funded by the Alpen-Adria-Universität Klagenfurt and by the Carinthian Economic Promotion Fund (KWF). My research stays abroad were funded by the Mobilitätsförderung für NachwuchswissenschaftlerInnen of the Alpen-Adria-Universität Klagenfurt.

I want to thank my advisor Clemens Heuberger for his support and for the time he devoted to discuss mathematical issues with me. I also want to thank my colleagues at the department for these nice working conditions. Furthermore, I am grateful to Hsien-Kuei Hwang, Helmut Prodinger and Stephan Wagner, who gave me the possibilities of research stays abroad.

Finally, I am very grateful to my family and my friends for their support during my studies. Thank you for listening to me, even in times I was talking a lot about mathematics.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Over the last decades, asymptotic properties of digital sequences have been studied by many authors. The simplest example is the q-ary sum of digits, see Delange [20]. This has been generalized to various other number systems (cf. [9, 29, 35, 36, 40, 50, 57, 69, 71, 99]). Similar results have been obtained for other digital sequences (cf. [10, 15]). Frequently observed phenomena in the asymptotic analysis of these sequences include periodic fluctuations in the second order term and asymptotic normality (see also [25]).

To illustrate the type of results we prove in this thesis, we recall the corresponding results for the sum-of-digits function. These results were obtained step by step in many different papers (cf. [18]).

Theorem 1.1 ([18–21, 67, 70])**.** *Let N be a fixed integer The expected value of the q-ary sum-of-digits function s_q for a equidistributed integer in the interval $[0, N)$ is*

$$\mathbb{E}(s_q(n)) = \frac{1}{N}\sum_{0\le n<N} s_q(n) = \frac{q-1}{2}\log_q N + \Psi_1(\log_q N),$$

with a periodic, continuous, nowhere differentiable function Ψ_1.

The variance of the q-ary sum-of-digits function is

$$\begin{aligned}\mathbb{V}(s_q(n)) &= \frac{1}{N}\sum_{0\le n<N} s_q(n)^2 - \mathbb{E}(s_q(n))^2\\ &= \frac{q^2-1}{12}\log_q N - \Psi_1^2(\log_q N) + \Psi_2(\log_q N)\end{aligned}$$

with a periodic, continuous function Ψ_2.

The q-ary sum-of-digits function is asymptotically normally distributed. In particular, we have

$$\mathbb{P}\left(\frac{s_q(n) - \frac{q-1}{2}\log_q N}{\sqrt{\frac{q^2-1}{12}\log_q N}} < x\right) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left(-\frac{1}{2}y^2\right)dy + \mathcal{O}(\log^{-\frac{1}{2}} N)$$

for all $x \in \mathbb{R}$.

The Fourier coefficients of the Fourier expansion $\sum_{k\in\mathbb{Z}} c_k \exp(2\pi i k x)$ of the periodic function Ψ_1 are given by

$$\begin{aligned}c_0 &= \frac{q-1}{2\log q}(\log(2\pi) - 1) - \frac{q+1}{4},\\ c_k &= -\frac{q-1}{\frac{2\pi i k}{\log q}(1+\frac{2\pi i k}{\log q})\log q}\zeta\Big(\frac{2\pi i k}{\log q}\Big) \quad \text{for } k \neq 0\end{aligned}$$

where ζ denotes the Riemann ζ-function.

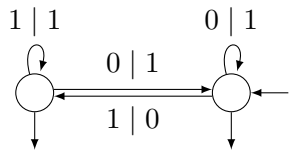


FIGURE 1.1. A small example of a transducer.

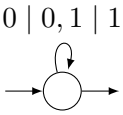


FIGURE 1.2. Transducer to compute the binary sum-of-digits function.

The purpose of this thesis is to use finite state machines as a uniform framework to derive such asymptotic results. The results mentioned above will follow as corollaries from our main results. We also analyze new sequences, like the number of carries occurring in the addition of two digit expansions.

Our main focus lies on transducers: These finite state machines transform input words to output words using a finite memory. A transducer is defined to consist of a finite set of states, an initial state, a set of final states, an input alphabet, an output alphabet and a finite set of transitions, where a transition starts in one state, leads to another state and has an input and an output label from the corresponding alphabet. Depending on the purpose in the different chapters, this definition will be refined. Especially, an automaton is a transducer with no output labels. An example of a transducer is given in Figure 1.1. We label the transitions with "input label | output label". The initial state is marked by an ingoing arrow starting at no other state and the final states are marked by outgoing arrows leading to no other state.

The input of the transducer is a random word. In some cases, we also use the digit expansion of a random integer as the input. The probability model of the input sequence varies in the different chapters. Depending on whether we are only interested in the main terms or also in the periodic fluctuation in the second order term, we choose one of the following two probability models:

- All digit expansions of integers less than a fixed number N are equally likely (see Chapter 2).
- All digit expansions or sequences of a fixed length ℓ are equally likely (see Chapters 3–6). In Chapters 3 and 6, the single digits of the digit expansions are independent. In Chapters 4 and 5, they are no longer independent in general.

We asymptotically analyze the output sum of a transducer, that is the sum of the sequence of output labels of the transducer for a given input sequence. For example, the transducer in Figure 1.2 computes the binary sum-of-digits function. The results in Theorem 1.1 follow from our asymptotic analysis of the output sum of this transducer.

In Figure 1.3, there is a further example of a transducer. Its output sum is the Hamming weight of the non-adjacent form. The non-adjacent form [89] is a digit expansion with base 2, digits -1, 0 and 1, and the syntactical rule that at least one of any two adjacent digits is non-zero. The non-adjacent form uniquely exists for all integers. For example, the non-adjacent form of 27 is $(100\bar{1}0\bar{1})_2$ where $\bar{1} = -1$.

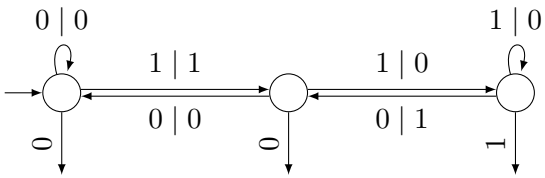


FIGURE 1.3. Transducer to compute the Hamming weight of the non-adjacent form.

Several notions abstracting the sum-of-digits and related problems have been studied. One of them is the notion of completely q-additive functions $a\colon \mathbb{N}_0 \to \mathbb{R}$ with

$$a(qn + \lambda) = a(n) + a(\lambda)$$

for $0 \le \lambda < q$ and $q \ge 2$ (cf. [10]). These have been generalized to digital sequences as defined in [1, 15]: A sequence $a(n)$ is a digital sequence if it can be represented as a sum $\sum_w f(w)$ where f is a given function and w runs over all windows of a fixed length κ of the q-ary digit representation of n. These digital sequences can easily be formulated by transducers.

Similarly to the previously mentioned results for the sum-of-digits function, we obtain the following asymptotic results for the output sum of a transducer in Chapter 2: The main term, the periodic fluctuation and an error term of the expected value and the variance of this sequence are established. The periodic fluctuation of the expected value is Hölder continuous and, in many cases, nowhere differentiable. A general formula for the Fourier coefficients of this periodic function is derived. Furthermore, it turns out that the sequence is asymptotically normally distributed for many transducers.

By definition, a finite state machine deterministically transforms an input sequence into an output sequence. Thus the output depends deterministically on the input. However, choosing a random input sequence, the dependence between the two random variables *sum of the input sequence* and *sum of the output sequence* may become negligible for long input sequences. Also if we consider two different outputs of a transducer, the two different output sums may become independent for long input sequences. We investigate for which transducers this is the case. The dependency is given by the covariance between the input and output sum (or between the two output sums): If the covariance goes to infinity, then the two random variables are asymptotically dependent.

There are many results on the variance of the sum of the output of explicit transducers under the same probability model as we use. See, for example, [6,36,38,42] for the variance of the Hamming weight of different digit expansions which are computed by transducers. In [60], the authors count the occurrences of a digit and give the expected value, the variance and the covariance between two different digits. The occurrence of a specific pattern in a word is investigated in e.g. [11, 31, 34, 80] (with generalizations to other probability models, too). In [11], the covariance between different patterns is also considered. In [40], Grabner and Thuswaldner consider a transducer whose output is the sum of digit function. However, they were only interested in the output and did not consider the joint distribution or the covariance of the input and output sum.

By contrast, we are interested in the joint distribution of the input and the output sum (or, more generally, of two different output sums) for a general transducer. We not only algebraically compute the expected value and the variance-covariance matrix of this distribution, but we also give combinatorial descriptions of these values. In particular, we combinatorially

characterize independent transducers in Chapters 3 and 4. This combinatorial connection is described by a condition on some weighted number of functional digraphs or on each cycle of the underlying graph of the transducer. To obtain these results, we apply a generalization of the Matrix-Tree Theorem by Chaiken [16] and Moon [75].

In many contexts, an unbounded variance (as in [66]) is necessary to prove a Gaussian limit law. In Chapters 3 and 4, we combinatorially describe transducers whose output sums have bounded variance. For strongly connected transducers, we prove that this is the case if and only if there exists a constant such that for each cycle, the output sum is proportional to its length with this proportionality constant. This is in turn equivalent to a quasi-deterministic output sum in the sense that the difference of the output sum and its expected value is bounded for *all* events, independently of the length of the input. In the special case where the transducer is strongly connected and aperiodic and the only possible outputs are 0 and 1, it turns out that the output sum has asymptotically bounded variance if and only if the output is constant for all transitions. This result is also extended to the joint distribution of two or more different output sums of one transducer.

As an application of the previous results, we analyze addition. Addition is an essential arithmetic operation in many algorithms. As the efficiency of addition is influenced by the number of occurring carries, we asymptotically analyze this number, which depends on the base and the digit set of the digit expansion.

We consider two different types of digit expansions: (q, d)-expansions and symmetric signed digit expansions [58], and two different types of addition: standard addition and von Neumann's addition [101].

Diaconis and Fulman [22] and Nakano and Sadahiro [79] consider the carries of the standard addition as a Markov chain. This is only valid if the digits of the digit expansion are independent. In their analysis, they obtain a stationary distribution. In Chapter 5, we determine the expectation and the variance of the number of positive and negative carries as well as the covariance between the positive and negative carries in the (q, d)-system and the symmetric signed digit system. Furthermore, we prove a central limit theorem for these numbers. The authors of [22] concentrate on an *odd* basis q and the symmetric digit set $\{-(q-1)/2, \dots, (q-1)/2\}$. The symmetric signed digit expansion (defined later) is the natural way to define a unique representation with a symmetric set of digits and an *even* base q. Thus, a part of the present thesis can be seen as a complement of [22].

The expected number of iterations of von Neumann's addition was analyzed in [72] and [59] for standard q-ary expansions and (q, d)-expansions, respectively. It turns out that the expected number of iterations is logarithmic in the length of the expansions. In [59], symmetric signed digit expansions are analyzed, too, but with a simplified probabilistic model since a precise probabilistic model exceeded computing resources available at that time. This simplification has a significant influence on the main term. In this thesis, we combine advances in soft- and hardware with sophisticated use of the finite state machine package of SageMath [96] to tackle the precise model in roughly 10 minutes of CPU time. The results include expectation, variance and convergence to a double exponential distribution.

In the sequel, we discuss the relation of our setting of sequences defined by automata and our results with the notion of q-regular sequences introduced in [1].

A sequence is q-regular if it is the first coordinate of a vector $\boldsymbol{v}(n)$ such that there exist matrices $V_0, \dots, V_{q-1}$ with

$$\boldsymbol{v}(qn + \varepsilon) = V_\varepsilon \boldsymbol{v}(n) \tag{1.1}$$

for $\varepsilon \in \{0, 1, \dots, q-1\}$.

While the output sum is a q-regular sequence for any transducer (see Remark 2.3.10), the converse is not necessarily true: Obviously, the sum of the output of a transducer reading the input n is always bounded by $\mathcal{O}(\log n)$. However, the 2-regular sequence[1]

$$a(n) = \begin{cases} n & \text{if } n \text{ is a power of 2,} \\ 0 & \text{otherwise} \end{cases}$$

can clearly not be bounded by $\mathcal{O}(\log n)$.

Thus, the concept of q-regular sequences is more general than our setting, but a broader variety of asymptotic behavior is observed which precludes any generalization of our results to general q-regular sequences.

Asymptotic estimates for q-regular sequences are given by Dumas [26, 27]. By restricting our attention to sequences defined by transducers, we obtain an asymptotic estimate of the variance, explicit expressions for the Fourier coefficients of the fluctuation in the second term of the expected value, non-differentiability of this fluctuation as well as a central limit theorem.

All chapters comprise computational aspects: First of all, we build a new transducer out of small known transducers by combining them accordingly such that the output sum of this transducer is the sequence we want to analyze. Then we compute certain values of this transducer to obtain the constants of the expected value, the variance and the Fourier coefficients of the periodic fluctuation. These computations involve determinants of matrices in several variables and partial sums of conditionally convergent series. These computations are implemented in the mathematical software system SageMath [97] using the finite state machine package described in the tutorial in Chapter 6 and [49]. This package is an integral part of SageMath, which allows reproducibility of our results. The code and its documentation are peer reviewed such that both meet the high quality standards of SageMath (see e.g. [48]).

[1] Use $\boldsymbol{v}(0) = (0, 1)^\top$ (where $^\top$ denotes transposition), $V_0 = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and $V_1 = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$.

CHAPTER 2

Output Sum of Transducers: Limiting Distribution and Periodic Fluctuation

We asymptotically analyze the output sum of a transducer. The input of the transducer is a random integer in the interval $[0, N)$ (or a higher dimensional analogue). Sequences defined by a certain class of recursions can be written in this framework.

Depending on properties of the transducer, the main term, the periodic fluctuation and an error term of the expected value and the variance of this sequence are established. The periodic fluctuation of the expected value is Hölder continuous and, in many cases, nowhere differentiable. A general formula for the Fourier coefficients of this periodic function is derived. Furthermore, it turns out that the sequence is asymptotically normally distributed for many transducers.

The construction of the transducers, the computations for the constants of the expectation, the variance and the Fourier coefficients can be done algorithmically by the mathematical software system SageMath [97]: The general framework and the code for the computation of the Fourier coefficients is included in SageMath 6.7 using its finite state machine package described in Chapter 6. The code for the construction from a recursion is submitted for inclusion in future versions of SageMath, see Ticket #17221.

This chapter corresponds to [55], which appeared in the *Electronic Journal of Combinatorics*. An extended abstract with less general Theorems 2.1, 2.2 and 2.4 and without proofs appeared as [52] in the proceedings of the *25th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithm*. This is joint work with Clemens Heuberger and Helmut Prodinger.

2.1. Introduction

For a transducer $\mathcal{T}$, let $\mathcal{T}(n)$ be the sum of the output labels of $\mathcal{T}$ when reading the q-ary expansion of n. For a positive integer N, we study the behavior of $\mathcal{T}(n)$ for a uniformly chosen random n in $\{0, \dots, N-1\}$. Assuming suitable connectivity properties of the underlying graph of the transducer, we obtain the following results.

- The expected value is given by
$$\mathbb{E}(\mathcal{T}(n)) = e_{\mathcal{T}} \log_q N + \Psi_1(\log_q N) + o(1)$$
for a constant $e_{\mathcal{T}}$ and a periodic, continuous function Ψ_1 (Theorem 2.1).
- The variance is
$$\mathbb{V}(\mathcal{T}(n)) = v_{\mathcal{T}} \log_q N - \Psi_1^2(\log_q N) + \Psi_2(\log_q N) + o(1)$$
with constant $v_{\mathcal{T}}$ and a periodic, continuous function $\Psi_2(x)$ (Theorem 2.1).
- After suitable renormalization, $\mathcal{T}(n)$ is asymptotically normally distributed (Theorem 2.1).

- The Fourier coefficients of Ψ_1 are explicitly given in Theorem 2.2 and the Fourier series converges absolutely and uniformly.
- The function Ψ_1 is nowhere differentiable provided that $e_{\mathcal{T}}$ is not an integer (Theorem 2.3).

The exact assumptions for the various results are given in detail in the respective theorems. Results for higher dimensional input are available for expectation, variance, normal distribution and Fourier coefficients.

Our theorems are generalizations of the following known results.

- For the sum of digits of the standard q-ary digit representations (cf. [20]), we obtain an asymptotic normal distribution, the Fourier coefficients and the non-differentiability (for even[1] q). The error term vanishes, as stated in Remark 2.3.4. Therefore, the formula is not only asymptotic but also exact. The formulas for the Fourier coefficients by Delange [20] also follow from our Theorem 2.2.
- The occurrence of subblocks in standard and non-standard digit representations is defined by a strongly connected, aperiodic transducer. Thus, we obtain the expected value, the variance, the limit law and the Fourier coefficients (cf. [35, 69, 71] for the expected value). For one dimensional digit representations, we also obtain the non-differentiability (assuming $e_{\mathcal{T}} \neq 0, 1$) of the fluctuation in the expectation.
- The Hamming weight is a special case of the occurrence of subblocks. Thus, Theorem 2.1 is a generalization of the results about the width-w non-adjacent form [50], the simple joint sparse form [36] and the asymmetric joint sparse form [50].
- A transducer defining a completely q-additive function consists of only one state. Therefore, we obtain an asymptotic normal distribution (as in [10]), the Fourier coefficients and the non-differentiability (assuming $e_{\mathcal{T}} \notin \mathbb{Z}$ and integer output). Here, the error term vanishes, too.
- A digital sequence is defined by a strongly connected, aperiodic transducer. Thus, digital sequences are asymptotically normally distributed or degenerate. Assuming $e_{\mathcal{T}} \notin \mathbb{Z}$ and integer output, the periodic fluctuation $\Psi_1(x)$ is non-differentiable. The Fourier coefficients can be computed by Theorem 2.2. See also [15] for results on the expected value.
- Automatic sequences [1] are also defined by transducers: The output labels of all transitions are 0 and the final output labels are as in the definition of such sequences. Theorem 2.1 gives the expected value with $e_{\mathcal{T}} = 0$ (see also [84]) and, depending on the transducer, also the variance with $v_{\mathcal{T}} = 0$. The Fourier coefficients of the periodic fluctuation of the expected value are given explicitly in Theorem 2.2.
- In [40], Grabner and Thuswaldner investigate the sum of digits function for negative bases $s_{-q}(n)$. They give a transducer to compute the function $s_{-q}(n) - s_{-q}(-n)$. Their result about the limit law follows directly from our Theorem 2.1.

While some of the examples can easily be formulated by transducers, other examples are more readily expressed in terms of recursions of the shape

$$a(q^\kappa n + \lambda) = a(q^{\kappa_\lambda} n + r_\lambda) + t_\lambda \quad \text{for} \quad 0 \leq \lambda < q^\kappa \tag{2.1}$$

with fixed κ, κ_λ, $r_\lambda \in \mathbb{Z}$, $t_\lambda \in \mathbb{R}$ and $\kappa_\lambda < \kappa$. We transform such a recursion into a transducer in Theorem 2.4 in Section 2.2.6.

[1] Our approach in Theorem 2.3 requires that the constant $e_{\mathcal{T}}$ of the main term of the expected value is not an integer. In this case, $e_{\mathcal{T}} = \frac{q-1}{2}$, which is an integer if q is odd.

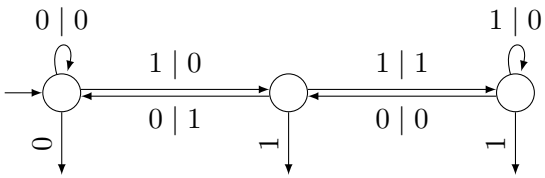


FIGURE 2.1. Transducer to compute the Hamming weight of the non-adjacent form.

As an example of a new result obtained by Theorem 2.1, we give an asymptotic estimate of the abelian complexity function of the paperfolding sequence in Example 2.2.8. In [74], the authors prove that this sequence satisfies a recursion of type (2.1). As consequences of Theorem 2.1, the expected value is $\sim \frac{8}{13}\log_2 N$, the variance is $\sim \frac{432}{2197}\log_2 N$ and the sequence is asymptotically normally distributed.

Section 2.2 contains all the theorems and the required notions. In Section 2.2.2, Theorem 2.1, formulas for the first and second moment of the output sum of a transducer and its limiting distribution are presented. In Theorem 2.2 in Section 2.2.4, the Fourier coefficients of the periodic fluctuation $\Psi_1(x)$ of the expected value are stated. We discuss the non-differentiability of $\Psi_1(x)$ in Theorem 2.3 in Section 2.2.5.

Section 2.2.6 deals with sequences satisfying the recursion (2.1) and higher dimensional analogues. We construct a transducer computing this sequence in Theorem 2.4. Thus, from Theorem 2.1, the expected value, the variance and the limit distribution follow in many cases.

In Sections 2.3 to 2.6, we give the proofs of all the theorems from Section 2.2.

2.2. Results

This section starts with the definition of some notions about the connectivity of a transducer. Then we will state the theorems about the moments and the limiting distribution, the Fourier coefficients, the non-differentiability, and the construction of a transducer computing a sequence given by a recursion as in (2.1).

2.2.1. Notions. We consider complete, deterministic and subsequential transducers (cf. [12, Chapter 1]). In our case, the input alphabet is $\{0,\dots,q-1\}^d$ for a positive integer d and the output alphabet $\mathbb{R}$. A transducer is said to be *deterministic* and *complete* if for every state and every digit of the input alphabet, there is exactly one transition starting in this state with this input label. A *subsequential* transducer $\mathcal{T}$ (cf. [91]) is defined to be a finite deterministic automaton with one initial state, an output label for every transition and a final output label for every state.

Figure 2.1 presents an example of a complete, deterministic, subsequential transducer. The label of a transition with input ε and output δ is written as $\varepsilon \mid \delta$.

The input of the transducer is the standard q-ary joint digit representation of an integer vector $\boldsymbol{n} \in \mathbb{N}_0^d$, i.e. the standard q-ary digit representation at each coordinate of the vector $\boldsymbol{n}$. The input is read from right (least significant digit) to left (most significant digit), without leading zeros. Then the output of the transducer is the sequence of the outputs of the transitions along the unique path starting in the initial state with the given input and the final output of the last state of this path. The element $\mathcal{T}(\boldsymbol{n})$ of the sequence defined by the transducer $\mathcal{T}$ is the sum of this output sequence.

Using final output labels is convenient for our purposes. Clearly, it would also be possible to model the final output labels by using an "end-of-input" marker and additional transitions. In the context of digital expansions, the behavior can usually also be obtained by reading a sufficient number of leading zeros. But the approach using final outputs is more general as it is not required that the final outputs are compatible with the output generated by leading zeros.

For the various results, different properties of the complete, deterministic, subsequential transducer and its underlying digraph are needed. All states of the underlying digraph are assumed to be accessible from the initial state. Contracting each strongly connected component of the underlying digraph gives an acyclic digraph, the so-called condensation. A strongly connected component is said to be *final strongly connected* if it corresponds to a leaf (i.e., a vertex with out-degree 0) in the condensation. Let c be the number of final strongly connected components. We call a transducer or a digraph *finally connected* if $c = 1$.

For the asymptotic expressions, only the final strongly connected components are important. All other strongly connected components only influence the error term. Thus, we are not interested in the periodicity of the whole underlying digraph, but in the periodicity of the final strongly connected components. The *period* of a digraph is defined as the greatest common divisor of all lengths of directed cycles of the digraph. For $j = 1, \dots, c$, let p_j be the period of the final strongly connected component C_j. Define the *final period* of the digraph as

$$p = \operatorname{lcm}\{p_j \mid j = 1, \dots, c\}.$$

We call a digraph *finally aperiodic* if $p = 1$. If the underlying digraph is strongly connected, its final period is equal to its period.

For proving the non-differentiability of the fluctuation, we not only need a finally aperiodic, finally connected digraph ($p = c = 1$), but also a reset sequence. A *reset sequence* is an input sequence such that starting at any state and reading this sequence leads to a specific state s. If the transducer is not finally aperiodic and finally connected, then there cannot exist a reset sequence.

2.2.2. Moments and Limiting Distribution. This section contains the theorem about the moments of the output sum $\mathcal{T}(\boldsymbol{n})$ and the limiting distribution. Further results about the periodic fluctuation can be found in Theorems 2.2 and 2.3. The proof follows the ideas in [36], also used and extended in [50].

As probability space, we use $\Omega_N = \{0, 1, \dots, N-1\}^d$ endowed with the equidistribution measure.

Denote by Φ_{μ,σ^2} the cumulative distribution function of the normal distribution with mean μ and variance $\sigma^2 \neq 0$. Thus,

$$\Phi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\Big(-\frac{1}{2}\Big(\frac{y-\mu}{\sigma}\Big)^2\Big)\, dy.$$

Theorem 2.1. *Let $d \geq 1$, $\mathcal{T}$ be a complete, deterministic, subsequential transducer with input alphabet $\{0, 1, \dots, q-1\}^d$, output alphabet $\mathbb{R}$, final period p, and c final components.*

Then $\mathcal{T}(\boldsymbol{n})$ has the expected value

$$\mathbb{E}(\mathcal{T}(\boldsymbol{n})) = e_{\mathcal{T}} \log_q N + \Psi_1(\log_q N) + \mathcal{O}(N^{-\xi} \log N) \tag{2.2}$$

where the constants $e_{\mathcal{T}}$ and $\xi > 0$ are given in (2.4) *in Section 2.2.3 and $\Psi_1(x)$ is a p-periodic, Hölder continuous function.*

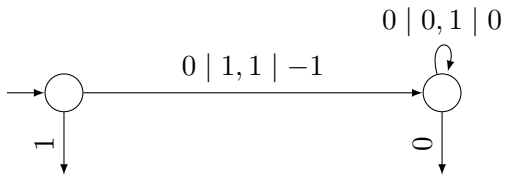


FIGURE 2.2. Transducer of Remark 2.2.1.

If all b_j given in (2.4) *are positive, the distribution function of $\mathcal{T}(\boldsymbol{n})$ can be approximated by a mixture of c Gaussian distributions with weights λ_j, means $a_j \log_q N$ and variances $b_j \log_q N$ for some constants a_j and $\lambda_j > 0$ with $\sum_{j=1}^{c} \lambda_j = 1$, given in* (2.4)*. In particular,*

$$\mathbb{P}\Big(\frac{\mathcal{T}(\boldsymbol{n})}{\sqrt{\log_q N}} \le x\Big) = \sum_{j=1}^{c} \lambda_j \Phi_{a_j\sqrt{\log_q N}, b_j}(x) + \mathcal{O}\big(\log^{-\frac{1}{2}} N\big)$$

for all $x \in \mathbb{R}$.

If all a_j are equal, then $\mathcal{T}(\boldsymbol{n})$ has the variance

$$\mathbb{V}(\mathcal{T}(\boldsymbol{n})) = v_{\mathcal{T}} \log_q N - \Psi_1^2(\log_q N) + \Psi_2(\log_q N) + \mathcal{O}(N^{-\xi} \log^2 N) \tag{2.3}$$

with constant $v_{\mathcal{T}} \in \mathbb{R}$ (given in (2.4)*) and a p-periodic, continuous function $\Psi_2(x)$. Otherwise, the variance is $\mathbb{V}(\mathcal{T}(\boldsymbol{n})) = \Theta(\log^2 N)$.*

If all a_j are equal, $\mathcal{T}(\boldsymbol{n})$ converges in distribution to a mixture of Gaussian (or degenerate) distributions with means 0 and variances b_j, weighted by λ_j. In particular, if all $b_j > 0$,

$$\mathbb{P}\Big(\frac{\mathcal{T}(\boldsymbol{n}) - \mathbb{E}(\mathcal{T}(\boldsymbol{n}))}{\sqrt{\log_q N}} \le x\Big) = \sum_{j=1}^{c} \lambda_j \Phi_{0,b_j}(x) + \mathcal{O}\big(\log^{-\frac{1}{2}} N\big)$$

holds for all $x \in \mathbb{R}$.

If furthermore $c = 1$ and $v_{\mathcal{T}} \neq 0$, then $\mathcal{T}(\boldsymbol{n})$ is asymptotically normally distributed.

We give the proof of this theorem in Section 2.3.

Remark 2.2.1. The assumption that $b_j > 0$ is essential for obtaining uniform convergence of the distribution function and the speed of convergence in particular. To see this, consider the transducer in Figure 2.2. It is easily seen that $\mathcal{T}(n) = (-1)^n$. For even N, the distribution function of $\mathcal{T}(n)/\sqrt{\log_2 N}$ is given by

$$\mathbb{P}\Big(\frac{\mathcal{T}(n)}{\sqrt{\log_2 N}} \le x\Big) = \begin{cases} 0 & \text{if } x < -1/\sqrt{\log_2 N}, \\ 1/2 & \text{if } -1/\sqrt{\log_2 N} \le x < 1/\sqrt{\log_2 N}, \\ 1 & \text{if } 1/\sqrt{\log_2 N} \le x, \end{cases}$$

which does not converge uniformly.

2.2.3. Eigenvalues and Eigenvectors of the Transition Matrix. For the constants in Theorem 2.1 and the Fourier coefficients in Theorem 2.2, we need the notion of a transition matrix of the transducer and properties of its eigenvalues and eigenvectors.

We label the states of the transducer with contiguous positive integers starting with 1. We denote the indicator vector of the initial state by $\boldsymbol{e}_1$.

Definition 2.2.2. Let $t \in \mathbb{R}$ be in a neighborhood of 0.

The transition matrix $M_{\boldsymbol{\varepsilon}}$ for $\boldsymbol{\varepsilon} \in \{0, \dots, q-1\}^d$ is the matrix whose (s_1, s_2)-th entry is $e^{it\delta}$ if there is a transition from state s_1 to state s_2 with input label $\boldsymbol{\varepsilon}$ and output label δ, and 0 otherwise.

Let M be the sum of all these transition matrices.

Lemma 2.2.3. *There are differentiable functions $\mu_j(t)$ in a neighborhood of $t = 0$ for $j = 1, \dots, c$ such that the dominant eigenvalues of M are $\mu_j(t)\exp(\frac{2\pi i l}{p})$ in this neighborhood of $t = 0$ for some of the $l \in \mathcal{P} = \{k \in \mathbb{Z} \mid -p/2 < k \le p/2\}$. For each of these dominant eigenvalues, the algebraic and geometric multiplicities coincide. For $t = 0$, $\mu_j(0) = q^d$.*

The proof of this lemma is given in Section 2.3.

Let $l \in \mathbb{Z}$. Consider the (not necessarily orthogonal) projection onto the direct sum of the left eigenspaces of M corresponding to the eigenvalues $\mu_j(t)\exp(\frac{2\pi i l}{p})$ for $j = 1, \dots, c$ such that the kernel is the direct sum of the remaining generalized left eigenspaces. Let $\boldsymbol{w}_l^\top(t)$ be the image of $\boldsymbol{e}_1^\top$ under this projection, where $^\top$ denotes transposition. The definition of $\boldsymbol{w}_l^\top(t)$ only depends on l modulo p.

We write $\boldsymbol{w}_l^\top$ for $\boldsymbol{w}_l^\top(0)$ and $\boldsymbol{w}_l'^\top$ for the derivative of $\boldsymbol{w}_l^\top(t)$ at $t = 0$. Furthermore, $\boldsymbol{w}_l^\top$ is either the null vector or a left eigenvector of M corresponding to the eigenvalue $q^d\exp(\frac{2\pi i l}{p})$.

Let C_j be a final component with corresponding indicator vector $\boldsymbol{c}_j$. Define the constants

$$\lambda_j = \boldsymbol{w}_0^\top \boldsymbol{c}_j.$$

In Section 2.3.1, we will show that $\lambda_j > 0$ and $\sum_{j=1}^c \lambda_j = 1$.

With these definitions, the constants in Theorem 2.1 can be expressed as

$$\begin{aligned} a_j &= -iq^{-d}\mu_j'(0), \\ e_{\mathcal{T}} &= \sum_{j=1}^c \lambda_j a_j, \\ b_j &= \frac{\mu_j'(0)^2 - q^d\mu_j''(0)}{q^{2d}}, \\ v_{\mathcal{T}} &= \sum_{j=1}^c \lambda_j b_j. \end{aligned} \tag{2.4}$$

Finally, $\xi > 0$ is chosen such that all non-dominant eigenvalues of M have modulus strictly less than $q^{d-\xi}$ at $t = 0$.

These constants can be interpreted as follows: $a_j \log_q N$ and $b_j \log_q N$ are the main terms of the mean and the variance, respectively, of the output sum of the final component C_j. These expressions including the derivatives of the eigenvalues correspond to the formulas for mean and variance given in [30, Theorem IX.9]. The constants $e_{\mathcal{T}}$ and $v_{\mathcal{T}}$ are convex combinations of the corresponding constants of the final components C_j.

The positive weight λ_j in these convex combinations turns out to be the asymptotic probability of reaching the final component C_j. This is connected to the following interpretation of the left eigenvector $\boldsymbol{w}_0^\top$: If the final period p is 1, the entries of $\boldsymbol{w}_0^\top$ will be shown to be the asymptotic probabilities of reaching the corresponding states. This corresponds to the left eigenvector used in a steady-state analysis. If $p > 1$, these probabilities depend on the length of the input modulo p. Then, we will prove that $\boldsymbol{w}_0^\top$ gives the average of these probabilities taken over all residues modulo p. These interpretations are justified in Section 2.3.1.

2.2.4. Fourier Coefficients. This section contains the formulas for the Fourier coefficients of the periodic fluctuation $\Psi_1(x)$. For this purpose, we need the following definitions.

Let $\chi_k = \frac{2\pi i k}{p \log q}$ for $k \in \mathbb{Z}$ and $\mathbf{1}$ be a vector whose entries are all one.

The s-th coordinate of the vector $\boldsymbol{b}(\boldsymbol{n})$ is the sum of the output of the transducer $\mathcal{T}$ (including the final output) if starting in state s with input the q-ary joint expansion of $\boldsymbol{n}$. In particular, the first coordinate of $\boldsymbol{b}(\boldsymbol{n})$ is $\mathcal{T}(\boldsymbol{n})$, and $\boldsymbol{b}(0)$ is the vector of final outputs. Furthermore, define the vector-valued function $\boldsymbol{H}(z)$ by the Dirichlet series

$$\boldsymbol{H}(z) = \sum_{\substack{\boldsymbol{n}\geq 0\\ \boldsymbol{n}\neq 0}} \boldsymbol{b}(\boldsymbol{n}) \|\boldsymbol{n}\|_\infty^{-z}, \tag{2.5}$$

where the inequality in the summation index is considered coordinate-wise and $\|\cdot\|_\infty$ is the maximum norm.

Theorem 2.2. *Let $\mathcal{T}$ be a subsequential, complete, deterministic transducer. Then the Fourier coefficients of the p-periodic fluctuation $\Psi_1(x)$ are*

$$\begin{aligned} c_0 &= -\frac{e_{\mathcal{T}}}{d \log q} - i\boldsymbol{w}_0'^{\top}\mathbf{1} + \frac{1}{d}\operatorname{Res}_{z=d} \boldsymbol{w}_0^{\top}\boldsymbol{H}(z),\\ c_k &= \frac{1}{d+\chi_k}\operatorname{Res}_{z=d+\chi_k} \boldsymbol{w}_k^{\top}\boldsymbol{H}(z) \end{aligned} \tag{2.6}$$

for $k \neq 0$.

The Fourier series $\sum_{k\in\mathbb{Z}} c_k \exp(\frac{2\pi i k}{p} x)$ converges absolutely and uniformly.

The function $\boldsymbol{w}_k^{\top}\boldsymbol{H}(z)$ is meromorphic in $\Re z > d-1$. It has a possible double pole at $z = d$ for $k = 0$ and possible simple poles at $z = d + \chi_k$ for $k \neq 0$.

The proof of this theorem is in Section 2.4.

The infinite recursion given in Lemma 2.4.5 can be used to numerically evaluate the Dirichlet series $\boldsymbol{H}(z)$ with arbitrary precision and to compute its residues at $z = d + \chi_l$ (see Lemma 2.4.7 and [39]). For $d = 1$, the computation of the Fourier coefficients can be done by the mathematical software system SageMath [97].

Example 2.2.4. The (artificial) transducer in Figure 2.3 has two final components with periods 2 and 3, respectively. Thus the final period is 6 and the function $\Psi_1(x)$ is 6-periodic. The constant $e_{\mathcal{T}}$ of the expected value is $\frac{11}{8}$. In Figure 2.4, the partial Fourier series with 2550 Fourier coefficients[2] is compared with the empirical values of the periodic fluctuation Ψ_1, i.e.,

$$\frac{1}{N}\sum_{n<N} \mathcal{T}(n) - \frac{11}{8}\log_2 N \tag{2.7}$$

with integers N and $4 \leq \log_2 N \leq 16$.

The computation of these 2550 Fourier coefficients took less than 6 minutes using a standard dual-core PC.

In Example 2.2.8 we compute the first 2550 Fourier coefficients of the abelian complexity function of the paperfolding sequence.

As a corollary of Theorem 2.2, we obtain the following result which was already proved by Delange [20].

[2]We use 2550 Fourier coefficients in this plot because the period length of the next summand of the Fourier series in Figure 2.4 is already less than the resolution of a standard printer.

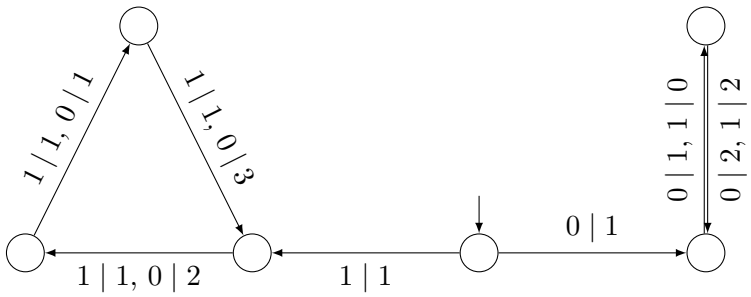


FIGURE 2.3. Transducer of Example 2.2.4: All states are final with final output 0.

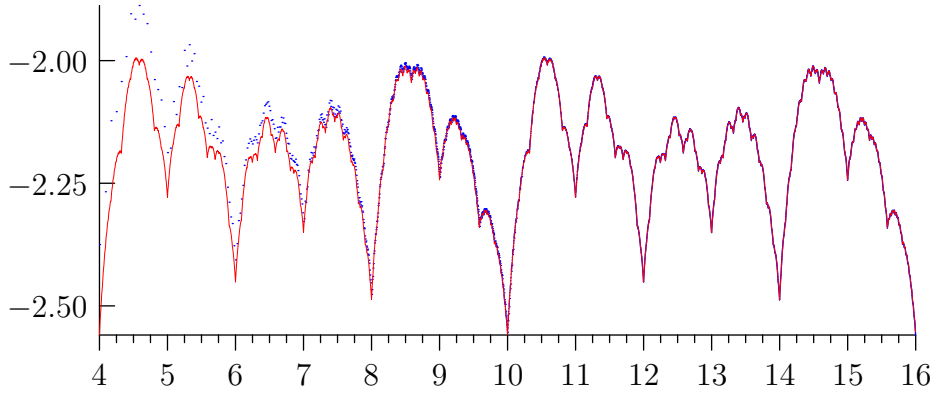


FIGURE 2.4. Partial Fourier series compared with the empirical values of the function Ψ_1 of Example 2.2.4.

Corollary 2.2.5. *The Fourier coefficients of the periodic fluctuation*

$$\Psi_1(\log_q N) = \frac{1}{N}\sum_{n<N} s_q(n) - \frac{q-1}{2}\log_q N$$

for the q-ary sum-of-digits function $s_q(n)$ are

$$\begin{aligned} c_0 &= \frac{q-1}{2\log q}(\log(2\pi)-1) - \frac{q+1}{4}, \\ c_k &= -\frac{q-1}{\chi_k(1+\chi_k)\log q}\zeta(\chi_k) \end{aligned} \tag{2.8}$$

for $k \neq 0$ and $\chi_k = \frac{2\pi i k}{\log q}$ where ζ denotes the Riemann ζ-function.

We prove this corollary in Section 2.4.

2.2.5. Non-differentiability. In this section, we prove that for certain transducers, the periodic fluctuation $\Psi_1(x)$ of the expected value is nowhere differentiable.

Theorem 2.3. *Let $d = 1$. Assume that $e_{\mathcal{T}} \notin \mathbb{Z}$ and that the transducer $\mathcal{T}$ has a reset sequence and output alphabet $\mathbb{Z}$. Then the function $\Psi_1(x)$ is non-differentiable for any $x \in \mathbb{R}$.*

The proof can be found in Section 2.5. There, we follow the method presented by Tenenbaum [98], see also Grabner and Thuswaldner [40].

In [40, 98], the reset sequence consists only of 0's. If working with digit expansions, it is often possible to choose such a reset sequence. However, in the context of recursions, this is not always possible, see Example 2.2.8. There the reset sequence is (00001).

For a general finally aperiodic, finally connected transducer, the existence of a reset sequence cannot be guaranteed.

2.2.6. Recursions. In this section, we describe how to reduce a recursion to a transducer computing the given sequence. All inequalities in this section are considered coordinate-wise.

Let $q \geq 2$, κ, $\kappa_{\boldsymbol{\lambda}} \in \mathbb{Z}$, $\boldsymbol{r}_{\boldsymbol{\lambda}} \in \mathbb{Z}^d$, $t_{\boldsymbol{\lambda}} \in \mathbb{R}$ and $0 \leq \kappa_{\boldsymbol{\lambda}} < \kappa$ for $0 \leq \boldsymbol{\lambda} < q^\kappa \mathbf{1}$. If $d \geq 2$, then additionally let $\boldsymbol{r}_{\boldsymbol{\lambda}} \geq 0$ for all $\boldsymbol{\lambda}$.

Consider the sequence $a(\boldsymbol{n})$, $\boldsymbol{n} \in \mathbb{N}_0^d$, defined by the recursion

$$a(q^\kappa \boldsymbol{n} + \boldsymbol{\lambda}) = a(q^{\kappa_{\boldsymbol{\lambda}}} \boldsymbol{n} + \boldsymbol{r}_{\boldsymbol{\lambda}}) + t_{\boldsymbol{\lambda}} \quad \text{for} \quad 0 \leq \boldsymbol{\lambda} < q^\kappa \mathbf{1} \tag{2.9}$$

and for all integer vectors $\boldsymbol{n}$ such that the arguments on both sides are non-negative. Furthermore, initial values $a(\boldsymbol{n})$ for $\boldsymbol{n} \in \mathcal{I}$ have to be given for a suitable finite set $\mathcal{I} \subset \mathbb{N}_0^d$.

It must be ensured that the recursion (2.9) does not lead to conflicts and that the set of $\mathcal{I}$ is appropriate. Additionally, we require that $\mathcal{I}$ is minimal (with respect to inclusion). In that case, we say that the recursion is well-posed.

In Section 2.6, we construct a subsequential, complete, deterministic transducer $\mathcal{T}$ (also when the recursion is not well-posed) reading the q-ary joint expansion of integer vectors without leading zeros. We will define a distinguished subset of its states, called *simple states*. Furthermore, disjoint classes $F_1, \ldots, F_K$ of integer vectors will be defined.

Theorem 2.4. *The recursion* (2.9) *is well-posed if and only if*

(1) for each cycle consisting of simple states with transitions with zero input label, the sum of its output transitions vanishes and
(2) the set $\mathcal{I}$ consists of one representative of each F_j, $1 \leq j \leq K$.

In that case, the sum of the output of $\mathcal{T}$ is the sequence a, i.e., $\mathcal{T}(\boldsymbol{n}) = a(\boldsymbol{n})$ for all $\boldsymbol{n} \geq 0$.

The proof of this theorem is in Section 2.6. Combining this result with Theorem 2.1 yields an asymptotic analysis of the sequence $a(\boldsymbol{n})$, as in Example 2.2.8. Moreover, this asymptotic analysis can be performed algorithmically in SageMath for $d = 1$ (using the code submitted at `http://trac.sagemath.org/17221`). A combinatorial description of the sets F_i involving an auxiliary transducer is given in Remark 2.6.1.

Remark 2.2.6. For $d \geq 2$, and $r_{\boldsymbol{\lambda}} \not\geq 0$, the sequence cannot be computed by a finite transducer: For every $j \geq 0$, there are non-zero integer vectors $\boldsymbol{n} \geq 0$, $\boldsymbol{n}' \geq 0$ with $\boldsymbol{n} \equiv \boldsymbol{n}' \pmod{q^j}$—i.e., a finite deterministic transducer cannot distinguish between $\boldsymbol{n}$ and $\boldsymbol{n}'$—such that the recursion (2.9) can be applied for the argument $q^\kappa \boldsymbol{n} + \boldsymbol{\lambda}$ but cannot be applied for $q^\kappa \boldsymbol{n}' + \boldsymbol{\lambda}$.

This problem does not arise in the case of dimension $d = 1$: if the end of the input is not yet reached (this is something the transducer knows), there is a guaranteed forthcoming digit ≥ 1 (instead of $\neq 0$ in the higher dimensional case). This information is enough to decide whether the recursion can be used.

Remark 2.2.7. Suppose that the given sequence is defined for $\boldsymbol{n} \geq \boldsymbol{n}_0$ for some constant $\boldsymbol{n}_0$. Then the sequence $b(\boldsymbol{n}) = a(\boldsymbol{n} + \boldsymbol{n}_0)$ fulfills (2.9) with $\kappa_{\boldsymbol{\lambda}}$, $\boldsymbol{r}_{\boldsymbol{\lambda}}$ and $t_{\boldsymbol{\lambda}}$ replaced by $\kappa_{\boldsymbol{\mu}}$, $q^{\kappa_{\boldsymbol{\mu}}} \boldsymbol{s} + \boldsymbol{r}_{\boldsymbol{\mu}} - \boldsymbol{n}_0$ and $t_{\boldsymbol{\mu}}$, respectively, where $\boldsymbol{n}_0 + \boldsymbol{\lambda} = q^\kappa \boldsymbol{s} + \boldsymbol{\mu}$ for $0 \leq \boldsymbol{\mu} < q^\kappa \mathbf{1}$. Then Theorem 2.4 can be applied.

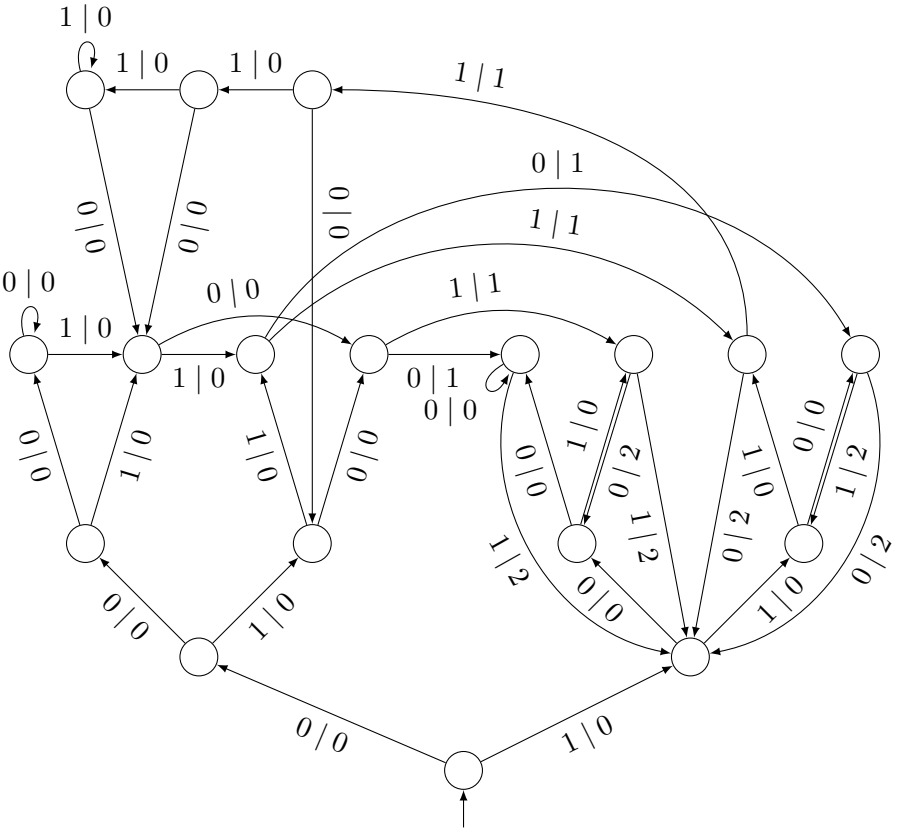


FIGURE 2.5. Transducer to compute the abelian complexity function $\rho(n)$ of the paperfolding sequence. For simplicity, the final output labels are omitted.

Example 2.2.8 ([85])**.** Consider the abelian complexity function $\rho(n)$ of the paperfolding sequence. The paperfolding sequence is obtained by repeatedly folding a strip of paper in half in the same direction. Then we open the strip and encode a right turn by 1 and a left turn by 0. The abelian complexity function $\rho(n)$ gives the number of abelian equivalence classes of subwords of length n of the paperfolding sequence. Two subwords of length n are equivalent if they are permutations of each other. In [74], the authors prove that this sequence satisfies the recursion

$$\begin{aligned}
\rho(4n) &= \rho(2n),\\
\rho(4n+2) &= \rho(2n+1)+1,\\
\rho(16n+1) &= \rho(8n+1),\\
\rho(16n+3) &= \rho(2n+1)+2,\\
\rho(16n+5) &= \rho(4n+1)+2,\\
\rho(16n+7) &= \rho(2n+1)+2,\\
\rho(16n+9) &= \rho(2n+1)+2,\\
\rho(16n+11) &= \rho(4n+3)+2,
\end{aligned}$$

l	c_l	l	c_l
0	1.5308151288	12	$-0.0002297481 + 0.0009687657\,i$
1	$-0.0162585750 + 0.0478637218\,i$	13	$0.0006425378 + 0.0006516706\,i$
2	$0.0054521982 + 0.0075023586\,i$	14	$0.0000413217 - 0.0003867709\,i$
3	$-0.0028294724 + 0.0086495903\,i$	15	$-0.0005632948 - 0.0001843541\,i$
4	$0.0036818110 + 0.0021908312\,i$	16	$0.0009051717 - 0.0000476354\,i$
5	$-0.0028244495 + 0.0014519078\,i$	17	$-0.0004621780 - 0.0000594551\,i$
6	$-0.0008962222 + 0.0030512180\,i$	18	$-0.0000127264 - 0.0003100798\,i$
7	$0.0015033904 + 0.0013217107\,i$	19	$0.0004112716 + 0.0001954204\,i$
8	$-0.0006766166 - 0.0015392566\,i$	20	$-0.0000011706 + 0.0004183253\,i$
9	$0.0016074870 - 0.0000503663\,i$	21	$-0.0001027596 + 0.0004091624\,i$
10	$-0.0006908394 + 0.0018753575\,i$	22	$-0.0004725451 + 0.0004237489\,i$
11	$-0.0008974336 + 0.0007658455\,i$	23	$-0.0000596181 + 0.0002323317\,i$

TABLE 2.1. First 24 Fourier coefficients of the abelian complexity function $\rho(n)$ of the paperfolding sequence.

$$\rho(16n+13) = \rho(2n+1) + 2,$$
$$\rho(16n+15) = \rho(2n+2) + 1$$

with $\rho(1) = 2$ and $\rho(0) = 0$. The constructed transducer is shown in Figure 2.5. For simplicity, we do not state the final output labels in this figure. The expected value and the variance are

$$\mathbb{E}(\rho(n)) = \frac{8}{13}\log_2 N + \Psi_1(\log_2 N) + \mathcal{O}(N^{-\xi}\log N),$$
$$\mathbb{V}(\rho(n)) = \frac{432}{2197}\log_2 N - \Psi_1^2(\log_2 N) + \Psi_2(\log_2 N) + \mathcal{O}(N^{-\xi}\log^2 N)$$

with $0 < \xi < 0.5604267891$, as the second largest eigenvalues of the transition matrix are $-0.7718445063 \pm 1.1151425080\,i$. The sequence $\rho(n)$ is asymptotically normally distributed. The functions $\Psi_1(x)$ and $\Psi_2(x)$ are 1-periodic and continuous. The reset sequence of the transducer is (00001) (reading from right to left). The function $\Psi_1(x)$ is nowhere differentiable and its Fourier series converges absolutely and uniformly. The first 24 Fourier coefficients of $\Psi_1(x)$ are listed in Table 2.1. In Figure 2.6, the trigonometric polynomial formed with the first 2550 Fourier coefficients is compared with the empirical values of the function $\Psi_1(x)$ (see (2.7)).

2.3. Asymptotic Distribution—Proof of Theorem 2.1

This section contains some lemmas which will together imply Theorem 2.1. Our plan is as follows: First, we give auxiliary lemmas about the eigenvalues and eigenvectors of the transition matrix M in Section 2.3.1. Section 2.3.2 contains an asymptotic formula for the characteristic function of the random variable $\mathcal{T}(\boldsymbol{n})$. We use this characteristic function to give formulas for the expected value and the variance in Section 2.3.3, and prove the continuity of the periodic fluctuations in Section 2.3.4. Finally, we prove the central limit theorem in Section 2.3.5.

We use the notation $(\boldsymbol{\varepsilon}_L \ldots \boldsymbol{\varepsilon}_0)_q$ for the standard q-ary joint digit representation of an integer vector with $\boldsymbol{\varepsilon}_L \neq 0$. For a real number in the interval $[0, q)$, we write $(\varepsilon_0.\varepsilon_1 \ldots)_q$

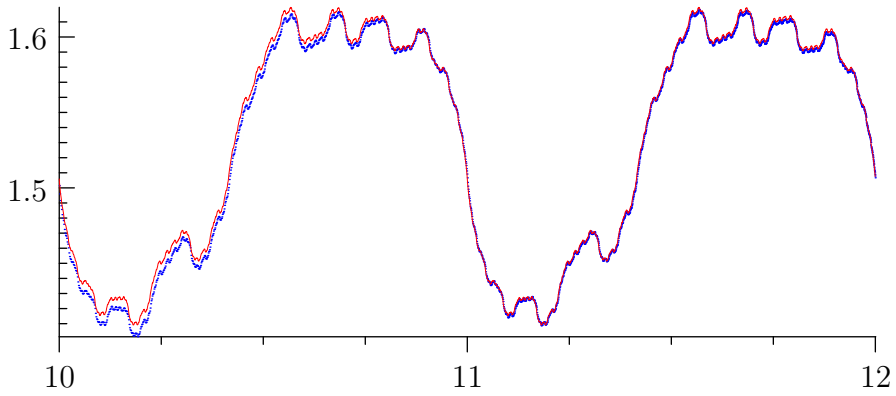


FIGURE 2.6. Partial Fourier series compared with the empirical values of $\Psi_1(x)$ of the abelian complexity function of the paperfolding sequence.

for the q-ary digit representation choosing the representation ending on 0^ω in the case of ambiguity. Furthermore, we use Iverson's notation [41]: [*expression*] is 1 if *expression* is true and 0 otherwise. All $\mathcal{O}$-constants depend only on q, d and the number of states.

2.3.1. Transition Matrix and its Eigenvectors. This section contains the proofs of some results on the eigenvalues, eigenvectors and eigenprojections of the transition matrix M.

For the proof of Theorem 2.1, we use the following lemma which describes the eigenvalues of a matrix in a similar way as the Perron–Frobenius theorem (cf. [33]).

Lemma 2.3.1. *Let M be a matrix with complex entries whose underlying directed graph is p-periodic and strongly connected. Then the set of non-zero eigenvalues of M can be partitioned into disjoint sets of cardinality p where each set is invariant under multiplication by $e^{2\pi i/p}$ and all eigenvalues in one set have the same algebraic multiplicities.*

PROOF. Since the underlying directed graph of $M \in \mathbb{C}^{n\times n}$ is a strongly connected, p-periodic graph, we can write M as

$$M = \begin{pmatrix} 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \ddots & A_3 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & \ddots & A_p \\ A_1 & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

with block matrices A_i by reordering the vertices. Then $M - xI$ is the product of the matrices

$$\begin{pmatrix} -xI & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -xI & 0 \\ A_1 & \frac{1}{x}\prod_{j=1}^{2} A_j & \cdots & \frac{1}{x^{p-2}}\prod_{j=1}^{p-1} A_j & \frac{1}{x^{p-1}}\prod_{j=1}^{p} A_j - xI \end{pmatrix}$$

and

$$\begin{pmatrix} I & -\frac{1}{x}A_2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & I & -\frac{1}{x}A_p \\ 0 & \cdots & \cdots & 0 & I \end{pmatrix}.$$

Let $h(x)$ be the characteristic polynomial of $\prod_{j=1}^{p} A_j \in \mathbb{C}^{m\times m}$. Thus the characteristic polynomial of M is $x^{n-m-(p-1)m}h(x^p)$. Therefore, the eigenvalues of M are either 0 or any p-th root of a non-zero eigenvalue of $\prod_{j=1}^{p} A_j$. □

With this lemma, we can prove Lemma 2.2.3 about the eigenvalues of the matrix M:

Proof of Lemma 2.2.3. First, consider the case $t = 0$. By construction, q^d is an eigenvalue with right eigenvector $\mathbf{1}$ of M. As $\|M\|_\infty \le q^d$, where $\|\cdot\|_\infty$ denotes the row sum norm, q^d is a dominant eigenvalue.

Consider the strongly connected components of the underlying graph of $\mathcal{T}$. Each final strongly connected component C_j induces a final transducer $\mathcal{T}_j$ which is strongly connected, complete, deterministic and p_j-periodic. Thus, the adjacency matrix at $t = 0$ of this final transducer has a dominant eigenvalue q^d with right eigenvector $\mathbf{1}$. By the Perron–Frobenius theorem (cf. [33, Theorem 8.8.1]), all dominant eigenvalues of this final transducer are $\{q^d e^{2\pi il/p} \mid l \in \mathcal{P} \text{ with } p \mid lp_j\}$, each with algebraic and geometric multiplicity one.

A non-final strongly connected component induces a transducer $\mathcal{S}$ with the adjacency matrix S. This transducer is not complete. Let $\mathcal{S}^+$ be the complete transducer where loops are added to states of $\mathcal{S}$ where necessary. The adjacency matrix of $\mathcal{S}^+$ is S^+. Since $\mathcal{S}^+$ is complete, deterministic and strongly connected, $\rho(S^+) = q^d$. As $S \le S^+$ but $S \ne S^+$, Theorem 8.8.1 in [33] implies $\rho(S) < \rho(S^+) = q^d$.

Thus, the dominant eigenvalues are $q^d e^{2\pi il/p}$ with an $l \in \mathcal{P}$ such that there exists a $j \in \{1, \dots, c\}$ with $p \mid lp_j$. We determine the geometric multiplicities of these dominant eigenvalues of M in Lemma 2.3.2.

Now, fix a final strongly connected component C_j and some $l \in \mathcal{P}$ with $p \mid lp_j$. In a small neighborhood of $t = 0$, let $\mu_{lj}(t)$ be the eigenvalue of the submatrix of M corresponding to the complete transducer $\mathcal{T}_j$ with $\mu_{lj}(0) = q^d e^{2\pi il/p}$. Because of Lemma 2.3.1 applied to the final component C_j separately, we have $\mu_{lj}(t) = e^{2\pi il/p}\mu_j(t)$ where $\mu_j(t)$ is defined to be $\mu_{0j}(t)$.

All other moduli of eigenvalues of M are less than $\min_{l,j} |\mu_{lj}(t)|$ because of the continuity of eigenvalues.

We prove the differentiability of the eigenvalues in Lemma 2.3.2. □

Lemma 2.3.2. *Let $\mu_j(t)\exp(\frac{2\pi il}{p})$ be a dominant eigenvalue of the matrix M. There exists a corresponding left eigenvector of M with zero entries except in coordinates corresponding to the final component C_j.*

At $t = 0$, the algebraic and geometric multiplicities of $q^d \exp(\frac{2\pi il}{p})$ coincide.

Furthermore the eigenvalues and the eigenprojection corresponding to the eigenvalues $\mu_j \exp(\frac{2\pi il}{p})$ are analytic at $t = 0$.

Proof. Let $q^d \exp(\frac{2\pi il}{p})$ be a dominant eigenvalue of M. Its algebraic multiplicity at $t = 0$ is $|\{j : p \mid lp_j\}|$. We construct exactly one left eigenvector in the neighborhood of

$t = 0$ for each final component C_j with $p \mid lp_j$: Let $\mathcal{T}_j$ be the induced transducer of the final component C_j. Let $\tilde{\boldsymbol{v}}^\top(t)$ be a left eigenvector of the adjacency matrix of $\mathcal{T}_j$ corresponding to the eigenvalue $\mu_j(t)\exp(\frac{2\pi il}{p})$. As the algebraic multiplicity is 1 in this final component, the choice of $\tilde{\boldsymbol{v}}^\top(t)$ is unique up to multiplication with a scalar function in t. Then, we construct the left eigenvector $\boldsymbol{v}^\top(t)$ by padding $\tilde{\boldsymbol{v}}^\top(t)$ with zeros.

These left eigenvectors are linearly independent because of the block structure induced by the final components. Thus the geometric and the algebraic multiplicities of $q^d\exp(\frac{2\pi il}{p})$ coincide.

Furthermore, $\mu_j(t)\exp(\frac{2\pi il}{p})$ is a simple eigenvalue of the adjacency matrix of $\mathcal{T}_j$. Therefore, [68, Chapter II] implies the differentiability of the eigenvalues and eigenprojections. □

From now on we use the convention that the eigenspace corresponding to $\mu_j(t)\exp(\frac{2\pi il}{p})$ is the null space if $\mu_j(t)\exp(\frac{2\pi il}{p})$ is not an eigenvalue. Then its eigenprojection is the constant null function.

Definition 2.3.3. Let $\boldsymbol{w}_{lj}^\top(t)$ be the eigenprojection of $\boldsymbol{e}_1^\top$ onto the left eigenspace corresponding to the possible eigenvalue $\mu_j(t)\exp(\frac{2\pi il}{p})$. The vector $\boldsymbol{w}_{lj}^\top(t)$ is thus a null vector or a left eigenvector of M corresponding to the eigenvalue $\mu_j(t)\exp(\frac{2\pi il}{p})$.

Define

$$\boldsymbol{w}^\top(t) = \boldsymbol{e}_1^\top - \sum_{l\in\mathcal{P}}\sum_{j=1}^{c}\boldsymbol{w}_{lj}^\top(t).$$

As an abbreviation, we write $\boldsymbol{w}_{lj}^\top$, $\boldsymbol{w}^\top$, $\boldsymbol{w}_{lj}'^\top$ and $\boldsymbol{w}'^\top$ for these projections and their derivatives at $t = 0$.

Remark 2.3.4. If there are only dominant eigenvalues, then $\boldsymbol{w}^\top(t) = 0$. This will imply that there is no error term in the asymptotic expansion of the expected value and the variance. This occurs in the case of the sum of digits of the standard q-ary digit representation and other completely q-additive functions because the transducer has only one state.

Lemma 2.3.5. *In a fixed neighborhood of $t = 0$, let $\xi > 0$ be as defined in (2.4), i.e., all non-dominant eigenvalues have modulus less than $q^{d-\xi}$. Then*

$$\Big\|\frac{d^k}{dt^k}\boldsymbol{w}^\top(t)M^m\Big\| = \mathcal{O}(c_k^{(1)}q^{(d-\xi)(m-k)}m^k)$$

for m, $k \geq 0$ and a constant $c_k^{(1)}$.

Proof. Let P be the matrix such that $x^\top \mapsto x^\top P$ is the sum of the eigenprojections onto the left eigenspaces corresponding to $\mu_j\exp(\frac{2\pi il}{p})$ for $j = 1, \ldots, c$ and $l \in \mathcal{P}$. Then $\boldsymbol{w}^\top = \boldsymbol{e}_1^\top(I - P)$ and

$$\boldsymbol{w}^\top M^m = \boldsymbol{e}_1^\top((I - P)M)^m.$$

As the spectral radius of $(I - P)M$ is less than $q^{d-\xi}$, we obtain the stated estimates. □

With $\boldsymbol{w}_l^\top$ defined in Section 2.2.3, we have

$$\boldsymbol{w}_l^\top(t) = \sum_{j=1}^{c}\boldsymbol{w}_{lj}^\top(t). \tag{2.10}$$

Note that left and right eigenvectors corresponding to different eigenvalues annihilate each other. Because of the block structure of the eigenvectors in Lemma 2.3.2 and because $\mathbf{1}$ is a right eigenvector to q^d, we have

$$[l = 0]\lambda_j = \boldsymbol{w}_{lj}^\top \mathbf{1} \tag{2.11}$$

where λ_j is defined in Section 2.2.3. Furthermore, $\boldsymbol{w}^\top \mathbf{1} = 0$ and

$$\sum_{j=1}^{c} \lambda_j = \sum_{l \in \mathcal{P}} \sum_{j=1}^{c} \boldsymbol{w}_{lj}^\top \mathbf{1} + \boldsymbol{w}^\top \mathbf{1} = \boldsymbol{e}_1^\top \mathbf{1} = 1.$$

Denote by $\boldsymbol{\delta}$ the vector whose s-th component is the sum of the outputs of all transitions leaving the state s. By the definition of the transition matrix $M(t)$, $\boldsymbol{\delta}$ can be expressed as

$$i\boldsymbol{\delta} = \frac{d}{dt} M(t)\mathbf{1}\Big|_{t=0}. \tag{2.12}$$

We now establish a relation between $\boldsymbol{\delta}$, the left eigenvector $\boldsymbol{w}_l^\top$ and its derivative at $t = 0$. By definition of the left eigenvectors $\boldsymbol{w}_{lj}^\top(t)$ and (2.10),

$$\boldsymbol{w}_l^\top(t) M \mathbf{1} = \sum_{j=1}^{c} \mu_j(t) \exp\Big(\frac{2\pi i l}{p}\Big) \boldsymbol{w}_{lj}^\top(t)\mathbf{1}.$$

Differentiation, (2.11), (2.4) and (2.10) yield

$$\boldsymbol{w}_l^\top \boldsymbol{\delta} = [l = 0] e_{\mathcal{T}} q^d - q^d (e^{\frac{2\pi i l}{p}} - 1) i \boldsymbol{w}_l'^\top \mathbf{1}. \tag{2.13}$$

To establish the interpretation of $\boldsymbol{w}_0^\top$ given at the end of Section 2.2.3, we consider

$$\hat{\boldsymbol{w}}_k^\top := \lim_{m \to \infty} \boldsymbol{e}_1^\top M^{mp+k} q^{-d(mp+k)},$$

the stationary distribution on the state space of all states of the transducer under the assumption that the input length is congruent to k modulo p. Using (2.10) and Lemma 2.3.5 yields

$$\begin{aligned}
\hat{\boldsymbol{w}}_k^\top &= \lim_{m \to \infty} \Big(\sum_{l \in \mathcal{P}} \boldsymbol{w}_l^\top + \boldsymbol{w}^\top \Big) M^{mp+k} q^{-d(mp+k)} \\
&= \lim_{m \to \infty} \sum_{l \in \mathcal{P}} \exp\Big(\frac{2\pi i l k}{p}\Big) \boldsymbol{w}_l^\top + \mathcal{O}(q^{-\xi(mp+k)}) \\
&= \sum_{l \in \mathcal{P}} \exp\Big(\frac{2\pi i l k}{p}\Big) \boldsymbol{w}_l^\top.
\end{aligned}$$

Summation leads to $\frac{1}{p} \sum_{k=0}^{p-1} \hat{\boldsymbol{w}}_k^\top = \boldsymbol{w}_0^\top$. Thus, λ_j is the hitting probability of the final component C_j when starting in the initial state. As every state is accessible from the initial state, λ_j is positive.

Finally, for $l = 0$, (2.13) reads $q^{-d} \boldsymbol{w}_0^\top \boldsymbol{\delta} = e_{\mathcal{T}}$, which can be interpreted as the steady state analysis of the expectation: the probability distribution $\boldsymbol{w}_0^\top$ is multiplied with the expected output $q^{-d}\boldsymbol{\delta}$.

2.3.2. Characteristic Function. To obtain a central limit law in Section 2.3.5, we compute an asymptotic formula for the characteristic function in this section.

The next lemma can be proved by induction on L. It is a generalization of Lemma 3 in [50].

Lemma 2.3.6. *Let A_ε, $\varepsilon = 0, \dots, q-1$ be matrices in $\mathbb{C}^{n\times n}$, $H_\varepsilon \colon \mathbb{N}_0 \to \mathbb{C}^{n\times n}$ be known functions with $H_0(0) = 0$. Let $G\colon \mathbb{N}_0 \to \mathbb{C}^{n\times n}$ be a function which satisfies the recurrence relation*

$$G(qN+\varepsilon) = A_\varepsilon G(N) + H_\varepsilon(N)$$

for $N \geq 0$, $\varepsilon \in \{0, \dots, q-1\}$ and $G(0) = 0$. Then

$$G\left((\varepsilon_L \dots \varepsilon_0)_q\right) = \sum_{l=0}^{L} \Big(\prod_{i=0}^{l-1} A_{\varepsilon_i} \Big) H_{\varepsilon_l}\left((\varepsilon_L \dots \varepsilon_{l+1})_q\right).$$

The solution of this recursion finally leads to an asymptotic formula for the characteristic function.

We choose the branch $-\pi + \frac{\pi}{p} < \arg z \leq \pi + \frac{\pi}{p}$ of the complex logarithm. After setting $t = 0$, we use only the logarithm of complex numbers for which our branch coincides the principal branch $-\pi < \arg z \leq \pi$.

Lemma 2.3.7. *The characteristic function of the random variable $\mathcal{T}(\boldsymbol{n})$ is*

$$\mathbb{E}(\exp(it\mathcal{T}(\boldsymbol{n}))) = \frac{1}{N^d} \sum_{l\in\mathcal{P}} \sum_{j=1}^{c} \mu_j(t)^{\log_q N} \exp\Big(\frac{2\pi i l \log_q N}{p}\Big) \Psi_{lj}(\log_q N, t) + R(N,t)$$

with functions $\Psi_{lj}(x,t)$ (defined in (2.23)), which are arbitrarily often differentiable in t and 1-periodic in x, and an error term $R(N,t)$. This error term $R(N,t)$ is arbitrarily often differentiable, too, and satisfies $\frac{d^k}{dt^k} R(N,t) = \mathcal{O}(c_k^{(2)} N^{-\xi} \log^k N)$, for $k \geq 0$, a constant $c_k^{(2)}$ and the constant $\xi > 0$ defined in Section 2.2.3, in a neighborhood of $t = 0$. At $t = 0$, we have $R(N,0) = 0$.

Proof. For a transducer $\mathcal{T}$, consider the characteristic function

$$F(N) = \mathbb{E}(\exp(it\mathcal{T}(n))) = \frac{1}{N^d} \sum_{\boldsymbol{n}\in\Omega_N} e^{it\mathcal{T}(\boldsymbol{n})} \tag{2.14}$$

of the discrete random variable $\mathcal{T}(\boldsymbol{n})$.

Then the summands in (2.14) can be expressed as a matrix product

$$e^{it\mathcal{T}(\boldsymbol{n})} = \boldsymbol{e}_1^\top \prod_{l=0}^{L} M_{\boldsymbol{\varepsilon}_l} \boldsymbol{u}$$

where $(\boldsymbol{\varepsilon}_L \dots \boldsymbol{\varepsilon}_0)_q$ is the standard q-ary joint digit representation of $\boldsymbol{n}$ with $\boldsymbol{\varepsilon}_L \neq 0$ and the vector $\boldsymbol{u}$ has entries $e^{itb(s)}$ where $b(s)$ is the final output of the state s. Again, the vector $\boldsymbol{e}_1$ is the indicator vector of the initial state.

Let

$$g(\boldsymbol{n}) = \prod_{l=0}^{L} M_{\boldsymbol{\varepsilon}_l}$$

and

$$G(N) = \sum_{\boldsymbol{n}\in\Omega_N} g(\boldsymbol{n}),$$

hence

$$F(N) = \frac{1}{N^d} \boldsymbol{e}_1^\top G(N) \boldsymbol{u}. \tag{2.15}$$

The function $g(\boldsymbol{n})$ satisfies the recursion

$$g(q\boldsymbol{n} + \boldsymbol{\varepsilon}) = M_{\boldsymbol{\varepsilon}} g(\boldsymbol{n}) \tag{2.16}$$

for $\boldsymbol{\varepsilon} \in \{0, 1, \dots, q-1\}^d$, $\boldsymbol{n} \geq 0$ with $q\boldsymbol{n} + \boldsymbol{\varepsilon} \neq 0$.

We define further functions

$$G_C(N) = \sum_{\substack{0 \leq n_i < N \\ i \notin C}} \sum_{\substack{n_i = N \\ i \in C}} g(\boldsymbol{n}) \tag{2.17}$$

where the coordinates $n_1, \dots, n_d$ of $\boldsymbol{n}$ with indices in the set $C \subseteq \{1, \dots, d\}$ are fixed to N. This yields $G(N) = G_\emptyset(N)$. Furthermore, we define the matrices

$$M^\varepsilon_{C,D} = \sum_{\substack{\beta_i = 0 \\ i \notin C \cup D}}^{q-1} \sum_{\substack{\beta_i = 0 \\ i \in D}}^{\varepsilon - 1} \sum_{\substack{\beta_i = \varepsilon \\ i \in C}} M_{\boldsymbol{\beta}} \tag{2.18}$$

for disjoint sets C, $D \subseteq \{1, \dots, d\}$ and $\varepsilon \in \{0, 1, \dots, q-1\}$. In this definition, we restrict the i-th coordinate β_i of $\boldsymbol{\beta}$ to be ε or less than ε if $i \in C$ or $i \in D$, respectively. Otherwise, the i-th coordinate can be arbitrary. Then, $M = M^\varepsilon_{\emptyset,\emptyset}$ holds independently of ε.

Then, (2.16) yields the following recursions for $G_C(N)$, $\varepsilon = 0, \dots, q-1$, $N \geq 0$ and $C \neq \{1, \dots, d\}$:

$$\begin{aligned} G_C(qN + \varepsilon) &= \sum_{\substack{\beta_i = 0 \\ i \notin C}}^{q-1} \sum_{\substack{\beta_i = \varepsilon \\ i \in C}} \sum_{\substack{0 \leq q m_i + \beta_i < qN + \varepsilon \\ i \notin C}} \sum_{\substack{q m_i + \beta_i = qN + \varepsilon \\ i \in C}} g(q\boldsymbol{m} + \boldsymbol{\beta}) \\ &= [C = \emptyset \wedge qN + \varepsilon \neq 0](I - M_0) \\ &\quad + \sum_{\substack{\beta_i = 0 \\ i \notin C}}^{q-1} \sum_{\substack{\beta_i = \varepsilon \\ i \in C}} M_{\boldsymbol{\beta}} \sum_{\substack{0 \leq m_i < N + \frac{\varepsilon - \beta_i}{q} \\ i \notin C}} \sum_{\substack{m_i = N \\ i \in C}} g(\boldsymbol{m}) \\ &= [C = \emptyset \wedge qN + \varepsilon \neq 0](I - M_0) + \sum_{D \subseteq C^c} M^\varepsilon_{C,D} G_{C \cup D}(N). \end{aligned} \tag{2.19}$$

This recursion for G_C only depends on $G_{C'}$ for $C' \supsetneq C$. As

$$G_{\{1,\dots,d\}}(N) = g(N\mathbf{1}),$$

we can recursively determine G_C using Lemma 2.3.6. In particular, for $G(N)$, this yields the recursion formula

$$G(qN + \varepsilon) = MG(N) + H_\varepsilon(N) \tag{2.20}$$

for $N \geq 0$, $\varepsilon \in \{0, \dots q-1\}$ where H_ε are known functions with

$$H_\varepsilon(N) = [qN + \varepsilon \neq 0](I - M_0) + \sum_{\emptyset \neq D \subseteq \{1,\dots,d\}} M^\varepsilon_{\emptyset,D} G_D(N). \tag{2.21}$$

Thus by Lemma 2.3.6, we get

$$G((\varepsilon_L \dots \varepsilon_0)_q) = \sum_{m=0}^{L} M^m H_{\varepsilon_m}((\varepsilon_L \dots \varepsilon_{m+1})_q). \tag{2.22}$$

By construction, $\|M_{\boldsymbol{\varepsilon}}\|_\infty = 1$ for every $\boldsymbol{\varepsilon} \in \{0, \dots, q-1\}^d$. We conclude that $\|M^{\varepsilon}_{C,D}\|_\infty \le q^{d-|C|-|D|}\varepsilon^{|D|}$. By the definition of $G_C(N)$, the growth rates of the functions $G_C(N)$ and $H_\varepsilon(N)$ are $\|G_C(N)\|_\infty = \mathcal{O}(N^{d-|C|})$ and $\|H_\varepsilon(N)\|_\infty = \mathcal{O}(N^{d-1})$, respectively. For $k \ge 0$, the k-th derivative of $H_\varepsilon(N)$ at $t = 0$ can be bounded by $\mathcal{O}(c_k^{(3)} N^{d-1} \log^k N)$ for a constant $c_k^{(3)}$.

We define

$$R(N,t) = \frac{1}{N^d} \sum_{m=0}^{L} \boldsymbol{w}^\top M^m H_{\varepsilon_m}((\varepsilon_L \dots \varepsilon_{m+1})_q)\boldsymbol{u},$$

which constitutes an explicit expression for the error term contributed by the non-dominant eigenvalues. By Lemma 2.3.5, its derivatives satisfy

$$\frac{d^k}{dt^k} R(N,t) = \mathcal{O}(c_k^{(2)} N^{-\xi} \log^k N)$$

for $k \ge 0$. Because $\boldsymbol{u}(0) = \mathbf{1}$ and left and right eigenvectors corresponding to different eigenvalues annihilate each other, we have $R(N,0) = 0$.

By (2.15), (2.22) and $\boldsymbol{e}_1^\top = \sum_{l\in\mathcal{P}} \sum_{j=1}^{c} \boldsymbol{w}_{lj}^\top + \boldsymbol{w}^\top$,

$$\begin{aligned} F(N) &= \frac{1}{N^d} \sum_{l\in\mathcal{P}} \sum_{j=1}^{c} \mu_j^L \exp\Big(\frac{2\pi i l L}{p}\Big) \\ &\quad \times \sum_{m=0}^{L} \mu_j^{m-L} \exp\Big(\frac{2\pi i l (m-L)}{p}\Big) \boldsymbol{w}_{lj}^\top H_{\varepsilon_m}((\varepsilon_L \dots \varepsilon_{m+1})_q)\boldsymbol{u} \\ &\quad + R(N,t) \\ &= \frac{1}{N^d} \sum_{l\in\mathcal{P}} \sum_{j=1}^{c} \mu_j^{\log_q N} \exp\Big(\frac{2\pi i l \log_q N}{p}\Big) \Psi_{lj}(\log_q N, t) + R(N,t) \end{aligned}$$

with

$$\Psi_{lj}(x,t) = \mu_j(t)^{-\{x\}} \exp\Big(-\frac{2\pi i l \{x\}}{p}\Big) \sum_{m=0}^{\infty} \mu_j(t)^{-m} \exp\Big(-\frac{2\pi i l m}{p}\Big) \boldsymbol{w}_{lj}^\top H_{x_m}((x_0 \dots x_{m-1})_q)\boldsymbol{u} \tag{2.23}$$

and $q^{\{x\}} = (x_0.x_1 \dots)_q$, choosing the representation ending on 0^ω in the case of ambiguity.

The functions $\Psi_{lj}(x,t)$ are periodic in x with period 1 and well defined for all $x \in \mathbb{R}$ since they are dominated by geometric series. Furthermore, they are arbitrarily often differentiable in t. □

2.3.3. Moments. In this section we give the moments of the output sum $\mathcal{T}(\boldsymbol{n})$.

Lemma 2.3.8. *The expected value and the variance of $\mathcal{T}(n)$ are as stated in Theorem 2.1 with constants given in* (2.4) *and periodic functions given in Lemma 2.3.9 and* (2.27).

PROOF. The derivative of $\mathbb{E}(\exp(it\mathcal{T}(\boldsymbol{n})))$ with respect to t at $t=0$ gives the expected value of the sum of the output of the transducer

$$\mathbb{E}(\mathcal{T}(n)) = \frac{1}{N^d}\sum_{\boldsymbol{n}\in\Omega_N}\mathcal{T}(n) = \Psi_0(\log_q N)\log_q N + \Psi_1(\log_q N) + \mathcal{O}(N^{-\xi}\log N)$$

with p-periodic functions

$$\begin{aligned}\Psi_0(x) &= \sum_{l\in\mathcal{P}}\sum_{j=1}^{c} a_j e^{\frac{2\pi i l x}{p}}\Psi_{lj}(x,0),\\ \Psi_1(x) &= -i\sum_{l\in\mathcal{P}}\sum_{j=1}^{c} e^{\frac{2\pi i l x}{p}}\Psi'_{lj}(x,0)\end{aligned} \tag{2.24}$$

and constants a_j defined in (2.4). Here, Ψ'_{lj} denotes the derivative with respect to t.

We now compute $\Psi_0(x)$ for some x with $q^{\{x\}} = (x_0.x_1\ldots)_q$. To compute $H_\varepsilon(N)$, we use (2.20) and the definition of $G(N)$ to obtain

$$H_\varepsilon(N)\mathbf{1} = ((qN+\varepsilon)^d - (qN)^d)\mathbf{1} \tag{2.25}$$

for $t=0$, because $\mathbf{1}$ is a right eigenvector of $M_{\boldsymbol{\varepsilon}}$ for every $\boldsymbol{\varepsilon}$. Together with (2.23), this results in

$$\Psi_{lj}(x,0) = q^{-d\{x\}}\exp\Big(-\frac{2\pi i l\{x\}}{p}\Big)\boldsymbol{w}_{lj}^{\top}\mathbf{1}D\Big(q^d e^{\frac{2\pi i l}{p}}\Big)$$

with

$$D(z) = \sum_{m=0}^{\infty} z^{-m}((x_0\ldots x_m)_q^d - (x_0\ldots x_{m-1}0)_q^d).$$

By (2.11), we have $\Psi_{lj}(x,0)=0$ for $l\neq 0$.

To compute $D(q^d)$, observe that

$$D(q^d) = \sum_{m=0}^{\infty}\big((x_0.x_1\ldots x_m)_q^d - (x_0.x_1\ldots x_{m-1})_q^d\big) = \lim_{m\to\infty}(x_0.x_1\ldots x_m)_q^d = q^{d\{x\}}$$

because $D(q^d)$ is a telescoping sum.

We conclude that

$$\Psi_{lj}(x,0) = \lambda_j[l=0] \tag{2.26}$$

and therefore

$$\Psi_0(x) = \sum_{j=1}^{c} a_j\lambda_j = e_{\mathcal{T}}$$

by (2.4). This completes the proof of the expectation as given in (2.2).

Using Lemma 2.3.7 and (2.26), the second derivative of $\mathbb{E}(\exp(it\mathcal{T}(n)))$ gives

$$\begin{aligned}\frac{1}{N^d}\sum_{\boldsymbol{n}\in\Omega_N}\mathcal{T}(n)^2 = \log_q^2 N\sum_{j=1}^{c} a_j^2\lambda_j + v_{\mathcal{T}}\log_q N\\ - 2i\log_q N\sum_{l\in\mathcal{P}}\sum_{j=1}^{c} a_j\exp\Big(\frac{2\pi i l\log_q N}{p}\Big)\Psi'_{lj}(\log_q N,0)\\ + \Psi_2(\log_q N) + \mathcal{O}(N^{-\xi}\log^2 N)\end{aligned}$$

with $v_{\mathcal{T}}$ given in (2.4) and

$$\Psi_2(x) = -\sum_{l\in\mathcal{P}}\sum_{j=1}^{c} e^{\frac{2\pi i l x}{p}}\Psi''_{lj}(x,0). \tag{2.27}$$

Here, Ψ''_{lj} denotes the second derivative with respect to t. Thus, by (2.2), the variance is

$$\begin{aligned}\mathbb{V}(\mathcal{T}(n)) &= \frac{1}{N^d}\sum_{\boldsymbol{n}\in\Omega_N}\mathcal{T}(n)^2 - \Big(\frac{1}{N^d}\sum_{\boldsymbol{n}\in\Omega_N}\mathcal{T}(n)\Big)^2\\ &= \Big(\sum_{j=1}^{c} a_j^2\lambda_j - e_{\mathcal{T}}^2\Big)\log_q^2 N\\ &\quad + \Big(v_{\mathcal{T}} - 2i\sum_{l\in\mathcal{P}}\sum_{j=1}^{c} a_j \exp\Big(\frac{2\pi i l\log_q N}{p}\Big)\Psi'_{lj}(\log_q N,0)\\ &\qquad - 2e_{\mathcal{T}}\Psi_1(\log_q N)\Big)\log_q N\\ &\quad + \Psi_2(\log_q N) - \Psi_1^2(\log_q N) + \mathcal{O}(N^{-\xi}\log^2 N).\end{aligned} \tag{2.28}$$

By Jensen's inequality, the coefficient of $\log_q^2 N$ is zero if and only if all a_j are equal. If all a_j are equal, then the coefficient of $\log_q N$ in (2.28) simplifies by (2.24), too, and we obtain (2.3). □

For the computation of the Fourier coefficients and the proof of the Hölder condition, we need an explicit expression for Ψ_1.

In analogy to the definition of G_C in (2.17), define

$$\boldsymbol{B}_C(N) = \sum_{\substack{0\le n_i<N\\ i\notin C}}\sum_{\substack{n_i=N\\ i\in C}}\boldsymbol{b}(\boldsymbol{n}) \tag{2.29}$$

for $C\subseteq\{1,\dots,d\}$.

Lemma 2.3.9. *For $q^{\{x\}} = (x_0.x_1\dots)_q$, the fluctuation $\Psi_1(x)$ can be expressed as*

$$\Psi_1(x) = -e_{\mathcal{T}}\{x\} - q^{-d\{x\}}\sum_{l\in\mathcal{P}}\sum_{m=0}^{\infty} q^{-dm}e^{\frac{2\pi i l}{p}(\lfloor x\rfloor - m)}f_l((x_0\dots x_m)_q) \tag{2.30}$$

with

$$\begin{aligned}f_l(r) &= [l=0]e_{\mathcal{T}}\big(\lfloor\log_q r\rfloor(r^d - (q\lfloor rq^{-1}\rfloor)^d) + (q\lfloor rq^{-1}\rfloor)^d\big)\\ &\quad + i\boldsymbol{w}_l'^{\top}\mathbf{1}\Big(r^d - \exp\Big(\frac{2\pi i l}{p}\Big)(q\lfloor rq^{-1}\rfloor)^d\Big)\\ &\quad - \boldsymbol{w}_l^{\top}\boldsymbol{B}_{\emptyset}(r) + q^d\exp\Big(\frac{2\pi i l}{p}\Big)\boldsymbol{w}_l^{\top}\boldsymbol{B}_{\emptyset}(\lfloor rq^{-1}\rfloor).\end{aligned} \tag{2.31}$$

The estimate $f_l(r) = \mathcal{O}(r^{d-1}\log r)$ holds.

Proof. From (2.24), (2.23), (2.4), (2.26) and (2.11) and the absolute convergence of Ψ_{lj}, we obtain (2.30) with

$$f_l(r) = [l=0]e_{\mathcal{T}}\lfloor\log_q r\rfloor(r^d - (q\lfloor rq^{-1}\rfloor)^d) + i\,\frac{d}{dt}\boldsymbol{w}_l^{\top}(t)H_{r \bmod q}(\lfloor rq^{-1}\rfloor)\boldsymbol{u}(t)\Big|_{t=0}.$$

From the combinatorial interpretation of $\boldsymbol{b}(\boldsymbol{n})$ and $g(\boldsymbol{n})\boldsymbol{u}(t)$, we obtain

$$i\boldsymbol{b}(\boldsymbol{n}) = \frac{d}{dt} g(\boldsymbol{n})\boldsymbol{u}(t)\Big|_{t=0}, \tag{2.32}$$

in analogy to (2.12). As the range of summation of G_C and $\boldsymbol{B}_C$ coincides, we immediately get

$$i\boldsymbol{B}_C(N) = \frac{d}{dt} G_C(N)\boldsymbol{u}(t)\Big|_{t=0}. \tag{2.33}$$

By (2.25) and by differentiating $H_\varepsilon(N)\boldsymbol{u}(t)$ using (2.20), (2.33) and (2.12),

$$\begin{aligned} f_l(r) = [l=0]e_{\mathcal{T}}\lfloor \log_q r\rfloor (r^d - (q\lfloor rq^{-1}\rfloor)^d) + i\boldsymbol{w}_l'^{\top}\mathbf{1}(r^d - (q\lfloor rq^{-1}\rfloor)^d) \\ - \boldsymbol{w}_l^{\top}(\boldsymbol{B}_\emptyset(r) - M\boldsymbol{B}_\emptyset(\lfloor rq^{-1}\rfloor) - \lfloor rq^{-1}\rfloor^d \boldsymbol{\delta}). \end{aligned}$$

The fact that $\boldsymbol{w}_l^{\top}$ is a left eigenvector of M and (2.13) establish (2.31).

For the growth estimate of $f_l(r)$, we use the explicit definition of H_ε in (2.21), (2.33) and the trivial estimate $\|\boldsymbol{b}(\boldsymbol{n})\| = \mathcal{O}(\log\|\boldsymbol{n}\|)$. □

To formulate $\mathcal{T}(\boldsymbol{n})$ as a q-regular sequence, we first define output vectors. The s-th entry of the vector $\boldsymbol{\delta}_\varepsilon$ is the output label of the transition from state s with input label $\boldsymbol{\varepsilon}$. By (2.16), (2.32), and

$$\frac{d}{dt} M_\varepsilon \mathbf{1}\Big|_{t=0} = i\boldsymbol{\delta}_\varepsilon, \tag{2.34}$$

we have

$$\boldsymbol{b}(q\boldsymbol{n}+\boldsymbol{\varepsilon}) = M_\varepsilon \boldsymbol{b}(\boldsymbol{n}) + \boldsymbol{\delta}_\varepsilon. \tag{2.35}$$

Remark 2.3.10. We can use the matrices

$$V_\varepsilon = \begin{pmatrix} M_\varepsilon & \boldsymbol{\delta}_\varepsilon & [\boldsymbol{\varepsilon}=0]I \\ 0 & 1 & 0 \\ 0 & 0 & [\boldsymbol{\varepsilon}=0]I \end{pmatrix}$$

and $\boldsymbol{v}(\boldsymbol{n}) = (\boldsymbol{b}(\boldsymbol{n}), 1, [\boldsymbol{n}=0](\boldsymbol{b}(0) - M_0\boldsymbol{b}(0) - \boldsymbol{\delta}_0))^{\top}$ in the definition of a q-regular sequence (1.1) to realize that the output sum of a transducer is q-regular. If $d > 1$, then this is a multidimensional q-regular sequence (cf. [1]).

2.3.4. Hölder Continuity. In this section, we prove the continuity of the fluctuations Ψ_1 and Ψ_2 as well as the Hölder continuity of Ψ_1. This will be used to establish the convergence of the Fourier series. The following lemma is an extension of the result in [50].

Lemma 2.3.11. *The functions $\Psi_1(x)$ and, if all a_j are equal, $\Psi_2(x)$ are continuous for $x \in \mathbb{R}$.*

Proof. First note that continuity of Ψ_1 for $x \in \mathbb{R}$ with $x = \log_q y$ where y has no finite q-ary expansion follows from the definitions (2.23) and (2.24). To prove it for $x = \log_q y$ with $0 \le x < p$ where y has a finite q-ary expansion, observe that the two one-sided limits exist due to the definition. Next, we prove that they are the same. Consider the two integer sequences $N_k = yq^{pk}$ and $\tilde{N}_k = N_k - 1$ for k large enough such that N_k is an integer. For a real number z, we write $\{z\}_p = p\{z/p\}$ for the unique real number in the interval $[0, p)$ such that $z - \{z\}_p$ is an integer multiple of p.

This yields

$$\lim_{k\to\infty} \{\log_q N_k\}_p = \lim_{k\to\infty} \{\log_q y + pk\}_p = \{x\}_p = \lim_{z\to x^+} \{z\}_p,$$

$$\lim_{k\to\infty}\{\log_q \tilde{N}_k\}_p = \lim_{k\to\infty}\{\log_q N_k + \log_q(1-N_k^{-1})\}_p$$
$$= \lim_{k\to\infty}\{x + \log_q(1-N_k^{-1})\}_p = \lim_{z\to x^-}\{z\}_p.$$

If we insert the two sequences N_k and $\tilde{N}_k$ in

$$\sum_{\boldsymbol{n}\in\Omega_N} \mathcal{T}(\boldsymbol{n}) = e_{\mathcal{T}} N^d \log_q N + N^d \Psi_1(\log_q N) + \mathcal{O}(N^{d-\xi}\log N)$$

(cf. (2.2)) and take the difference, we get

$$\mathcal{O}(N_k^{d-1}\log N_k) = N_k^d \Psi_1(\log_q N_k) - \tilde{N}_k^d \Psi_1(\log_q \tilde{N}_k) + \mathcal{O}(N_k^{d-\xi}\log N_k).$$

Because $\Psi_1(x)$ is bounded by a geometric series by definition, we have

$$\Psi_1(\log_q N_k) - \Psi_1(\log_q \tilde{N}_k) = O(N_k^{-\xi}\log N_k)$$

and in particular

$$\lim_{k\to\infty}\Psi_1(\{\log_q N_k\}_p) = \lim_{k\to\infty}\Psi_1(\{\log_q \tilde{N}_k\}_p).$$

Therefore, Ψ_1 is continuous in x.

The continuity of $\Psi_2(x)$ at $x = \log_q(y)$ for y with infinite q-ary expansion again follows from the definition of Ψ_2. If all a_j are equal, the continuity of the fluctuation $-\Psi_1^2 + \Psi_2$ of the variance (2.3) follows as above, where $\log N_k$ has to be replaced by $\log^2 N_k$ in the error terms. Thus Ψ_2 is also continuous in this case. □

Lemma 2.3.12 ([46])**.** *The function Ψ_1 satisfies a Hölder condition of order α for all $\alpha \in (0,1)$.*

Proof. Let $0 < \alpha < 1$ be any constant. We want to prove that there exists a positive constant C such that

$$|\Psi_1(y) - \Psi_1(x)| \le C|y-x|^\alpha \tag{2.36}$$

holds for all x, $y \in \mathbb{R}$.

For $x = y$, the left-hand side of (2.36) is 0 and the inequality is obviously satisfied. From now on, assume that $x < y$. By the periodicity of Ψ_1, it is sufficient to prove (2.36) for $0 \le x < p$.

First, we prove (2.36) for the case $0 \le x < y$ and sufficiently small $y - x < 1$.

Fix such x and y and choose the integer k such that

$$q^{-k-1} \le |q^y - q^x| < q^{-k}.$$

Note that the continuous differentiability of $z \mapsto q^z$ on the compact interval $[0, p+1]$ implies that $q^y - q^x = \mathcal{O}(|y-x|)$ and therefore

$$q^{-k} = \mathcal{O}(|y-x|). \tag{2.37}$$

We prove (2.36) in three steps.

Statement 2.3.13. *Let a, $b \in \mathbb{R}$ with $x \le a < b \le y$ and $\lfloor a \rfloor = \lfloor b \rfloor$ such that the first $k+1$ digits of the expansions*

$$q^{\{a\}} = (a_0.a_1\ldots)_q, \quad q^{\{b\}} = (b_0.b_1\ldots)_q$$

coincide, i.e., $a_i = b_i$ for $0 \le i \le k$. Then

$$|\Psi_1(b) - \Psi_1(a)| = \mathcal{O}(|y-x|^\alpha).$$

PROOF. Lemma 2.3.9 yields

$$\begin{aligned}|\Psi_1(b)-\Psi_1(a)| &\le |e_{\mathcal{T}}||\{b\}-\{a\}|\\ &\quad + q^{-d\{b\}}\sum_{l\in\mathcal{P}}\sum_{m\ge 0} q^{-dm}|f_l((b_0\dots b_m)_q)-f_l((a_0\dots a_m)_q)|\\ &\quad + |q^{-d\{b\}}-q^{-d\{a\}}|\sum_{l\in\mathcal{P}}\sum_{m\ge 0} q^{-dm}|f_l((a_0\dots a_m)_q)|\\ &\le |e_{\mathcal{T}}||\{b\}-\{a\}|\\ &\quad + \sum_{l\in\mathcal{P}}\sum_{m>k} q^{-dm}(|f_l((b_0\dots b_m)_q)|+|f_l((a_0\dots a_m)_q)|)\\ &\quad + |q^{-d\{b\}}-q^{-d\{a\}}|\sum_{l\in\mathcal{P}}\sum_{m\ge 0} q^{-dm}|f_l((a_0\dots a_m)_q)|\end{aligned}$$

because the summands for $m \le k$ cancel in the first sum as the first $k+1$ digits coincide. By using the estimates

$$\begin{aligned}|\{b\}-\{a\}| &\le |\{b\}-\{a\}|^\alpha = |b-a|^\alpha,\\ |q^{-d\{b\}}-q^{-d\{a\}}| &= \mathcal{O}(|b-a|^\alpha),\\ |f_l((b_0\dots b_m)_q)| &= \mathcal{O}(q^{(d-1)m}m)\end{aligned}$$

(see Lemma 2.3.9 for the last estimate), we obtain

$$\begin{aligned}|\Psi_1(b)-\Psi_1(a)| &= \mathcal{O}\Big(|b-a|^\alpha+\sum_{m>k} mq^{-m}+|b-a|^\alpha\Big)\\ &= \mathcal{O}(|b-a|^\alpha+kq^{-k}) = \mathcal{O}(|b-a|^\alpha+q^{-\alpha k})\\ &= \mathcal{O}(|b-a|^\alpha+|y-x|^\alpha) = \mathcal{O}(|y-x|^\alpha).\end{aligned}$$

Here, (2.37) has been used in the penultimate step. □

We now use the continuity of Ψ_1 and Statement 2.3.13 to remove the condition on coinciding digits from Statement 2.3.13.

Statement 2.3.14. *Let a, $b \in \mathbb{R}$ with $x \le a < b \le y$ and $\lfloor a\rfloor = \lfloor b\rfloor$. Then*

$$|\Psi_1(b)-\Psi_1(a)| = \mathcal{O}(|y-x|^\alpha).$$

PROOF. We write the expansions of $q^{\{a\}}$ and $q^{\{b\}}$ as

$$q^{\{a\}} = (a_0.a_1\dots)_q, \quad q^{\{b\}} = (b_0.b_1\dots)_q.$$

This yields

$$0 < q^{\{b\}}-q^{\{a\}} = \frac{1}{q^{\lfloor a\rfloor}}(q^b-q^a) \le q^b-q^a \le q^y-q^x < q^{-k}.$$

Thus

$$0 \le (b_0\dots b_k)_q-(a_0\dots a_k)_q \le 1.$$

If $(b_0\dots b_k)_q = (a_0\dots a_k)_q$, the result follows immediately from Statement 2.3.13. Otherwise, we have

$$(b_0\dots b_k)_q = (a_0\dots a_k)_q+1. \tag{2.38}$$

For $m \ge 0$, define z and z_m by $\lfloor z\rfloor = \lfloor z_m\rfloor = \lfloor a\rfloor = \lfloor b\rfloor$ and

$$q^{\{z\}} = (b_0.b_1\dots b_k)_q,$$

$$q^{\{z_m\}} = (a_0.a_1 \dots a_k(q-1)^m)_q.$$

Then $\lim_{m\to\infty} z_m = z$ because of (2.38).

By construction of z and z_m, we have $a < z_m < z \leq b$ for sufficiently large m.

By continuity of Ψ_1,

$$|\Psi_1(z) - \Psi_1(z_m)| \leq |y-x|^\alpha \tag{2.39}$$

holds for sufficiently large m.

This yields

$$|\Psi_1(b) - \Psi_1(a)| \leq |\Psi_1(b) - \Psi_1(z)| + |\Psi_1(z) - \Psi_1(z_m)| + |\Psi_1(z_m) - \Psi_1(a)|.$$

The third summand can be bounded by Statement 2.3.13 (for a and z_m) and the second by (2.39). The first summand is either 0 or can be bounded by Statement 2.3.13 (for z and b). □

To finally prove (2.36) for sufficiently small $y - x < 1$, we only have to remove the assumption $\lfloor a \rfloor = \lfloor b \rfloor$ from Statement 2.3.14. We use the idea of the proof of Statement 2.3.14 once more.

Assume that $\lfloor y \rfloor > \lfloor x \rfloor$. By our assumption $y < x + 1$, this amounts to $\lfloor y \rfloor = \lfloor x \rfloor + 1$. For $m \geq 0$, define z and z_m by $z = \lfloor y \rfloor$, $\lfloor z_m \rfloor = \lfloor x \rfloor$ and $q^{\{z_m\}} = ((q-1).(q-1)^m)_q$. Then $\lim_{m\to\infty} z_m = z$. By continuity of Ψ_1, we have

$$|\Psi_1(z) - \Psi_1(z_m)| \leq |y-x|^\alpha \tag{2.40}$$

and $x < z_m < z \leq y$ for sufficiently large m.

Then, this yields

$$|\Psi_1(y) - \Psi_1(x)| \leq |\Psi_1(y) - \Psi_1(z)| + |\Psi_1(z) - \Psi_1(z_m)| + |\Psi_1(z_m) - \Psi_1(x)|.$$

The third summand can be bounded by Statement 2.3.14 for x and z_m and the second by (2.40). The first vanishes or can be bounded by Statement 2.3.14 for z and y.

This yields

$$|\Psi_1(y) - \Psi_1(x)| = \mathcal{O}(|y-x|^\alpha).$$

Therefore, (2.36) is satisfied with a suitable positive constant C for $y - x < \varepsilon$ for some $\varepsilon > 0$.

Assume $y - x \geq \varepsilon$. As Ψ_1 is continuous and periodic, $|\Psi_1(y) - \Psi_1(x)|$ is bounded. Thus, (2.36) holds for a suitable positive constant C for $|y - x| \geq \varepsilon$.

Therefore, the function Ψ_1 is Hölder continuous of order $\alpha < 1$. □

2.3.5. Limiting Distribution. Finally, we can prove the parts of Theorem 2.1 concerning the approximation of the distribution function and the central limit theorem.

Proof. To prove that the distribution function can be approximated by a Gaussian mixture, we use the Berry-Esseen inequality (cf., for instance, [30, Theorem IX.5]) to estimate the difference between distribution functions. The proof follows the proof of Hwang's Quasi-Power Theorem [66]. First, we describe the two corresponding characteristic functions.

Let $\hat{g}_N(t)$ be the characteristic function of a mixture of Gaussian or degenerate distributions with weights λ_j, means $a_j\sqrt{\log_q N}$ and variances b_j for $j = 1, \dots, c$, that is

$$\hat{g}_N(t) = \sum_{j=1}^{c} \lambda_j \exp\left(i a_j \sqrt{\log_q N} t - \frac{b_j}{2} t^2\right)$$

with a_j, b_j and λ_j defined in (2.4).

By Lemma 2.3.7, the characteristic function $\hat{f}_N(t)$ of $\mathcal{T}(\boldsymbol{n})/\sqrt{\log_q N}$ is

$$\hat{f}_N(t) = \sum_{j=1}^{c} \exp\Big(ia_j\sqrt{\log_q N}t - \frac{b_j}{2}t^2 + \mathcal{O}\Big(\frac{t^3}{\sqrt{\log N}}\Big)\Big) \\ \times \sum_{l\in\mathcal{P}} e^{\frac{2\pi i l}{p}\log_q N}\Psi_{lj}\Big(\log_q N, \frac{t}{\sqrt{\log_q N}}\Big) + R\Big(N, \frac{t}{\sqrt{\log N}}\Big)$$

for $t\log_q^{-\frac{1}{2}} N$ in a fixed neighborhood of 0.

Because of (2.26) and $R(N,0) = 0$ (see Lemma 2.3.7), we have

$$\hat{f}_N(t) = \sum_{j=1}^{c} \exp\Big(ia_j\sqrt{\log_q N}t - \frac{b_j}{2}t^2\Big)\exp\Big(\mathcal{O}\Big(\frac{t^3}{\sqrt{\log N}}\Big)\Big) \\ \times \Big(\lambda_j + \mathcal{O}\Big(\frac{t}{\sqrt{\log N}}\Big)\Big) + \mathcal{O}\Big(N^{-\xi}t\sqrt{\log N}\Big).$$

Now we use the inequality $|e^w - 1| \le |w|e^{|w|}$, valid for all complex numbers w, to obtain

(2.41)
$$\Big|\frac{1}{t}(\hat{f}_N(t) - \hat{g}_N(t))\Big| = \sum_{j=1}^{c} \mathcal{O}\Big(\Big(\frac{t^2+1}{\sqrt{\log N}}\Big)\exp\Big(-\frac{b_j}{2}t^2 + \mathcal{O}\Big(\frac{t^3}{\sqrt{\log N}}\Big)\Big)\Big) \quad + \mathcal{O}(N^{-\xi}\log^{-\frac{1}{2}} N)$$

for $t\log_q^{-\frac{1}{2}} N$ in a small neighborhood of 0.

From now on, we assume that $b_j \neq 0$. There is a small neighborhood of 0 for $t\log_q^{-\frac{1}{2}} N$ such that

$$\mathcal{O}\Big(\exp\Big(-\frac{b_j}{2}t^2 + \mathcal{O}\Big(\frac{t^3}{\sqrt{\log N}}\Big)\Big)\Big) = \mathcal{O}\Big(\exp\Big(-\frac{b_j}{4}t^2\Big)\Big)$$

holds.

This yields

$$\Big|\frac{1}{t}(\hat{f}_N(t) - \hat{g}_N(t))\Big| = \sum_{j=1}^{c} \mathcal{O}\Big(\exp\Big(-\frac{b_j}{4}t^2\Big)\frac{t^2+1}{\sqrt{\log N}}\Big) + \mathcal{O}(N^{-\xi}\log^{\frac{1}{2}} N).$$

Now, the Berry-Esseen inequality with $T = c\sqrt{\log_q N}$ for a small constant $c > 0$ (cf., for instance, [30, Theorem IX.5]) implies that

$$\sup_{x\in\mathbb{R}} |F_N(x) - G_N(x)| = \mathcal{O}\Big(\frac{1}{\sqrt{\log N}}\Big)$$

where F_N is the cumulative distribution function of $\mathcal{T}(\boldsymbol{n})$ and G_N is the cumulative distribution function of the mixture of Gaussian distributions.

If all a_j are equal and $b_j \ge 0$, G_N is the distribution function of a mixture of normal (or degenerate) distributions with mean $e_{\mathcal{T}}\sqrt{\log_q N}$ and variances $b_j \ge 0$. After subtracting the mean, (2.41) converges to 0. Thus,

$$\frac{\mathcal{T}(\boldsymbol{n}) - \mathbb{E}(\mathcal{T}(\boldsymbol{n}))}{\sqrt{\log_q N}}$$

converges in distribution. If all $b_j > 0$, then the same estimates as above yield the speed of convergence. □

This completes the proof of Theorem 2.1.

2.4. Fourier Coefficients—Proof of Theorem 2.2

This section contains the proof of the theorem about the Fourier coefficients. First, we investigate some Dirichlet series which we will use later. Then, we prove the formulas given in Theorem 2.2. We use the Hölder condition for Ψ_1 to prove that its Fourier series converges.

Lemma 2.4.1. *The Dirichlet series*

$$L(z) = \sum_{r\geq 1} \lfloor \log_q r \rfloor (r^d - (r-1)^d) r^{-z}$$

is meromorphic in $\Re z > d-1$ with poles in $z = d + \frac{2\pi i l}{\log q}$ for $l \in \mathbb{Z}$. The main part at $z = d$ is

$$\frac{d}{(z-d)^2 \log q} - \frac{d}{2(z-d)}$$

and, for $l \neq 0$, the residue at $z = d + \frac{2\pi i l}{\log q}$ is $\frac{d}{2\pi i l}$.

Proof. First, we use the binomial theorem to obtain

$$L(z) = dL_1(z-d+1) - \sum_{j=0}^{d-2} \binom{d}{j} (-1)^{d-j} L_1(z-j) \tag{2.42}$$

with $L_1 = \sum_{r\geq 1} \lfloor \log_q r \rfloor r^{-z}$. The Dirichlet series $L_1(z)$ is holomorphic for $\Re z > 1$. Thus, the second summand in (2.42) is holomorphic for $\Re z > d-1$. To obtain the expansion of $L(z)$ at z with $\Re z = d$, we investigate the Dirichlet series $L_1(z)$ at $\Re z = 1$.

Let $k \geq 0$ be an integer. We use Euler-Maclaurin summation with $f(x) = kx^{-z}$ to obtain

$$\begin{aligned}
\sum_{q^k \leq r < q^{k+1}} \frac{\lfloor \log_q r \rfloor}{r^z} &= \int_{q^k}^{q^{k+1}} kx^{-z}\, dx - \frac{k}{2}(q^{-(k+1)z} - q^{-kz}) \\
&\quad - kz \int_{q^k}^{q^{k+1}} B_1(\{x\}) x^{-z-1}\, dx \\
&= \frac{1}{1-z}(kq^{(k+1)(1-z)} - kq^{k(1-z)}) \\
&\quad - \frac{1}{2}(kq^{-(k+1)z} - kq^{-kz}) \\
&\quad - z \int_{q^k}^{q^{k+1}} B_1(\{x\}) x^{-z-1} \lfloor \log_q(x) \rfloor\, dx
\end{aligned}$$

where $B_1(x)$ is the first Bernoulli polynomial. For $\Re z > 1$, summation over $k \geq 0$ yields

$$\begin{aligned}
L_1(z) &= \frac{1}{1-z} \sum_{k\geq 1} q^{k(1-z)}((k-1)-k) - \frac{1}{2} \sum_{k\geq 1} q^{-zk}((k-1)-k) \\
&\quad - z \int_1^\infty B_1(\{x\}) x^{-z-1} \lfloor \log_q(x) \rfloor\, dx \\
&= \frac{1}{z-1} \frac{1}{q^{z-1}-1} + \frac{1}{2} \frac{1}{q^z - 1} - z \int_1^\infty B_1(\{x\}) x^{-z-1} \lfloor \log_q(x) \rfloor\, dx.
\end{aligned}$$

The second summand and the integral are clearly holomorphic for $\Re z > 0$. Thus, $L_1(z)$ can be continued meromorphically to $\Re z > 0$ with poles coming from the first summand.

The expansion around $z = 1$ is

$$\frac{1}{z-1}\frac{1}{q^{z-1}-1} + O(1) = \frac{1}{(z-1)^2 \log q} - \frac{1}{2(z-1)} + O(1).$$

Thus, by (2.42), we obtain the main part and the residues of $L(z)$ at $z = d + \frac{2\pi i l}{\log q}$ for $l \in \mathbb{Z}$ as stated in the lemma. □

Lemma 2.4.2. *The Dirichlet series*

$$Z(z) = \sum_{r \geq 1} (r^d - (r-1)^d) r^{-z}$$

is meromorphic in $\mathbb{C}$ *with simple poles in* $z = j$, $j \in \{1, \dots, d\}$ *with residues* $\binom{d}{j-1}(-1)^{d-j}$.

Proof. The binomial theorem yields

$$Z(z) = \sum_{j=0}^{d-1} \binom{d}{j} (-1)^{d-j+1} \zeta(z-j),$$

where ζ is the Riemann ζ-function. The result follows from the unique pole of $\zeta(z)$ at $z = 1$ with residue 1. □

Denote by $\zeta(z, \alpha)$ the Hurwitz ζ-function

$$\zeta(z, \alpha) = \sum_{r > -\alpha} (r + \alpha)^{-z}.$$

Furthermore ψ is the digamma function.

Lemma 2.4.3. *For* $0 \leq \alpha < 1$ *and and an integer* $0 \leq j \leq d-1$, *the Dirichlet series*

$$J(z, \alpha, j) = \sum_{r \geq 1} r^j (r + \alpha)^{-z}$$

is analytic for $\Re z > j + 1$. *For* $j = d - 1$, *it is meromorphic for* $\Re z > d - 1$ *with a simple pole at* $z = d$ *with expansion*

$$\begin{aligned} J(z, \alpha, d-1) = {} & \frac{1}{z-d} - \psi(\alpha + [\alpha = 0]) - [\alpha > 0 \wedge d = 1]\alpha^{-1} \\ & + \sum_{k=0}^{d-2} \binom{d-1}{k} (-\alpha)^{d-1-k} \zeta(d-k, \alpha) + \mathcal{O}(z-d). \end{aligned} \tag{2.43}$$

Proof. As $r^j (r+\alpha)^{-z} = \mathcal{O}(r^{j - \Re z})$, J is analytic for $\Re z > j + 1$. Now, let $j = d - 1$. The binomial theorem yields

$$\begin{aligned} J(z, \alpha, d-1) &= \sum_{r \geq 1} (r + \alpha - \alpha)^{d-1} (r+\alpha)^{-z} \\ &= \sum_{k=0}^{d-1} \binom{d-1}{k} (-\alpha)^{d-1-k} \sum_{r \geq 1} (r+\alpha)^{-(z-k)} \\ &= \sum_{k=0}^{d-1} \binom{d-1}{k} (-\alpha)^{d-1-k} (\zeta(z-k, \alpha) - [\alpha > 0]\alpha^{-z+k}) \\ &= \zeta(z-d+1, \alpha) + \sum_{k=0}^{d-2} \binom{d-1}{k} (-\alpha)^{d-1-k} \zeta(z-k, \alpha) \end{aligned}$$

$$- [\alpha > 0 \wedge d = 1]\alpha^{-z}.$$

Using the expansion (cf. [103, p. 271])

$$\zeta(z, \alpha) = \frac{1}{z-1} - \psi(\alpha + [\alpha = 0]) + \mathcal{O}(z-1)$$

yields (2.43). □

Lemma 2.4.4. *Let $k \in \mathbb{Z}$. The Dirichlet series*

$$B(z) = \boldsymbol{w}_k^\top \sum_{r=1}^{\infty} \left(\boldsymbol{B}_\emptyset(r+1) - 2\boldsymbol{B}_\emptyset(r) + \boldsymbol{B}_\emptyset(r-1)\right) r^{-z}$$

is analytic for $\Re z > d - 1$.

Proof. By the definition (2.29), we have

$$\boldsymbol{B}_\emptyset(r+1) - \boldsymbol{B}_\emptyset(r) = \sum_{\emptyset \neq C \subseteq \{1,\dots,d\}} \boldsymbol{B}_C(r), \tag{2.44}$$

which can be bounded by $\|\boldsymbol{B}_C(r)\| = \mathcal{O}(r^{d-1} \log r)$. Thus,

$$B(z) = \boldsymbol{w}_k^\top \sum_{\emptyset \neq C \subseteq \{1,\dots,d\}} \sum_{r \geq 1} (\boldsymbol{B}_C(r) - \boldsymbol{B}_C(r-1)) r^{-z}$$

which converges for $\Re z > d - 1$ by [2, Theorem 8.1]. □

The vector-valued functions $\boldsymbol{H}_C(z)$ are defined by the Dirichlet series

$$\boldsymbol{H}_C(z) = \sum_{r \geq 1} \boldsymbol{B}_C(r) r^{-z}. \tag{2.45}$$

By (2.5) and (2.44), this yields

$$\boldsymbol{H}(z) = \sum_{\emptyset \neq C \subseteq \{1,\dots,d\}} \boldsymbol{H}_C(z) = \sum_{r \geq 1} (\boldsymbol{B}_\emptyset(r+1) - \boldsymbol{B}_\emptyset(r)) r^{-z}. \tag{2.46}$$

Next, we investigate the Dirichlet series $\boldsymbol{H}_C$. In particular, we determine its behavior at $z = d + \chi_k$ and provide an infinite functional equation to compute its residues at these points. This will finally give us the residues of $\boldsymbol{H}$ in (2.6). We use a similar method as Grabner and Hwang in [39].

For this infinite recursion, define

$$\boldsymbol{\delta}_{C,D}^{\varepsilon} = \sum_{\substack{\beta_i = 0 \\ i \notin C \cup D}}^{q-1} \sum_{\substack{\beta_i = 0 \\ i \in D}}^{\varepsilon - 1} \sum_{\substack{\beta_i = \varepsilon \\ i \in C}} \boldsymbol{\delta}_\beta, \tag{2.47}$$

in analogy to the definition of $M_{C,D}^{\varepsilon}$. As before, the s-th entry of $\boldsymbol{\delta}_{\boldsymbol{\varepsilon}}$ is the output label of the transition starting in s with input label $\boldsymbol{\varepsilon}$. Then, $\boldsymbol{\delta} = \boldsymbol{\delta}_{\emptyset,\emptyset}^{\varepsilon}$ holds independently of ε. Furthermore, $\boldsymbol{\delta}_{C,D}^{\varepsilon} = \frac{d}{dt} M_{C,D}^{\varepsilon} \mathbf{1}\Big|_{t=0}$ by (2.34).

Lemma 2.4.5. *Let $C \neq \emptyset$. For $\Re z > d$ and $C \neq \emptyset$, the Dirichlet series $\boldsymbol{H}_C(z)$ satisfies the following infinite recursion*

$$\begin{aligned}\Big(1 - q^{-z}\sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,\emptyset}\Big)\boldsymbol{H}_C(z) &= \sum_{\varepsilon=1}^{q-1}\boldsymbol{B}_C(\varepsilon)\varepsilon^{-z} + q^{-z}\sum_{\emptyset\neq D\subseteq C^c}\sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,D}\boldsymbol{H}_{C\cup D}(z)\\ &\quad + q^{-z}\sum_{D\subseteq C^c}\sum_{\varepsilon=0}^{q-1}\boldsymbol{\delta}^{\varepsilon}_{C,D}J\Big(z,\frac{\varepsilon}{q},d-|D|-|C|\Big)\\ &\quad + \sum_{D\subseteq C^c}\sum_{m\geq 1}\binom{-z}{m}q^{-z-m}\sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,D}\varepsilon^m\boldsymbol{H}_{C\cup D}(z+m).\end{aligned} \tag{2.48}$$

It is analytic for $\Re z > d - |C| + 1$. For $|C| = 1$ and $k \neq 0$, $\boldsymbol{w}_k^{\top}\boldsymbol{H}_C$ has a possible simple pole in $z = d + \chi_k$ with residue the right-hand side of (2.48) *evaluated at $z = d + \chi_k$ and divided by $\log q$. For $|C| = 1$, $\boldsymbol{w}_0^{\top}\boldsymbol{H}_C$ has a possible double pole with main part*

$$\frac{e_{\mathcal{T}}}{\log q}\frac{1}{(z-d)^2} + \Big(\frac{e_{\mathcal{T}}}{2} + \frac{h_C}{\log q}\Big)\frac{1}{z-d}$$

where h_C is given in (2.51).

Remark 2.4.6. The infinite recursion (2.48) can be used to numerically compute the values of $\boldsymbol{H}_C$ and its residues at $z = d + \chi_k$ with arbitrary precision. It numerically converges fast if the first terms of the Dirichlet series $\boldsymbol{H}_C$ are computed explicitly.

Proof. As $\boldsymbol{B}_C(r) = \mathcal{O}(r^{d-|C|}\log r)$, the Dirichlet series $\boldsymbol{H}_C$ is analytic for $\Re z > d - |C| + 1$.

By multiplying (2.19) with $\boldsymbol{u}(t)$, differentiating with respect to t at $t = 0$ and using (2.33), (2.18) and (2.47), we obtain the recursion

$$\boldsymbol{B}_C(qr+\varepsilon) = \sum_{D\subseteq C^c} M^{\varepsilon}_{C,D}\boldsymbol{B}_{C\cup D}(r) + \boldsymbol{\delta}^{\varepsilon}_{C,D}r^{d-|D|-|C|} \tag{2.49}$$

for $C \neq \emptyset$, $\{1,\dots,d\}$ and $qr + \varepsilon \geq 0$. By (2.16), this recursion is also valid for $C = \{1,\dots,d\}$ and $qr + \varepsilon > 0$.

By (2.49), we have

$$\begin{aligned}\boldsymbol{H}_C(z) &= \sum_{\varepsilon=1}^{q-1}\boldsymbol{B}_C(\varepsilon)\varepsilon^{-z} + \sum_{\varepsilon=0}^{q-1}\sum_{r\geq 1}\boldsymbol{B}_C(qr+\varepsilon)(qr+\varepsilon)^{-z}\\ &= \sum_{\varepsilon=1}^{q-1}\boldsymbol{B}_C(\varepsilon)\varepsilon^{-z}\\ &\quad + \sum_{D\subseteq C^c}\sum_{\varepsilon=0}^{q-1}\sum_{r\geq 1}(M^{\varepsilon}_{C,D}\boldsymbol{B}_{C\cup D}(r) + \boldsymbol{\delta}^{\varepsilon}_{C,D}r^{d-|D|-|C|})q^{-z}r^{-z}\Big(1+\frac{\varepsilon}{qr}\Big)^{-z}\end{aligned}$$

for $C \neq \emptyset$. Expanding $(1 + \varepsilon/(qr))^{-z}$ as a binomial series yields

$$\boldsymbol{H}_C(z) = \sum_{\varepsilon=1}^{q-1}\boldsymbol{B}_C(\varepsilon)\varepsilon^{-z}$$

$$
\begin{aligned}
&+ \sum_{D\subseteq C^c} \sum_{\varepsilon=0}^{q-1} \sum_{r\geq 1} \sum_{m\geq 0} \binom{-z}{m} M^{\varepsilon}_{C,D} \varepsilon^m q^{-z-m} \boldsymbol{B}_{C\cup D}(r) r^{-z-m} \\
&+ q^{-z} \sum_{D\subseteq C^c} \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}^{\varepsilon}_{C,D} J\Big(z, \frac{\varepsilon}{q}, d-|D|-|C|\Big) \\
&= \sum_{\varepsilon=1}^{q-1} \boldsymbol{B}_C(\varepsilon)\varepsilon^{-z} + q^{-z} \sum_{D\subseteq C^c} \sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,D} \boldsymbol{H}_{C\cup D}(z) \\
&+ q^{-z} \sum_{D\subseteq C^c} \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}^{\varepsilon}_{C,D} J\Big(z, \frac{\varepsilon}{q}, d-|D|-|C|\Big) \\
&+ \sum_{D\subseteq C^c} \sum_{m\geq 1} \binom{-z}{m} q^{-z-m} \sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,D} \varepsilon^m \boldsymbol{H}_{C\cup D}(z+m)
\end{aligned}
$$

for $\Re z > d$ and $C \neq \emptyset$. Collecting $\boldsymbol{H}_C(z)$ on the left-hand side results in (2.48).

To compute the residues of $\boldsymbol{w}_k^\top \boldsymbol{H}_C$ for $|C| = 1$ at $z = d + \chi_k$, note that $\sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,\emptyset} = M$ holds independently of C.

We multiply (2.48) with the left eigenvector $\boldsymbol{w}_k^\top$ which results in

$$
\begin{aligned}
&\Big(1 - q^{d-z} \exp\Big(\frac{2\pi i k}{p}\Big)\Big) \boldsymbol{w}_k^\top \boldsymbol{H}_C(z) = \\
&\qquad \boldsymbol{w}_k^\top \sum_{\varepsilon=1}^{q-1} \boldsymbol{B}_C(\varepsilon)\varepsilon^{-z} \\
&\qquad + q^{-z} \boldsymbol{w}_k^\top \sum_{\emptyset\neq D\subseteq C^c} \sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,D} \boldsymbol{H}_{C\cup D}(z) \\
&\qquad + q^{-z} \boldsymbol{w}_k^\top \sum_{D\subseteq C^c} \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}^{\varepsilon}_{C,D} J\Big(z, \frac{\varepsilon}{q}, d-|D|-1\Big) \\
&\qquad + \boldsymbol{w}_k^\top \sum_{D\subseteq C^c} \sum_{m\geq 1} \binom{-z}{m} q^{-z-m} \sum_{\varepsilon=0}^{q-1} M^{\varepsilon}_{C,D} \varepsilon^m \boldsymbol{H}_{C\cup D}(z+m).
\end{aligned} \tag{2.50}
$$

As $|C\cup D| \geq 2$ or $\Re z + m > d$, all $\boldsymbol{H}_{C\cup D}$ used on right-hand side of (2.50) are well defined for $\Re z > d-1$. The Dirichlet series J have simple poles at $z = d$ for $|C| = 1$ and $D = \emptyset$ (Lemma 2.4.3). Thus the right-hand side of (2.50) is meromorphic for $\Re z > d-1$ with a simple pole at $z = d$.

The factor $1 - q^{d-z}\exp(\frac{2\pi i k}{p})$ has a zero exactly for $z = d + \chi_k$, $k \in \mathbb{Z}$. Thus for $k \neq 0$, $\boldsymbol{w}_k^\top \boldsymbol{H}_C$ has a possible simple pole at $z = d + \chi_k$. Its residue is the right-hand side of (2.50) evaluated at $z = d + \chi_k$ divided by $\log q$.

If $k = 0$, we have $z = d$. In this case the expansion of the right-hand side of (2.50) is

$$
\frac{e_{\mathcal{T}}}{z-d} + h_C + \mathcal{O}(z-d)
$$

with

$$
\begin{aligned}
h_C = {} & -e_{\mathcal{T}} \log q - q^{-d} \boldsymbol{w}_0^\top \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}_{C,\emptyset}^{\varepsilon} \psi\Big(\frac{\varepsilon}{q} + [\varepsilon = 0]\Big) \\
& - [d=1] \boldsymbol{w}_0^\top \sum_{\varepsilon=1}^{q-1} \boldsymbol{\delta}_{C,\emptyset}^{\varepsilon} \varepsilon^{-1} \\
& + q^{-d} \boldsymbol{w}_0^\top \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}_{C,\emptyset}^{\varepsilon} \sum_{k=0}^{d-2} \binom{d-1}{k} \Big(-\frac{\varepsilon}{q}\Big)^{d-1-k} \zeta\Big(d-k, \frac{\varepsilon}{q}\Big) \\
& + \boldsymbol{w}_0^\top \sum_{\varepsilon=1}^{q-1} \boldsymbol{B}_C(\varepsilon) \varepsilon^{-d} + q^{-d} \boldsymbol{w}_0^\top \sum_{\emptyset \neq D \subseteq C^c} \sum_{\varepsilon=0}^{q-1} M_{C,D}^{\varepsilon} H_{C \cup D}(d) \\
& + q^{-d} \boldsymbol{w}_0^\top \sum_{\emptyset \neq D \subseteq C^c} \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}_{C,D}^{\varepsilon} J\Big(d, \frac{\varepsilon}{q}, d - |D| - 1\Big) \\
& + \boldsymbol{w}_0^\top \sum_{D \subseteq C^c} \sum_{m \geq 1} \binom{-d}{m} q^{-d-m} \sum_{\varepsilon=0}^{q-1} M_{C,D}^{\varepsilon} \varepsilon^m \boldsymbol{H}_{C \cup D}(d+m)
\end{aligned} \tag{2.51}
$$

where we used the expansion of J in Lemma 2.4.3, $\boldsymbol{\delta} = \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}_{C,\emptyset}^{\varepsilon}$ and (2.13). □

From the previous lemma and (2.46), the residues of the Dirichlet function $\boldsymbol{H}$ follow. Only $\boldsymbol{H}_C$ with $|C| = 1$ contribute as all other summands are holomorphic.

Lemma 2.4.7. *The Dirichlet function $\boldsymbol{H}$ is meromorphic in $\Re z > d-1$ with possible simple poles at $z = d + \chi_k$, $k \neq 0$ and a possible double pole at $z = d$.*

The residue at $z = d + \chi_k$, $k \neq 0$ is

$$
\begin{aligned}
& \frac{1}{\log q} \sum_{j=1}^{d} \Bigg(\sum_{\varepsilon=1}^{q-1} \boldsymbol{B}_{\{j\}}(\varepsilon) \varepsilon^{-d-\chi_k} \\
& + q^{-d-\chi_k} \sum_{\emptyset \neq D \subseteq \{j\}^c} \sum_{\varepsilon=0}^{q-1} M_{\{j\},D}^{\varepsilon} \boldsymbol{H}_{\{j\} \cup D}(d+\chi_k) \\
& + q^{-d-\chi_k} \sum_{D \subseteq \{j\}^c} \sum_{\varepsilon=0}^{q-1} \boldsymbol{\delta}_{\{j\},D}^{\varepsilon} J\Big(d+\chi_k, \frac{\varepsilon}{q}, d - |D| - 1\Big) \\
& + \sum_{D \subseteq \{j\}^c} \sum_{m \geq 1} \binom{-d-\chi_k}{m} q^{-d-m-\chi_k} \sum_{\varepsilon=0}^{q-1} M_{\{j\},D}^{\varepsilon} \varepsilon^m \boldsymbol{H}_{\{j\} \cup D}(d+m+\chi_k) \Bigg).
\end{aligned}
$$

The main part at $z = d$ is

$$
\frac{e_{\mathcal{T}} d}{\log q} \frac{1}{(z-d)^2} + \Big(\frac{e_{\mathcal{T}} d}{2} + \sum_{j=1}^{d} \frac{h_{\{j\}}}{\log q} \Big) \frac{1}{z-d}
$$

where $h_{\{j\}}$ is defined in (2.51).

Now we can prove the formulas for the Fourier coefficients.

PROOF OF THEOREM 2.2. The periodic fluctuation Ψ_1 of the expected value is a p-periodic function. We use the explicit expression of Ψ_1 given in Lemma 2.3.9.

Due to absolute convergence, the k-th Fourier coefficient of $\Psi_1(x)$ is

$$\begin{aligned} c_k &= \frac{1}{p}\int_0^p \Psi_1(x) e^{-\frac{2\pi i k}{p}x}\,dx \\ &= -\frac{e_{\mathcal{T}}}{p}\int_0^p \{x\} e^{-\frac{2\pi i k}{p}x}\,dx - \sum_{l\in\mathcal{P}}\sum_{m=0}^{\infty} q^{-dm} e^{-\frac{2\pi i l m}{p}} I_{l,m} \end{aligned}$$

with

$$I_{l,m} = \frac{1}{p}\int_0^p q^{-d\{x\}} \exp\Big(\frac{2\pi i l}{p}\lfloor x\rfloor - \frac{2\pi i k}{p}x\Big) f_l((x_0 \dots x_m)_q)\,dx$$

and $q^{\{x\}} = (x_0.x_1\dots)_q$. The value of the first integral is given by $-\frac{e_{\mathcal{T}}}{2}$ for $k=0$, and $[k \equiv 0 \bmod p]\frac{e_{\mathcal{T}}}{\chi_k \log q}$ otherwise. Thus, we focus on the second integral $I_{l,m}$.

First, we partition the interval $[0,p)$ into intervals $[r, r+1)$ for $r = 0, \dots, p-1$. After simplifying the sum of p-th roots of unity, we obtain

$$I_{l,m} = [k \equiv l \bmod p]\int_0^1 q^{-dx} f_l((x_0 \dots x_m)_q) e^{-\frac{2\pi i k}{p}x}\,dx.$$

After partitioning the interval $[0,1)$ into the intervals $[\log_q r - m, \log_q(r+1) - m)$ for $r = q^m, \dots, q^{m+1}-1$, the function $f_l((x_0 \dots x_m)_q)$ is constant on the interval of integration. Therefore, we obtain

$$\sum_{l\in\mathcal{P}}\sum_{m=0}^{\infty} q^{-md} e^{-\frac{2\pi i l m}{p}} I_{l,m} = \frac{1}{(d+\chi_k)\log q}\sum_{r=1}^{\infty} f_{k \bmod p}(r)\left(r^{-d-\chi_k} - (r+1)^{-d-\chi_k}\right).$$

Next, consider the function

$$A(z) = \sum_{r=1}^{\infty} f_{k \bmod p}(r)\left(r^{-z} - (r+1)^{-z}\right).$$

We know that $f_l(r) = \mathcal{O}(r^{d-1}\log r)$. Thus, $A(z)$ is analytic for $\Re z > d-1$.

By summation by parts, we can rearrange the series for $\Re z > d$ and obtain a sum of Dirichlet series

$$A(z) = [p \mid k] e_{\mathcal{T}} S_1(z) + i\boldsymbol{w}_k'^{\top}\mathbf{1} S_2(z) - S_3(z) + q^d \exp\Big(\frac{2\pi i k}{p}\Big) S_4(z) \tag{2.52}$$

with coefficients $s_1(r)$, $s_2(r)$, $s_3(r)$ and $s_4(r)$ respectively. These coefficients are differences of the four summands in $f_{k \bmod p}(r)$ and $f_{k \bmod p}(r-1)$ in (2.31), respectively, e.g.,

$$\begin{aligned} s_1(r) = {} & \lfloor \log_q(r)\rfloor (r^d - (q\lfloor r/q\rfloor)^d) + (q\lfloor r/q\rfloor)^d \\ & - [r>1]\big(\lfloor \log_q(r-1)\rfloor((r-1)^d - (q\lfloor (r-1)/q\rfloor)^d) - (q\lfloor (r-1)/q\rfloor)^d\big). \end{aligned}$$

After some simplifications using $\lfloor \frac{r-1}{q} \rfloor = \lfloor \frac{r}{q} \rfloor - [q \mid r]$ and $\lfloor \log_q(r-1) \rfloor = \lfloor \log_q r \rfloor - [r \text{ is a power of } q]$ (for $r \geq 2$), we obtain

$$\begin{aligned}
s_1(r) &= \lfloor \log_q r \rfloor (r^d - (r-1)^d) \\
&\quad - [q \mid r] q^d \lfloor \log_q r q^{-1} \rfloor ((rq^{-1})^d - (rq^{-1}-1)^d) \\
&\quad + [r \neq 1 \text{ is a power of } q]((r-1)^d - (r-q)^d), \\
s_2(r) &= r^d - (r-1)^d - [q \mid r] q^d \exp\Big(\frac{2\pi i k}{p}\Big)((rq^{-1})^d - (rq^{-1}-1)^d), \\
s_3(r) &= \boldsymbol{w}_k^\top (\boldsymbol{B}_\emptyset(r) - \boldsymbol{B}_\emptyset(r-1)), \\
s_4(r) &= [q \mid r] \boldsymbol{w}_k^\top (\boldsymbol{B}_\emptyset(rq^{-1}) - \boldsymbol{B}_\emptyset(rq^{-1}-1)).
\end{aligned} \tag{2.53}$$

For $\Re z > d$, we can split up the summation into the different cases in (2.53). This yields

$$\begin{aligned}
S_1(z) &= (1-q^{d-z})L(z) + \sum_{j=0}^{d-1} \binom{d}{j} (-1)^{d-j} \frac{1-q^{d-j}}{q^{z-j}-1}, \\
S_2(z) &= \Big(1 - q^{d-z} \exp\Big(\frac{2\pi i k}{p}\Big)\Big) Z(z), \\
S_3(z) &= \boldsymbol{w}_k^\top \boldsymbol{H}(z) - B(z), \\
S_4(z) &= q^{-z} \boldsymbol{w}_k^\top \boldsymbol{H}(z) - q^{-z} B(z)
\end{aligned}$$

where we used (2.44), (2.46) and the Dirichlet series defined in Lemmas 2.4.1, 2.4.2 and 2.4.4.

Thus, in (2.52), we obtain

$$\begin{aligned}
A(z) &= [p \mid k] e_{\mathcal{T}} \sum_{j=0}^{d-1} \binom{d}{j} (-1)^{d-j} \frac{1-q^{d-j}}{q^{z-j}-1} \\
&\quad + i \boldsymbol{w}_k'^\top \mathbf{1} (1 - q^{d-z} e^{\frac{2\pi i k}{p}}) Z(z) \\
&\quad - (1 - q^{d-z} e^{\frac{2\pi i k}{p}}) \boldsymbol{w}_k^\top \boldsymbol{H}(z) \\
&\quad + [p \mid k] e_{\mathcal{T}} (1 - q^{d-z}) L(z) \\
&\quad + (1 - q^{d-z} e^{\frac{2\pi i k}{p}}) B(z).
\end{aligned} \tag{2.54}$$

We want to evaluate A at $z = d + \chi_k$. The factors $1 - q^{d-z} e^{\frac{2\pi i k}{p}}$ are zero if and only if $z = d + \chi_k$. Thus, the following Dirichlet series contribute to (2.54):

- The Dirichlet series Z only contributes if $k = 0$ (Lemma 2.4.2).
- The Dirichlet series $\boldsymbol{w}_k^\top \boldsymbol{H}$ has poles at $z = d + \chi_k$ for $k \in \mathbb{Z}$. The possible double pole at $z = d$ cancels with the one of L (Lemma 2.4.7).
- The residues of the Dirichlet series L contribute to the Fourier coefficients. The possible double pole at $z = d$ cancels with that of $\boldsymbol{w}_0^\top \boldsymbol{H}$ (Lemma 2.4.1).
- As the Dirichlet series B converges for $\Re z > d - 1$ (Lemma 2.4.4), it does not contribute to the Fourier coefficients.

As the second order poles of $\boldsymbol{w}_0^\top \boldsymbol{H}$ and L cancel, the right-hand side of (2.54) is well defined for the limit $z \to d + \chi_k$. After computing the limit and simplifying the summation, we obtain (2.6).

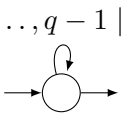


FIGURE 2.7. Transducer to compute the q-ary sum-of-digits function.

Then Lemma 2.3.12 and Bernstein's theorem (cf. [105, p. 240]) imply the absolute and uniform convergence of the Fourier series. □

Now we use Theorem 2.2 to prove Corollary 2.2.5.

PROOF OF COROLLARY 2.2.5, [102]. The transducer in Figure 2.7 computes the q-ary sum-of-digits function $s_q(n)$ and we can use Theorem 2.2.

We transform the Dirichlet series

$$D(z) = \sum_{m\geq 1} (s_q(m) - s_q(m-1))m^{-z}$$

in two different ways. This series is absolutely convergent for $\Re z > 1$.

First, we can rearrange the summation of the Dirichlet series $D(z)$ such that the Dirichlet series $H(z) = \sum_{m\geq 1} s_q(m)m^{-z}$, defined in (2.45), appears. We have

$$\begin{aligned} |H(z) - 1| &= \mathcal{O}\Big(2^{-\Re z} + \sum_{m\geq 3} m^{-\Re z}\log m\Big) \\ &= \mathcal{O}\Big(2^{-\Re z} + \int_2^\infty x^{-\Re z}\log x\, dx\Big) \\ &= \mathcal{O}(2^{-\Re z}) \end{aligned} \tag{2.55}$$

for $\Re z > 1$. By partial summation, we obtain

$$\begin{aligned} D(z) &= 1 - 2^{-z} + \sum_{m\geq 2} s_q(m)(m^{-z} - (m+1)^{-z}) \\ &= 1 - 2^{-z} + \sum_{m\geq 2} s_q(m)m^{-z}(1 - (1+m^{-1})^{-z}). \end{aligned}$$

Expanding the binomial series yields

$$\begin{aligned} D(z) &= 1 - 2^{-z} - \sum_{m\geq 2} s_q(m)m^{-z} \sum_{l\geq 1} \binom{-z}{l} m^{-l} \\ &= 1 - 2^{-z} - \sum_{l\geq 1} \binom{-z}{l} (H(z+l) - 1). \end{aligned} \tag{2.56}$$

By (2.56), we have

$$D(z) = 1 - 2^{-z} + zH(z+1) - z - \sum_{l\geq 2} \binom{-z}{l} (H(z+l) - 1)$$

which is equivalent to

$$H(z+1) = \frac{1}{z}D(z) + \frac{1}{z}(2^{-z} - 1) + 1 - \sum_{l\geq 2} \frac{1}{l}\binom{-z-1}{l-1}(H(z+l) - 1)$$

for $\Re z > 1$. The sum on the right-hand side is holomorphic at $\Re z = 0$ because of (2.55). By meromorphic continuation, this equation also holds for $\Re z = 0$. This yields

$$\operatorname{Res}_{z=1+\chi_k} H(z) = \operatorname{Res}_{z=\chi_k} H(z+1) = \operatorname{Res}_{z=\chi_k} \frac{1}{z} D(z). \tag{2.57}$$

On the other hand, we split up the summation in the definition of $D(z)$ into the q equivalence classes modulo q and we use the recursions[3]

$$s_q(qm + \varepsilon) = s_q(m) + \varepsilon$$

for $0 \le \varepsilon < q$. This results in

$$s_q(m) - s_q(m-1) = 1 + [q \mid m] \left(s_q\left(q^{-1}m\right) - s_q\left(q^{-1}m - 1\right) - q \right)$$

for $m \ge 1$. Thus we obtain

$$\begin{aligned} D(z) &= \sum_{m \ge 1} \left(1 + [q \mid m] \left(s_q\left(q^{-1}m\right) - s_q\left(q^{-1}m - 1\right) - q \right) \right) m^{-z} \\ &= \zeta(z) + q^{-z} D(z) - q^{1-z} \zeta(z). \end{aligned}$$

Thus, we obtain[4]

$$D(z) = \frac{1 - q^{1-z}}{1 - q^{-z}} \zeta(z). \tag{2.58}$$

This formula yields

$$\operatorname{Res}_{z=\chi_k} D(z) = -\frac{q-1}{\log q} \zeta(\chi_k). \tag{2.59}$$

For $k = 0$, we further use the expansion

$$\zeta(z) = -\frac{1}{2} - \frac{1}{2} \log(2\pi) z + \mathcal{O}(z^2)$$

(cf. [23, 25.6.1 and 25.6.11]) and (2.58) to obtain

$$D(z) = \frac{q-1}{2z \log q} + \frac{(q-1)\log(2\pi)}{2 \log q} - \frac{q+1}{4} + \mathcal{O}(z). \tag{2.60}$$

Thus, by (2.55) and (2.59), we obtain

$$\operatorname{Res}_{z=1+\chi_k} H(z) = \frac{1}{\chi_k} \operatorname{Res}_{z=\chi_k} D(z) = -\frac{q-1}{\chi_k \log q} \zeta(\chi_k)$$

for $k \ne 0$. For $k = 0$, (2.60) and (2.57) yield

$$\operatorname{Res}_{z=1} H(z) = \frac{(q-1)\log(2\pi)}{2 \log q} - \frac{q+1}{4}.$$

Now, (2.6) with $e_{\mathcal{T}} = \frac{q-1}{2}$ and $\boldsymbol{w}_0'^{\top} = 0$ yields (2.8). □

[3] Actually, these recursions are (2.35).

[4] Note that this well-known identity can also be derived [85] from $s_q(m) - s_q(m-1) = 1 - (q-1)v_q(m)$, where $v_q(m)$ is the q-adic valuation of m.

2.5. Non-Differentiability—Proof of Theorem 2.3

In this section, we give the proof of the non-differentiability of $\Psi_1(x)$. We follow the method presented by Tenenbaum [98], see also Grabner and Thuswaldner [40].

PROOF OF THEOREM 2.3. Let $r = (r_{m-1} \dots r_0)_q$ be the value of the reset sequence $(r_{m-1} \dots r_0)$ leading to state ν.

Assume that Ψ_1 is differentiable at $x \in [0,1)$. Let $q^x = (\varepsilon_0 . \varepsilon_1 \dots)_q$ be the standard q-ary digit expansion choosing the representation ending on 0^ω in the case of ambiguity. Further, let x_k be such that $q^{x_k} = (\varepsilon_0 . \varepsilon_1 \dots \varepsilon_k)_q$. Thus, we have $\lim_{k\to\infty} x_k = x$. For $f \in \{0,1\}$, the function $L_f \colon \mathbb{Z} \to \mathbb{Z}$ is defined as $L_f(k) = ck + f$ with c a positive integer such that $c > \frac{1}{\xi} - 1$. Define $N_k = q^{x_k+k+L_f(k)}$ and $h(k) = \lfloor q^{ck+\frac{c}{c+1}x_k - m - 2} \rfloor$. Let y_k and z_k be such that $N_k + q^{ck-m-1}r = q^{y_k+k+L_f(k)}$ and $N_k + q^{ck-m-1}r + h(k) = q^{z_k+k+L_f(k)}$.

From these definitions, we know that

$$\begin{aligned} \frac{h(k)}{N_k} &= \Theta(q^{-k}), \\ N_k^{1-\xi} \log N_k &= o(h(k)) \end{aligned}$$

for $k \to \infty$. Apart from x_k, also, y_k and z_k converge to x and satisfy the following bounds:

$$\begin{aligned} z_k - y_k &= \frac{1}{\log q} \frac{h(k)}{N_k} + \mathcal{O}\left(\frac{h(k)^2}{N_k^2}\right), \\ |y_k - x_k| &= \mathcal{O}(q^{-k}), \\ x - x_k &= \mathcal{O}(q^{-k}). \end{aligned}$$

Now, we compute

$$\frac{1}{h(k)} \sum_{n \in \mathcal{N}_k} \mathcal{T}(n) \tag{2.61}$$

in two different ways where $\mathcal{N}_k = \{n \in \mathbb{Z} \mid N_k + q^{ck-m-1}r \le n < N_k + q^{ck-m-1}r + h(k)\}$.

First, observe that $q^{ck-1} \mid N_k$ and $h(k) < q^{ck-m-1}$. Thus, the digit representations of the three summands in $N_k + q^{ck-m-1}r + n$ are not overlapping at non-zero digits for $n < h(k)$. Since the digit expansion of r is a reset sequence, we have

$$\mathcal{T}(N_k + q^{ck-m-1}r + n) = \boldsymbol{e}_\nu^\top \boldsymbol{b}(N_k q^{-ck+1}) + \mathcal{T}(q^{ck-m-1}r + n) - b(\nu)$$

where $\boldsymbol{e}_\nu^\top \boldsymbol{b}(N)$ is the output of the transducer when starting in state ν with input N and $b(\nu)$ is the final output at state ν.

Thus, we have

$$\begin{aligned} \frac{1}{h(k)} \sum_{n \in \mathcal{N}_k} \mathcal{T}(n) &= \frac{1}{h(k)} \sum_{0 \le n < h(k)} \mathcal{T}(N_k + q^{ck-m-1}r + n) \\ &= \boldsymbol{e}_\nu^\top \boldsymbol{b}(N_k q^{-ck+1}) - b(\nu) + \frac{1}{h(k)} \sum_{n < h(k)} \mathcal{T}(q^{ck-m-1}r + n) \end{aligned}$$

where only the first summand depends on $L_f(k)$ and hence on f.

Taking the difference in (2.2), there is a second way of computing the sum in (2.61). Using the periodicity and continuity of $\Psi_1(x)$ yields

$$\begin{aligned}\sum_{n\in\mathcal{N}_k}\mathcal{T}(n) &= (N_k+q^{ck-m-1}r)e_{\mathcal{T}}(z_k-y_k)+h(k)e_{\mathcal{T}}(x+k+L_f(k))\\ &\quad+(N_k+q^{ck-m-1}r)(\Psi_1(z_k)-\Psi_1(y_k))\\ &\quad+h(k)\Psi(x)+o(h(k)).\end{aligned}\tag{2.62}$$

Next, we use our assumption that Ψ_1 is differentiable at x to replace the difference by the derivative

$$\Psi_1(z_k)-\Psi_1(y_k)=\Psi_1'(x)(z_k-y_k)+o(|z_k-x|)+o(|x-y_k|).$$

Now, we insert this into (2.62), divide by $h(k)$ and obtain

$$\frac{1}{h(k)}\sum_{n\in\mathcal{N}_k}\mathcal{T}(n)=\frac{e_{\mathcal{T}}}{\log q}+e_{\mathcal{T}}(x+k+L_f(k))+\frac{1}{\log q}\Psi_1'(x)+\Psi_1(x)+o(1).$$

Thus, we have the following equality

$$\begin{aligned}\boldsymbol{e}_\nu^\top\boldsymbol{b}(N_kq^{-ck+1})-b(\nu)+\frac{1}{h(k)}\sum_{n<h(k)}\mathcal{T}(q^{ck-m-1}r+n)&\\ =\frac{e_{\mathcal{T}}}{\log q}+e_{\mathcal{T}}(x+k+L_f(k))+\frac{1}{\log q}\Psi_1'(x)+\Psi_1(x)+o(1)&\end{aligned}$$

twice, for $f\in\{0,1\}$. Subtracting these two from each other yields

$$\boldsymbol{e}_\nu^\top\boldsymbol{b}(q^{x_k+k+2})-\boldsymbol{e}_\nu^\top\boldsymbol{b}(q^{x_k+k+1})=e_{\mathcal{T}}+o(1).$$

Since the left-hand side is an integer, but the right-hand side is not for k large enough, this contradicts our assumption that Ψ_1 is differentiable at x. □

2.6. Recursions—Proof of Theorem 2.4

In this section, we construct a transducer associated to the sequence defined by the recursion in (2.9). All inequalities, maxima and minima in this section are considered coordinate-wise.

Define the function $A\colon\mathbb{N}_0^d\to\mathbb{N}_0^d\cup\{\infty\}$ by

$$A(q^\kappa\boldsymbol{n}+\boldsymbol{\lambda})=\begin{cases}q^\kappa\boldsymbol{n}+\boldsymbol{r}_{\boldsymbol{\lambda}} & \text{if } q^\kappa\boldsymbol{n}+\boldsymbol{r}_{\boldsymbol{\lambda}}\geq 0,\\ \infty & \text{else}\end{cases}$$

for $0\leq\boldsymbol{\lambda}<q^\kappa\mathbf{1}$ and $\boldsymbol{n}\geq 0$. So, if $A(\boldsymbol{n})<\infty$, then the recursion (2.9) can be used for this argument because the argument on the right-hand side is non-negative, i.e., $a(\boldsymbol{n})=a(A(\boldsymbol{n}))+t_{\boldsymbol{n}\bmod q^\kappa}$.

First, we construct a non-deterministic transducer $\widetilde{\mathcal{T}}$. A priori, it has an infinite number of states; later, we will prove that only finitely many of them are accessible. We then simplify it to obtain a finite, deterministic, subsequential, complete transducer $\mathcal{T}$.

The set of states of $\widetilde{\mathcal{T}}$ is

$$\{(\boldsymbol{l},j)_F\mid\boldsymbol{l}\in\mathbb{Z}^d,j\in\mathbb{N}_0\}\cup\{(\boldsymbol{l},j)_N\mid\boldsymbol{l}\in\mathbb{Z}^d,j\in\mathbb{N}_0\}.$$

The initial state is $(0,0)_F$; all states $(\boldsymbol{l},j)_F$ are final states with final output $a(\boldsymbol{l})$ if $\boldsymbol{l}\geq 0$ and final output 0 otherwise[5]. As an abbreviation, we will frequently speak about "a state $(\boldsymbol{l},j)$" if we do not want to distinguish between $(\boldsymbol{l},j)_F$ and $(\boldsymbol{l},j)_N$. We call $\boldsymbol{l}$ the *carry* and j the *level* of the state $(\boldsymbol{l},j)$. A state $(\boldsymbol{l},j)_F$ is called *simple*, if it is final, $\boldsymbol{l}\geq 0$ and $j\leq\kappa$.

There are two types of transitions in $\widetilde{\mathcal{T}}$, *recursion transitions* and *storing transitions*. Each state is either the origin of one recursion transition or of q^d storing transitions.

There is a recursion transition leaving $(\boldsymbol{l},j)$ if

- $j\geq\kappa$ and
- $A(q^j\boldsymbol{n}+\boldsymbol{l})<\infty$ for all $\boldsymbol{n}\geq 0$ with $\boldsymbol{n}\neq 0$.

In that case, we write $\boldsymbol{l}=q^\kappa\boldsymbol{s}+\boldsymbol{\lambda}$ for a $0\leq\boldsymbol{\lambda}<q^\kappa\mathbf{1}$ and the transition leads to the state $(\boldsymbol{l}',j')_N$ with $j'=\kappa_{\boldsymbol{\lambda}}+j-\kappa$ and $\boldsymbol{l}'=q^{\kappa_{\boldsymbol{\lambda}}}\boldsymbol{s}+\boldsymbol{r}_{\boldsymbol{\lambda}}$. The input label is empty, the output label is $t_{\boldsymbol{\lambda}}$. Thus

$$A(q^j\boldsymbol{n}+\boldsymbol{l})=q^{j'}\boldsymbol{n}+\boldsymbol{l}' \tag{2.63}$$

for $\boldsymbol{n}\geq 0$ with $\boldsymbol{n}\neq 0$. Note that (2.63) holds for $\boldsymbol{n}=0$ if and only if $\boldsymbol{l}\geq 0$ and $\boldsymbol{l}'\geq 0$.

Otherwise, there are storing transitions from $(\boldsymbol{l},j)$ to $(q^j\boldsymbol{\varepsilon}+\boldsymbol{l},j+1)_F$ with input $\boldsymbol{\varepsilon}$ and output 0 for all $0\leq\boldsymbol{\varepsilon}<q\mathbf{1}$.

We now define the classes $F_1,\ldots,F_K$ announced in Section 2.2.6. For each accessible cycle in $\widetilde{\mathcal{T}}$ with simple states and input 0, the carries of its states form one of these classes. The other classes are the singletons of those carries $\boldsymbol{l}\geq 0$ in the accessible part of $\widetilde{\mathcal{T}}$ with $A(\boldsymbol{l})=\infty$. These sets will turn out to be disjoint by Lemma 2.6.6 and the finiteness of K will follow from the finiteness of the accessible part of $\widetilde{\mathcal{T}}$ (Lemma 2.6.4).

Remark 2.6.1. We also give a combinatorial description of those classes $F_1,\ldots,F_K$ which do not come from cycles in $\widetilde{\mathcal{T}}$: Let $\boldsymbol{l}\geq 0$ be a carry of an accessible state of $\widetilde{\mathcal{T}}$. Then $A(\boldsymbol{l})=\infty$ if and only if there is a recursion transition from some $(\boldsymbol{l},j)$ to some $(\boldsymbol{l}',j')$ with $\boldsymbol{l}'\not\geq 0$.

Proof. Let $(\boldsymbol{l},j_0)$ be any accessible state with carry $\boldsymbol{l}$. We use the longest path with input 0 using storing transitions only to arrive in some state $(\boldsymbol{l},j)$—again, finiteness of this process will follow from the finiteness of the accessible part and the fact that the levels increase along storing transitions. As there is no storing transition leaving $(\boldsymbol{l},j)$ by construction, there is a recursion transition from $(\boldsymbol{l},j)$ to some $(\boldsymbol{l}',j')$. By the remark following (2.63), $\boldsymbol{l}'=A(\boldsymbol{l})$ or $\boldsymbol{l}'\not\geq 0$. □

As usual, if reaching a state which is the origin of a transition with empty input, the process may stay in that state or may continue to the destination state writing the output of the transition without reading an input. This is the reason why the transducer is non-deterministic.

Note that in our case, transitions with empty input (i.e., recursion transitions) lead to non-final states and transitions with non-empty input (i.e., storing transitions) lead to final states. Combined with the fact that each state is either the origin of one recursion transition or of q^d storing transitions, processing an input is in fact deterministic: For every admissible input—we do not allow leading zeros—, there exists exactly one path leading from the initial state to a final state with the given input. This will enable us to simplify the transducer $\widetilde{\mathcal{T}}$ to a deterministic transducer $\mathcal{T}$ later on.

[5]In fact, we will prove that a path with valid input will never end in a state $(\boldsymbol{l},j)_F$ with $\boldsymbol{l}\not\geq 0$, but the framework of subsequential transducers requires us to specify a final output even in that case. The non-final states $(\boldsymbol{l},j)_N$ will disappear in the reduction to $\mathcal{T}$ anyway.

We need the property that the carries of accessible states are not "too negative":

Lemma 2.6.2. *(1) If $(\boldsymbol{l}, j)$ is an accessible state, then*

$$q^j \boldsymbol{n} + \boldsymbol{l} \geq 0 \tag{2.64}$$

holds for all $\boldsymbol{n} \geq 0$ with $\boldsymbol{n} \neq 0$.

(2) If $d \geq 2$ and $(\boldsymbol{l}, j)$ is an accessible state, then

$$\boldsymbol{l} \geq 0.$$

(3) Any accessible transition with input $\boldsymbol{\varepsilon} \neq 0$ leads to a state $(\boldsymbol{l}, j)$ with $\boldsymbol{l} \geq 0$.

(4) If $d = 1$ and (l, j) is an accessible state, then

$$l \geq l_{\min} = \min_{\lambda} \left\{ 0, \frac{-1 + \frac{r_\lambda}{q^{\kappa_\lambda}}}{\frac{1}{q^{\kappa_\lambda}} - \frac{1}{q^{\kappa}}} \right\}.$$

Proof. The first assertion is easily shown by induction and (2.63). The second assertion follows by induction and from the assumption that $r_{\boldsymbol{\lambda}} \geq 0$ holds for all $\boldsymbol{\lambda}$. To prove the third assertion, we use (2.64) on the originating state of the transition.

The last assertion is shown by induction. It is clearly valid in the initial state. For storing transitions, the value of l is non-decreasing. If there is a recursion transition from some (l, j) to some $(l', j')_N$, we have

$$\begin{aligned} l' &= q^{\kappa_\lambda} \left\lfloor \frac{l}{q^{\kappa}} \right\rfloor + r_\lambda \geq q^{\kappa_\lambda} \left(\frac{l}{q^{\kappa}} - 1 + \frac{r_\lambda}{q^{\kappa_\lambda}} \right) \\ &\geq q^{\kappa_\lambda} \left(\frac{l_{\min}}{q^{\kappa}} + l_{\min} \left(\frac{1}{q^{\kappa_\lambda}} - \frac{1}{q^{\kappa}} \right) \right) = l_{\min}. \end{aligned}$$

□

As leading zeros are not allowed, the last transition in the computation path of any valid input has input $\boldsymbol{\varepsilon} \neq 0$ and thus leads to a state with a non-negative carry.

For our further investigations and finally the correctness proof, we need a suitable invariant:

Lemma 2.6.3. *Consider a path from $(\boldsymbol{l}, j)$ to $(\boldsymbol{l}', j')$ with input label $\boldsymbol{\varepsilon}_{m-1} \dots \boldsymbol{\varepsilon}_0$, output label $\delta_{m'-1} \dots \delta_0$ using L recursion transitions and $\boldsymbol{n} \geq 0$. Thus m' is the number of transitions and $m = m' - L$ is the number of storing transitions.*

If $\boldsymbol{n} \neq 0$ or if the last transition is a storing transition with non-zero input $\boldsymbol{\varepsilon}_{m-1}$, then

$$A^L(q^j(q^m \boldsymbol{n} + (\boldsymbol{\varepsilon}_{m-1} \dots \boldsymbol{\varepsilon}_0)_q) + \boldsymbol{l}) = q^{j'} \boldsymbol{n} + \boldsymbol{l}', \tag{2.65}$$

and, if the recursion (2.9) is well-posed,

$$a(q^j(q^m \boldsymbol{n} + (\boldsymbol{\varepsilon}_{m-1} \dots \boldsymbol{\varepsilon}_0)_q) + \boldsymbol{l}) = a(q^{j'} \boldsymbol{n} + \boldsymbol{l}') + \sum_{k=0}^{m'-1} \delta_k. \tag{2.66}$$

Proof. First consider the case that the path consists of a single transition. If it is a storing transition, then $L = 0$, $m = 1$, and all assertions follow from the definition and Lemma 2.6.2. On the other hand, if the transition is a recursion transition, we have $L = 1$, $m = 0$, and all assertions again follow from the definition, Lemma 2.6.2 and (2.63).

By induction on the length of the path, we obtain (2.65) and (2.66). □

We are now able to prove the finiteness of the accessible part.

Lemma 2.6.4. *The transducer has a finite number of accessible states.*

PROOF. For a recursion transition from $(\boldsymbol{l}, j)$ to $(\boldsymbol{l}', j')_N$, we have $j > j'$. Thus, there are no infinite paths consisting only of recursion transitions. In particular, there exist no cycles of recursion transitions.

For $d = 1$, let $J \geq \kappa$ be minimal such that $q^{J-\kappa} \geq -\lfloor \frac{l_{\min}}{q^\kappa} \rfloor - \min_\lambda q^{-\kappa_\lambda} r_\lambda$. Then $A(q^j + l) < \infty$ holds for all accessible states (l, j) with $j \geq J$. This implies $j \leq J$ for all accessible states (l, j). For $d \geq 2$, we have $j \leq \kappa =: J$ for all accessible states $(\boldsymbol{l}, j)$. Thus there are at most J consecutive recursion transitions.

To prove that only finitely many states are accessible, we introduce the notion of heights of states: The height of a state $(\boldsymbol{l}, j)$ is defined to be $\boldsymbol{h} = \boldsymbol{l} q^{-j}$. If there exists a storing transition from $(\boldsymbol{l}, j)$ of height $\boldsymbol{h}$ to $(\boldsymbol{l}', j')_F$ of height $\boldsymbol{h}'$, we have $\frac{1}{q}\boldsymbol{h} \leq \boldsymbol{h}' \leq \frac{1}{q}\boldsymbol{h} + \mathbf{1}$. If there exists a recursion transition from $(\boldsymbol{l}, j)$ of height $\boldsymbol{h}$ to $(\boldsymbol{l}', j')_N$ of height $\boldsymbol{h}'$, we have $\boldsymbol{h} + \boldsymbol{s}^- - \mathbf{1} \leq \boldsymbol{h}' \leq \boldsymbol{h} + \boldsymbol{s}^+$ where $\boldsymbol{s}^+ = \max_{\boldsymbol{\lambda}}\{\boldsymbol{r}_{\boldsymbol{\lambda}} q^{-\kappa_{\boldsymbol{\lambda}}}, 0\}$ and $\boldsymbol{s}^- = \min_{\boldsymbol{\lambda}}\{\boldsymbol{r}_{\boldsymbol{\lambda}} q^{-\kappa_{\boldsymbol{\lambda}}}, 0\}$.

Assume that there is a path from $(\boldsymbol{l}, j)$ of height $\boldsymbol{h}$ to $(\boldsymbol{l}', j')$ of height $\boldsymbol{h}'$ with $L \leq J$ recursion transitions and one storing transition (in this order). Then we have

$$\frac{1}{q}\boldsymbol{h} + \frac{J}{q}(\boldsymbol{s}^- - \mathbf{1}) \leq \boldsymbol{h}' < \frac{1}{q}\boldsymbol{h} + \frac{J}{q}\boldsymbol{s}^+ + \mathbf{1}.$$

We can subdivide every path in the transducer starting with the initial state into a sequence of such paths and a final path consisting of only recursion transitions. Let $\boldsymbol{h}_m$ be the sequence of heights of the states where the subpaths starts. Then, we have

$$\frac{1}{q}\boldsymbol{h}_m + \frac{J}{q}(\boldsymbol{s}^- - \mathbf{1}) \leq \boldsymbol{h}_{m+1} < \frac{1}{q}\boldsymbol{h}_m + \frac{J}{q}\boldsymbol{s}^+ + \mathbf{1}.$$

Iteration leads to

$$\frac{J(\boldsymbol{s}^- - \mathbf{1})}{q-1} \leq \boldsymbol{h}_m \leq \frac{J\boldsymbol{s}^+ + q\mathbf{1}}{q-1}$$

for all m. Therefore, the height $\boldsymbol{h}$ of an accessible state is bounded. Since $0 \leq j \leq J$ is also bounded, the integer carry $\boldsymbol{l} = q^j \boldsymbol{h}$ of an accessible state $(\boldsymbol{l}, j)$ can only take finitely many different values. The accessible part of the transducer is thus finite. □

Lemma 2.6.5 ([46])**.** *Let $\mathcal{P}$ be an infinite path with input zero starting at some state of level j such that all of its states have non-negative carries. Then, after at most j transitions, it reaches a state $(\boldsymbol{l}_0, \kappa)$. From that point on, it only passes through simple states, namely*

$$\begin{aligned}(\boldsymbol{l}_0, \kappa), (\boldsymbol{l}_1, j_1)_N, (\boldsymbol{l}_1, j_1 + 1)_F, \dots, (\boldsymbol{l}_1, \kappa)_F,\\ (\boldsymbol{l}_2, j_2)_N, (\boldsymbol{l}_2, j_2 + 1)_F, \dots, (\boldsymbol{l}_2, \kappa)_F,\\ (\boldsymbol{l}_3, j_3)_N, (\boldsymbol{l}_3, j_3 + 1)_F, \dots, (\boldsymbol{l}_3, \kappa)_F,\\ \dots\end{aligned}$$

where $\boldsymbol{l}_i = A(\boldsymbol{l}_{i-1})$ and $j_i = \kappa_{\boldsymbol{l}_{i-1} \bmod q^\kappa}$ for $i \geq 1$.

PROOF. Denote the first state of $\mathcal{P}$ by $(\boldsymbol{l}, j)$.

First, assume that $j \geq \kappa$. As storing transitions always increase the level and the levels are bounded by Lemma 2.6.4, the path has to contain at least one recursion transition. Thus the path starts with $k \geq 0$ storing transitions leading from $(\boldsymbol{l}, j)$ to $(\boldsymbol{l}, j + k)$, followed by a recursion transition from $(\boldsymbol{l}, j + k)$ to $(\boldsymbol{l}', j')$. By assumption, we have $\boldsymbol{l} \geq 0$ and $\boldsymbol{l}' \geq 0$. Thus $A(\boldsymbol{l}) = \boldsymbol{l}' \neq \infty$ by (2.63). Therefore, there is a recursion transition leaving $(\boldsymbol{l}, j)$, i.e., there

were no leading storing transitions. Recall that $j' < j$ holds for any recursion transition. We repeat the argument at most $j - \kappa$ times until we reach a simple state.

If we are in a simple state $(\boldsymbol{l}', j')$ with $j' < \kappa$, the next $\kappa - j'$ steps will be storing transitions, leading to $(\boldsymbol{l}', \kappa)$. This means that after at most j steps, we reach a state $(\boldsymbol{l}_0, \kappa)$.

We now apply the argument of the second paragraph again. Thus a recursion transition leads to $(\boldsymbol{l}_1, j_1)$ with $\boldsymbol{l}_1 = A(\boldsymbol{l}_0)$ and $j_1 = \kappa_{\boldsymbol{l}_0 \bmod q^\kappa}$.

The remainder of the lemma follows by induction. □

As an auxiliary structure for deciding the well-posedness of the recursion, we introduce the *recursion digraph* $\mathcal{R}$. It has set of vertices $\mathbb{N}_0^d$ and arcs $(\boldsymbol{n}, A(\boldsymbol{n}))$ with label $t_{\boldsymbol{n} \bmod q^\kappa}$ for all $\boldsymbol{n} \in \mathbb{N}_0^d$ with $A(\boldsymbol{n}) < \infty$. Thus $a(\boldsymbol{n})$ can be computed from the successor of $\boldsymbol{n}$ in $\mathcal{R}$ using the recursion (2.9). By definition, each vertex of $\mathcal{R}$ has out-degree 1 or 0. Each component of $\mathcal{R}$ is a functional digraph or a rooted tree (oriented towards the root).

If

$$\|\boldsymbol{n}\|_\infty > \max_{\boldsymbol{\lambda}} \frac{\|\boldsymbol{\lambda}\|_\infty + \|\boldsymbol{r}_{\boldsymbol{\lambda}}\|_\infty}{q^\kappa - q^{\kappa_{\boldsymbol{\lambda}}}},$$

we have

$$q^\kappa \|\boldsymbol{n}\|_\infty - \|\boldsymbol{\lambda}\|_\infty > q^{\kappa_{\boldsymbol{\lambda}}} \|\boldsymbol{n}\|_\infty + \|\boldsymbol{r}_{\boldsymbol{\lambda}}\|_\infty$$

and therefore

$$\|q^\kappa \boldsymbol{n} + \boldsymbol{\lambda}\|_\infty > \|q^{\kappa_{\boldsymbol{\lambda}}} + \boldsymbol{r}_{\boldsymbol{\lambda}}\|_\infty$$

for all $0 \le \boldsymbol{\lambda} < q^\kappa \mathbf{1}$. Thus we have $\|\boldsymbol{n}'\|_\infty < \|\boldsymbol{n}\|_\infty$ for all but finitely many arcs $(\boldsymbol{n}, \boldsymbol{n}')$ of $\mathcal{R}$.

Thus for every vertex of $\mathcal{R}$, there is a unique path starting in this vertex and leading to a vertex with out-degree 0 or a finite cycle.

From this description, it is clear that the recursion is well-posed if and only if

- the sum of the labels of each cycle in $\mathcal{R}$ is 0 and
- the set $\mathcal{I}$ consists of one element for every cycle in $\mathcal{R}$ as well as of the vertices with out-degree 0 in $\mathcal{R}$.

We now prove the essential connection between the recursive digraph and the transducer $\widetilde{\mathcal{T}}$. This also implies that the classes $F_1, \ldots, F_K$ are disjoint.

Lemma 2.6.6. *There exists a bijection between cycles in the recursive digraph $\mathcal{R}$ and accessible cycles in the transducer $\widetilde{\mathcal{T}}$ with input 0 and simple states. Corresponding cycles under this bijection have the same output sum and sum of labels.*

Proof. Let $\boldsymbol{n}_0, \ldots, \boldsymbol{n}_L = \boldsymbol{n}_0$ be a cycle in the recursive digraph with $\boldsymbol{n}_R \ge 0$ for all $0 \le R < L$.

Let k_0 be the length of the path $\mathcal{P}_0$ in $\widetilde{\mathcal{T}}$ starting in the initial state and reading the q-ary expansion of $\boldsymbol{n}_0$.

We determine the destinations of certain paths in the transducer associated with the cycle in the recursive digraph.

Statement 2.6.7. *Let $k \ge k_0$ and $\mathcal{P}$ be the path from the initial state $(0,0)$ to $(\boldsymbol{l}, j)$ of length k whose input label is the q-ary expansion of $\boldsymbol{n}_0$, padded with leading zeros. Assume that the number of recursion transitions in this path is $LQ + R$ for some $Q \ge 0$ and $0 \le R < L$. Then $\boldsymbol{l} = \boldsymbol{n}_R \ge 0$.*

PROOF OF STATEMENT 2.6.7. Let $k' = k-(LQ+R)$ be the number of storing transitions of $\mathcal{P}$. By (2.65), we have

$$A^{LQ+R}(q^{k'}\boldsymbol{n} + \boldsymbol{n}_0) = q^j \boldsymbol{n} + \boldsymbol{l} \tag{2.67}$$

for $\boldsymbol{n} \geq 0$, $\boldsymbol{n} \neq 0$.

Note that for $M \geq \kappa$ and $\boldsymbol{n} \equiv \boldsymbol{n}' \pmod{q^M}$ with $A(\boldsymbol{n}) < \infty$ and $A(\boldsymbol{n}') < \infty$, the definition of A implies $A(\boldsymbol{n}) \equiv A(\boldsymbol{n}') \pmod{q^{M-\kappa}}$.

Together with the definitions of $\boldsymbol{n}_R$ and the recursive digraph $\mathcal{R}$ as well as (2.67), this implies

$$\begin{aligned}\boldsymbol{n}_R = A^{LQ+R}(\boldsymbol{n}_0) &\equiv A^{LQ+R}(q^{k'+M}\mathbf{1} + \boldsymbol{n}_0)\\ &= q^{j+M}\mathbf{1} + \boldsymbol{l} \pmod{q^{k'+M-(LQ+R)\kappa}}\end{aligned}$$

for sufficiently large M. Coarsening yields

$$\boldsymbol{n}_R \equiv \boldsymbol{l} \pmod{q^{M-(LQ+R)\kappa}},$$

still valid for sufficiently large M. As $\boldsymbol{l}$ is bounded by Lemma 2.6.4, this implies $\boldsymbol{n}_R = \boldsymbol{l}$. □

Now, we conclude the proof of Lemma 2.6.6.

Let $\mathcal{P}$ be the infinite path in $\widetilde{\mathcal{T}}$ starting at the destination of $\mathcal{P}_0$ and reading zeros. By Lemma 2.6.5 applied to $\mathcal{P}$ together with Statement 2.6.7 applied to $\mathcal{P}_0$ concatenated with prefixes of $\mathcal{P}$, $\mathcal{P}$ leads to a cycle in $\widetilde{\mathcal{T}}$. Its states are simple and have carries $\boldsymbol{n}_0, \ldots, \boldsymbol{n}_{L-1}$ and levels determined by $\boldsymbol{n}_0, \ldots, \boldsymbol{n}_{L-1}$ as in Lemma 2.6.5.

This construction defines a map from the cycles of the recursive digraph $\mathcal{R}$ to the accessible cycles with input 0 in the transducer with simple states. This map is injective by construction. Under this map, the sum of the labels of the cycle in $\mathcal{R}$ equals the sum of output labels of the cycle in $\widetilde{\mathcal{T}}$ by construction.

On the other hand, let

$$\begin{aligned}&(\boldsymbol{n}_0, j_0), (\boldsymbol{n}_0, j_0+1), \ldots, (\boldsymbol{n}_0, \kappa),\\ &(\boldsymbol{n}_1, j_1), (\boldsymbol{n}_1, j_1+1), \ldots, (\boldsymbol{n}_1, \kappa), \ldots\\ &(\boldsymbol{n}_{L-1}, j_{L-1}), (\boldsymbol{n}_{L-1}, j_{L-1}+1), \ldots, (\boldsymbol{n}_{L-1}, \kappa),\\ &(\boldsymbol{n}_0, j_0)\end{aligned}$$

be an accessible cycle of simple states in the transducer with input 0. Lemma 2.6.5 yields $A(\boldsymbol{n}_R) = \boldsymbol{n}_{R+1 \bmod L} \geq 0$ for $0 \leq R < L$. Thus, this cycle in the transducer is the image of the cycle $\boldsymbol{n}_0, \ldots, \boldsymbol{n}_L = \boldsymbol{n}_0$ in the recursive digraph. Thus the map is surjective. □

To use Theorem 2.1, we simplify $\widetilde{\mathcal{T}}$ to obtain the deterministic transducer $\mathcal{T}$, that is one without transitions with empty input. As a first step, we remove all non-accessible states. By Lemma 2.6.4, this leaves us with finitely many states.

By Lemma 2.6.4 and the fact that recursion transitions decrease the level, the length of paths consisting of recursion transitions only is bounded. As a recursion transition always leads to a non-final state, processing an input never ends with a recursion transition.

Consider a recursion transition from $(\boldsymbol{l}, j)$ to $(\boldsymbol{l}', j')_N$ with output t such that no recursion transition originates in $(\boldsymbol{l}', j')_N$. For each transition originating in $(\boldsymbol{l}', j')_N$, say to some $(\boldsymbol{l}'', j'')_F$ with input $\boldsymbol{\varepsilon}$ and output t', we insert a storing transition from $(\boldsymbol{l}, j)$ to $(\boldsymbol{l}'', j'')_F$ with input $\boldsymbol{\varepsilon}$ and output $t + t'$. Then, the recursion transition from $(\boldsymbol{l}, j)$ to $(\boldsymbol{l}', j')_N$ is removed. The number of recursion transitions decreased by one and the new transducer generates the

same output as the old transducer. We repeat this process until there are no more recursion transitions. Then, all non-final states are inaccessible and are removed.

Proof of Theorem 2.4. By Lemma 2.6.6 and the characterization of well-posedness via the recursive digraph, the recursion (2.9) is well-posed if and only if $\mathcal{I}$ consists of exactly one representative of each of the sets F_j, $1 \le j \le K$, and if $\widetilde{\mathcal{T}}$ has no cycle with simple states, input 0 and non-vanishing output sum.

We now show that the cycles of simple states with input 0 in $\mathcal{T}$ are exactly the reductions of the cycles of simple states with input 0 in $\widetilde{\mathcal{T}}$. As a cycle with simple states and input 0 in $\widetilde{\mathcal{T}}$ does not have consecutive recursion transitions (cf. Lemma 2.6.5), it is reduced to a cycle with simple states in $\mathcal{T}$. On the other hand, consider a cycle of $\widetilde{\mathcal{T}}$ with input 0 containing a non-simple state. If there is a state of level $> \kappa$, the state with largest level is final and is not removed. If all states have level $\le \kappa$, then there are no two consecutive recursion transitions, so no negative carry is completely removed from the cycle in the reduction to $\mathcal{T}$. Therefore, such a cycle is not reduced to a cycle with simple states and input 0 in $\mathcal{T}$.

Therefore, the assertion on well-posedness is proved.

To prove correctness of the transducer, we use (2.65) with $(\boldsymbol{l}, j) = (0, 0)$, the joint q-ary expansion of $\boldsymbol{n}$ as input leading to some state $(\boldsymbol{l}', j')_F$ with output $\delta_{m'-1} \dots \delta_0$. By Lemma 2.6.2, we have $\boldsymbol{l}' \ge 0$ because the last transition is a storing transition with non-zero input. Thus by (2.66), $a(\boldsymbol{n}) = a(\boldsymbol{l}') + \sum_{k=0}^{m'-1} \delta_k$. As the final output of $(\boldsymbol{l}', j')_F$ is defined to be $a(\boldsymbol{l}')$, we obtain $\mathcal{T}(\boldsymbol{n}) = a(\boldsymbol{l}') + \sum_{k=0}^{m'-1} \delta_k = a(\boldsymbol{n})$, as requested. □

CHAPTER 3

Variances and Covariances in the Central Limit Theorem for the Output of a Transducer

In this chapter, we study the joint distribution of the input sum and the output sum of a deterministic transducer. Here, the input of this finite state machine is a uniformly distributed random sequence of real numbers.

We give a simple combinatorial characterization of transducers for which the output sum has bounded variance, and we also provide algebraic and combinatorial characterizations of transducers for which the covariance of input and output sum is bounded, so that the two are asymptotically independent.

The results of Theorem 3.3 have been implemented [51] in the open-source mathematics software system SageMath [95], based on its package for finite state machines, see [49] or Chapter 6. This code is included in SageMath 6.3.

The content of this chapter corresponds to [56], which appeared in the *European Journal of Combinatorics*. This is joint work with Clemens Heuberger and Stephan Wagner.

In Chapter 4, we generalize the results of this chapter to more than one output of the transducer and not independently identically distributed input sequences. While writing this thesis, this generalization was considered after [56] has appeared, thus the results are presented separately in two different chapters.

3.1. Introduction

We asymptotically investigate the two random variables *sum of the input* and *sum of the output* of a transducer for inputs of length n for $n \to \infty$. If these two random variables converge in distribution to independent random variables, then the transducer is called independent.

Our probability model is the equidistribution on all input sequences of a fixed length n. Under this probability model, the expected value of the sum of the input and the output are $e_1 n$ and $e_2 n + \mathcal{O}(1)$, respectively, for some constants e_1 and e_2. For the sum of the input, the expressions are exact without error term because the input letters are independent and identically distributed. Furthermore, under appropriate connectivity conditions, the variances and the covariance turn out to be $v_1 n$, $v_2 n + \mathcal{O}(1)$ and $cn + \mathcal{O}(1)$, respectively, for suitable constants v_1, v_2 and c. We investigate for which transducers one of the constants v_2 and c is zero.

A special case of the output sum is the Hamming weight, which is the number of non-zero elements of a sequence. To give an example of an independent transducer, we discuss the Hamming weight of the non-adjacent form as defined by Reitwiesner [89] in Example 3.2.5. In [61], Heuberger and Prodinger prove that the Hamming weights of the standard binary expansion and the non-adjacent form are asymptotically independent. The independent transducer computing these Hamming weights is shown in Figure 3.2.

We formally define our setting in Section 3.2. In Section 3.3, we state our main results. In Section 3.4, we present several examples where these main results are applied. In Section 3.5, we give the proofs of the theorems.

We give an algebraic description of independent transducers in Theorem 3.3. We also state there that the input sum and the output sum are asymptotically jointly normally distributed if the variance-covariance matrix is invertible. In Theorem 3.4, we present a combinatorial characterization of independent transducers.

In Section 3.4, we give a variety of examples of independent and dependent transducers and transducers with bounded and unbounded variance to illustrate our results. One of those examples is a transducer computing the minimal Hamming weight of τ-adic digit representations on a digit set $\mathcal{D}$. Building on the results of [42], we prove that the variance of the minimal Hamming weight is unbounded, which yields a central limit theorem.

In Section 3.5, we also prove an extension of the 2-dimensional Quasi-Power Theorem [44] to singular Hessian matrices as an auxiliary result.

3.2. Preliminaries

A *transducer* is defined to consist of a finite set of states $\{1, 2, \dots, S\}$, a finite input alphabet $\mathcal{A}_I \subseteq \mathbb{R}$, an output alphabet $\mathcal{A}_O \subseteq \mathbb{R}$, a set of transitions $\mathcal{E} \subseteq \{1, 2, \dots, S\}^2 \times \mathcal{A}_I$ with input labels in $\mathcal{A}_I$, output labels $\delta\colon \mathcal{E} \to \mathcal{A}_O$ and the initial state 1. The transducer is called *deterministic* if for all states s and input labels $\varepsilon \in \mathcal{A}_I$, there exists at most one state t such that $(s, t, \varepsilon) \in \mathcal{E}$. Furthermore, the transducer is said to be *subsequential* (cf. [91]) if it is deterministic, every state is final and it has a final output $a\colon \{1, 2, \dots, S\} \to \mathcal{A}_O$. A transducer is called *complete* if for every state s and digit $\varepsilon \in \mathcal{A}_I$, there is a transition from s to a state t with input label ε, i.e., $(s, t, \varepsilon) \in \mathcal{E}$.

Definition 3.2.1. A transducer is said to be *finally connected* if there exists a state which can be reached from any other state. The *final component* of such a transducer is defined to be the transducer induced by the set of states which can be reached from any other state. A finally connected transducer is said to be *finally aperiodic* if the underlying graph of the final component is aperiodic (i.e., the gcd of the lengths of all walks starting and ending at a given vertex is 1).

Remark 3.2.2. The final component of a transducer is a strongly connected component of the underlying graph of the transducer. If the underlying graph is strongly connected, then being finally aperiodic is equivalent to being aperiodic. We then call the transducer strongly connected and aperiodic. The final component of a complete transducer is complete itself.

In the following, we consider subsequential, complete, deterministic, finally connected, finally aperiodic transducers. We require that the input alphabet $\mathcal{A}_I$ has at least two elements. Throughout the chapter, we use ε for the input of a transition and δ for the output of a transition. We denote the number of states in the final component by N.

The input of the transducer is a sequence in $\mathcal{A}_I^*$. It is not important whether we read the input from right to left or in the other direction, we just have to fix it for one specific transducer. The output of the transducer is the sequence of output labels of the unique path starting at the initial state 1 with the given input as input label, together with the final output label of the final state of this path.

Let X_n be a uniformly distributed random variable on $\mathcal{A}_I^n$. Let $\mathsf{Output}(X_n)$ be the sum of the output sequence of the transducer if the input is X_n. Furthermore, let $\mathsf{Input}(X_n)$ be the

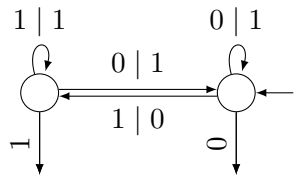


FIGURE 3.1. Subsequential, complete, strongly connected, aperiodic transducer from Example 3.2.3.

sum of the input sequence. Without loss of generality, we fix the direction of reading from right to left.

Example 3.2.3. The transducer in Figure 3.1 is a subsequential, complete, strongly connected, aperiodic transducer.

For example, when reading the input (110) from right to left, the transducer in Figure 3.1 writes the output (1101). The leftmost 1 in the output is the final output of the last state. The output sum is $\mathsf{Output}(110) = 3$.

We investigate the 2-dimensional random vector

$$\mathbf{\Omega}_n = (\mathsf{Input}(X_n), \mathsf{Output}(X_n))^\top$$

for $n \to \infty$, where $^\top$ denotes transposition. We will prove that each component of this random vector either converges in distribution to a normally distributed random variable or to a degenerate random variable. Here, a random variable is said to be *degenerate* if it is constant with probability 1. By definition, a degenerate random variable is independent of any other random variable. Thus, the variance of a degenerate random variable and the covariance of a degenerate and any other random variable are always 0.

For a finally connected, aperiodic transducer, the expected value and the variance of $\mathbf{\Omega}_n$ will turn out to be $(e_1, e_2)^\top n + \mathcal{O}(1)$, $(v_1, v_2)^\top n + \mathcal{O}(1)$, respectively, for suitable constants e_1, e_2, v_1 and v_2 (see Theorem 3.3). The covariance between the two coordinates will be $cn + \mathcal{O}(1)$ for some constant c. We call $\Sigma = \left(\begin{smallmatrix} v_1 & c \\ c & v_2 \end{smallmatrix}\right)$ the *asymptotic variance-covariance matrix* of $\mathbf{\Omega}_n = (\mathsf{Input}(X_n), \mathsf{Output}(X_n))^\top$. Its entries are called the *asymptotic variances* and the *asymptotic covariance*.

For transducers with output alphabet $\{0, 1\}$, the characterization of vanishing asympototic variance v_2 turns out to be particularly simple: All transitions of the final component have to have the same output (see Corollary 3.3.4). This output alphabet occurs naturally when only the Hamming weight (the number of non-zero elements) of an expansion is of interest.

For brevity, we introduce the notion of independent transducers.

Definition 3.2.4. A transducer is *independent* if the random vector $\mathbf{\Omega}_n$ converges in distribution to a random vector with two independent components, i.e., the sum of the input $\mathsf{Input}(X_n)$ and the sum of the output $\mathsf{Output}(X_n)$ are asymptotically independent random variables.

Example 3.2.5. In [61], Heuberger and Prodinger prove that the Hamming weight of the standard binary expansion and the Hamming weight of the non-adjacent form are asymptotically independent. The non-adjacent form is the unique digit expansion with digits $\{-1, 0, 1\}$,

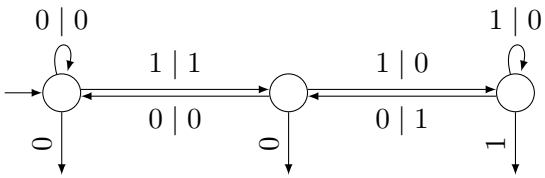


FIGURE 3.2. Transducer to compute the Hamming weight of the non-adjacent form.

base 2 and the syntactical rule that at least one of any two adjacent digits has to be 0. It has minimal Hamming weight among all digit expansions with digits $\{-1, 0, 1\}$ in base 2.

The transducer in Figure 3.2 computes the Hamming weight of the non-adjacent form when reading the binary expansion from right to left. The transducer is a slight simplification of the one in, e.g., [61], taking into account that we are only interested in the Hamming weight. Thus, the transducer in Figure 3.2 is an example of an independent transducer by the results in [61].

3.3. Main Results

In this section, we state the main theorems and corollaries describing independent transducers and transducers with bounded variance. First, we investigate transducers with bounded variance. Then, we give an algebraic description and a combinatorial characterization of independent transducers. All proofs can be found in Section 3.5.

3.3.1. Bounded Variance and Singular Asymptotic Variance-Covariance Matrix. We give a combinatorial characterization of transducers whose output sum has asymptotic variance 0. We also give a combinatorial description of transducers with singular asymptotic variance-covariance matrix. These characterizations are given in terms of cycles and closed walks of directed graphs.

As usual, a *cycle* is a strongly connected digraph such that every vertex has out-degree 1. A *closed walk* is an alternating sequence of vertices and edges $(s_1, e_1, s_2, \ldots, s_{n+1} = s_1)$ such that e_j is an edge from s_j to s_{j+1}.

For a function g and a walk C of the underlying graph of the transducer, we define

$$g(C) = \sum_{e \in C} g(e)$$

taking multiplicities into account. Here, the function g is either the constant function $\mathbb{1}(e) = 1$, the input $\varepsilon(e)$ or the output $\delta(e)$ of the transition e.

Theorem 3.1 ([102])**.** *For a subsequential, complete, finally connected and finally aperiodic transducer with an arbitrary finite input alphabet $\mathcal{A}_I$, the following assertions are equivalent:*

(a) The asymptotic variance v_2 of the output sum is 0.

(b) There exists a state s of the final component and a constant $k \in \mathbb{R}$ such that

$$\delta(C) = k\mathbb{1}(C)$$

holds for every closed walk C of the final component visiting the state s exactly once.

(c) There exists a constant $k \in \mathbb{R}$ such that

$$\delta(C) = k\mathbb{1}(C)$$

holds for every directed cycle C of the final component of the transducer $\mathcal{T}$.

In that case, $kn + \mathcal{O}(1)$ is the expected value of the output sum and Statement (b) holds for all states s of the final component.

We want to emphasize that only cycles and closed walks of the final component are considered in this theorem (see also Remark 3.3.7). The proofs can be found in Section 3.5.3.

In the case of a strongly connected transducer, the equivalent conditions of Theorem 3.1 will be shown to be equivalent to another condition which, at first glance, seems to be even stronger.

Definition 3.3.1. The output sum of a transducer is called *quasi-deterministic* if there is a constant $k \in \mathbb{R}$ such that

$$\mathsf{Output}(X_n) = kn + \mathcal{O}(1)$$

holds for all n and all inputs.

We now characterize quasi-deterministic output sums. In weakly connected graphs, it turns out that being "quasi-deterministic" is a stronger notion than the conditions in Theorem 3.1.

Theorem 3.2. *Let $\mathcal{T}$ be a subsequential, complete transducer whose underlying graph is weakly connected. Then the following two assertions are equivalent:*

(d) There exists a constant $k \in \mathbb{R}$ such that the random variable $\mathsf{Output}(X_n)$ is quasi-deterministic with value $kn + \mathcal{O}(1)$.

(e) There exists a constant $k \in \mathbb{R}$ such that

$$\delta(C) = k\mathbb{1}(C)$$

holds for every directed cycle C of the transducer.

This result and the following corollaries are proved in Section 3.5.3. By comparing statements (c) of Theorem 3.1 and (e) of Theorem 3.2, it is obvious that in strongly connected transducers, all these statements are actually equivalent.

Corollary 3.3.2. *Let $\mathcal{T}$ be a subsequential, complete, strongly connected, aperiodic transducer. Then the asymptotic variance v_2 of the output sum is zero if and only if the output sum is a quasi-deterministic random variable.*

Remark 3.3.3. If the transducer is not strongly connected (so that there are states that do not belong to the final component), the output sum can have bounded variance without being quasi-deterministic. A simple example is a transducer that counts the number of 1s in a binary string before the first 0. In such a case, however, the transducer formed only by the final component still needs to have quasi-deterministic output sum.

When considering the special case of the Hamming weight, bounded variance only occurs in trivial cases:

Corollary 3.3.4. *For $\mathcal{A}_O = \{0, 1\}$, the only output weights of the final component with asymptotic variance $v_2 = 0$ are $(0, \dots, 0)$ and $(1, \dots, 1)$.*

The following corollary of Theorem 3.1 gives a combinatorial characterization of transducers whose asymptotic variance-covariance matrix is singular.

Corollary 3.3.5 ([46, 102])**.** *Let $\mathcal{T}$ be a complete, subsequential, finally connected, finally aperiodic transducer whose input alphabet has at least size 2. Then the asymptotic variance-covariance matrix Σ has rank 1 if and only if there exist a, $b \in \mathbb{R}$ with*

$$\delta(C) = a\mathbb{1}(C) + b\varepsilon(C) \tag{3.1}$$

for all cycles C of the final component.

In that case, the constants are $a = -\frac{c}{v_1}e_1 + e_2$ and $b = \frac{c}{v_1}$.

Furthermore, the random variables $\mathsf{Input}(X_n)$ and $\mathsf{Output}(X_n)$ are asymptotically perfectly positively or negatively correlated (i.e., they have asymptotic correlation coefficient ± 1) if and only if (3.1) holds with $b \neq 0$.

3.3.2. Algebraic Description of Independent Transducers. For giving an algebraic description of independent transducers, we define transition matrices of the transducer.

Definition 3.3.6. For $\varepsilon \in \mathcal{A}_I$, let a *transition matrix* $M_\varepsilon(y)$ of the final component be the $N \times N$-matrix whose entry (s,t) is y^δ if there is a transition from state s to state t in the final component with input ε and output δ, and 0 otherwise.

Similarly, let W_ε be the transition matrix of the whole transducer. The ordering of the states is considered to be fixed in such a way that the initial state 1 is the first state and W_ε has the block structure

$$\begin{pmatrix} * & * \\ 0 & M_\varepsilon \end{pmatrix} \tag{3.2}$$

where $*$ are matrices with arbitrary entries. If the transducer is strongly connected, the matrices $*$ are not present (they have 0 rows).

Theorem 3.3. *Let $\mathcal{T}$ be a complete, subsequential, finally connected, finally aperiodic transducer, and let the transition matrices of the final component be $M_\varepsilon(y)$ for $\varepsilon \in \mathcal{A}_I$. Set*

$$f(x,y,z) = \det\Big(I - \frac{z}{|\mathcal{A}_I|}\sum_{\varepsilon\in\mathcal{A}_I} x^\varepsilon M_\varepsilon(y)\Big).$$

Then the random variables $\mathsf{Input}(X_n)$ and $\mathsf{Output}(X_n)$ have the expected values, variances and covariance

$$\begin{aligned} \mathbb{E}(\mathsf{Input}(X_n)) &= e_1 n, \\ \mathbb{E}(\mathsf{Output}(X_n)) &= e_2 n + \mathcal{O}(1), \\ \mathbb{V}(\mathsf{Input}(X_n)) &= v_1 n, \\ \mathbb{V}(\mathsf{Output}(X_n)) &= v_2 n + \mathcal{O}(1), \\ \mathrm{Cov}(\mathsf{Input}(X_n), \mathsf{Output}(X_n)) &= cn + \mathcal{O}(1) \end{aligned} \tag{3.3}$$

with

$$\begin{aligned} e_1 &= \frac{f_x}{f_z}\Big|_{\mathbf{1}}, \\ e_2 &= \frac{f_y}{f_z}\Big|_{\mathbf{1}}, \\ v_1 &= \frac{1}{f_z^3}(f_x^2(f_{zz}+f_z) + f_z^2(f_{xx}+f_x) - 2f_xf_zf_{xz})\Big|_{\mathbf{1}}, \\ v_2 &= \frac{1}{f_z^3}(f_y^2(f_{zz}+f_z) + f_z^2(f_{yy}+f_y) - 2f_yf_zf_{yz})\Big|_{\mathbf{1}}, \end{aligned}$$

$$c = \frac{1}{f_z^3}(f_x f_y (f_{zz} + f_z) + f_z^2 f_{xy} - f_y f_z f_{xz} - f_x f_z f_{yz})\Big|_{\mathbf{1}}$$

where $\mathbf{1} = (1,1,1)^\top$ *and* $f_z(\mathbf{1}) \neq 0$.

The constants e_1 *and* v_1 *can also be expressed as*

$$e_1 = \frac{1}{|\mathcal{A}_I|}\sum_{\varepsilon\in\mathcal{A}_I}\varepsilon, \qquad v_1 = \frac{1}{|\mathcal{A}_I|}\sum_{\varepsilon\in\mathcal{A}_I}\varepsilon^2 - \Big(\frac{1}{|\mathcal{A}_I|}\sum_{\varepsilon\in\mathcal{A}_I}\varepsilon\Big)^2. \tag{3.4}$$

The random vector $\mathbf{\Omega}_n$ *is asymptotically jointly normally distributed if and only if the asymptotic variance-covariance matrix* Σ *is regular.*

The transducer $\mathcal{T}$ *is independent if and only if*

$$(f_x f_y (f_{zz} + f_z) + f_z^2 f_{xy} - f_y f_z f_{xz} - f_x f_z f_{yz})\big|_{\mathbf{1}} = 0 \tag{3.5}$$

or, equivalently,

$$(e_1 f_y (f_{zz} + f_z) + f_z f_{xy} - f_y f_{xz} - e_1 f_z f_{yz})\big|_{\mathbf{1}} = 0. \tag{3.6}$$

The proof of this theorem is in Section 3.5.1. This result has been implemented as the method

`FiniteStateMachine.asymptotic_moments()`

in the mathematics software system SageMath, cf. [51], using the finite state machines package described in [49] and Chapter 6.

Remark 3.3.7. Neither the final output nor the non-final components influence the asymptotic result because it only depends on $f(x,y,z)$ and thus on the transitions of the final component.

Now we consider the following "inverse" problem: Given the underlying graph and the input digits of the transducer; how can we choose the output labels such that the transducer is independent?

Let $(a_1,\dots,a_N)$ be the output labels of the final component of the transducer. We say, as usual, that a linear equation is homogeneous if the zero vector is a solution. Then (3.5) is a linear, homogeneous equation in $a_1,\dots,a_N$ with real coefficients. The equation is linear because the variables a_i only occur linearly in the exponents of y and there are only first derivatives with respect to y in the covariance condition (3.5). Furthermore, (3.5) is homogeneous because all derivatives with respect to y (and maybe other additional variables) at $(x,y,z)^\top = \mathbf{1}$ are homogeneous. A solution of this linear, homogeneous equation corresponds to an independent transducer.

Let us first consider the situation where all outputs are equal to 1. Then, the determinant $f(x,y,z)$ consists of monomials $x^a y^b z^b$ with $a \in \mathbb{R}$ and $b \in \mathbb{Z}$. Therefore, we obtain

$$\begin{aligned} f_y|_{\mathbf{1}} &= f_z|_{\mathbf{1}}, \\ f_{xy}|_{\mathbf{1}} &= f_{xz}|_{\mathbf{1}}, \\ f_{yz}|_{\mathbf{1}} &= f_{zz} + f_z|_{\mathbf{1}}, \end{aligned}$$

and it follows that (3.5) and (3.6) are satisfied. This means that a constant output $(k,\dots,k)$ for $k \in \mathcal{A}_O$ is always a trivial solution to these equations because (3.5) is homogeneous.

But for these trivial solutions, the sum of the output is an asymptotically degenerate random variable. Hence, we are not really interested in the independent transducers given by these solutions.

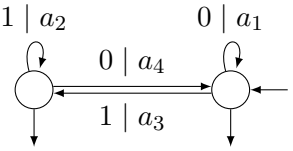


FIGURE 3.3. Transducer of Example 3.3.8.

Example 3.3.8. In Figure 3.3, we have a transducer with variable output weights a_1, a_2, a_3 and a_4. We do not give the final output labels as they do not influence the asymptotic result. In this example, (3.5) simplifies to

$$-a_1 + a_2 = 0.$$

3.3.3. Combinatorial Characterization of Independent Transducers. We connect the derivatives of $f(x, y, z)$ with a weighted sum of subgraphs of the underlying graph. Thus, in Theorem 3.4, we can give a combinatorial description of (3.5).

Definition 3.3.9. We define the following types of directed graphs as subgraphs of the final component of the transducer.

- A *rooted tree* is a weakly connected digraph with one vertex which has out-degree 0, while all other vertices have out-degree 1. The vertex with out-degree 0 is called the *root* of the tree.
- A *functional digraph* is a digraph whose vertices have out-degree 1. Each component of a functional digraph consists of a directed cycle and some trees rooted at vertices of the cycle. For a functional digraph D, let $\mathcal{C}_D$ be the set of all cycles of D.

Definition 3.3.10. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be the sets of all spanning subgraphs of the final component of the transducer $\mathcal{T}$ which are functional digraphs and have one and two components, respectively.

For functions g and $h\colon \mathcal{E} \to \mathbb{R}$, we define

$$\begin{aligned} g(\mathcal{D}_1) &= \sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} g(C), \\ gh(\mathcal{D}_1) &= \sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} g(C)h(C), \\ gh(\mathcal{D}_2) &= \sum_{D\in\mathcal{D}_2} \sum_{C_1\in\mathcal{C}_D} \sum_{\substack{C_2\in\mathcal{C}_D \\ C_2\neq C_1}} g(C_1)h(C_2). \end{aligned}$$

With these definitions, we give a combinatorial characterization of independent transducers.

Theorem 3.4. *Let $\mathcal{T}$ be a complete, subsequential, finally connected, finally aperiodic transducer.*

Then the random variables $\mathsf{Input}(X_n)$ *and* $\mathsf{Output}(X_n)$ *have the expected values given by* (3.3)*, where the constants are*

$$e_1 = \frac{\varepsilon(\mathcal{D}_1)}{\mathbb{1}(\mathcal{D}_1)},$$

$$e_2 = \frac{\delta(\mathcal{D}_1)}{\mathbb{1}(\mathcal{D}_1)}.$$

The variances and the covariance are given by (3.3)*, with the constants*

$$\begin{aligned} v_1 &= \frac{1}{\mathbb{1}(\mathcal{D}_1)}\big((\varepsilon - e_1\mathbb{1})(\varepsilon - e_1\mathbb{1})(\mathcal{D}_1) - (\varepsilon - e_1\mathbb{1})(\varepsilon - e_1\mathbb{1})(\mathcal{D}_2)\big),\\ v_2 &= \frac{1}{\mathbb{1}(\mathcal{D}_1)}\big((\delta - e_2\mathbb{1})(\delta - e_2\mathbb{1})(\mathcal{D}_1) - (\delta - e_2\mathbb{1})(\delta - e_2\mathbb{1})(\mathcal{D}_2)\big),\\ c &= \frac{1}{\mathbb{1}(\mathcal{D}_1)}\big((\varepsilon - e_1\mathbb{1})(\delta - e_2\mathbb{1})(\mathcal{D}_1) - (\varepsilon - e_1\mathbb{1})(\delta - e_2\mathbb{1})(\mathcal{D}_2)\big). \end{aligned}$$

The transducer $\mathcal{T}$ is independent if and only if

$$(\varepsilon - e_1\mathbb{1})(\delta - e_2\mathbb{1})(\mathcal{D}_1) = (\varepsilon - e_1\mathbb{1})(\delta - e_2\mathbb{1})(\mathcal{D}_2) \tag{3.7}$$

We emphasize that, by Definition 3.3.10, only edges in the final component of the transducer are considered in Theorem 3.4. The non-final components do not influence the asymptotic main terms (see also Remark 3.3.7). The proof of this theorem can be found in Section 3.5.2.

In the following corollary, we consider the case of a normalized input and output, i.e., the constants of the expected values satisfy $e_1 = e_2 = 0$. This can be obtained by subtracting the original constants e_1 and e_2 from every input label and output label, respectively. Then the corollary follows directly from Theorem 3.4.

Corollary 3.3.11. *Suppose that $\mathbb{E}(\mathsf{Input}(X_n))$ and $\mathbb{E}(\mathsf{Output}(X_n))$ are both bounded. Then the transducer $\mathcal{T}$ is independent if and only if*

$$\varepsilon\delta(\mathcal{D}_2) = \varepsilon\delta(\mathcal{D}_1).$$

Example 3.3.12. We again consider the transducer of Example 3.3.8 in Figure 3.3. The set $\mathcal{D}_1$ consists of 3 functional digraphs and $\mathcal{D}_2$ consists of only one functional digraph (see Figure 3.4). By (3.7), we obtain the same equation as before, namely

$$a_1 - a_2 = 0,$$

as condition for the transducer to be independent.

Also by Theorem 3.4, the expected value of the output sum is

$$\frac{a_1 + a_2 + a_3 + a_4}{4} n + \mathcal{O}(1)$$

and the asymptotic variance is

$$\frac{5a_1^2 - 6a_1a_2 + 5a_2^2 - 2a_1a_3 - 2a_2a_3 + a_3^2 - 2a_1a_4 - 2a_2a_4 + 2a_3a_4 + a_4^2}{16}.$$

The covariance between the input sum and the output sum is

$$-\frac{a_1 - a_2}{4} n + \mathcal{O}(1).$$

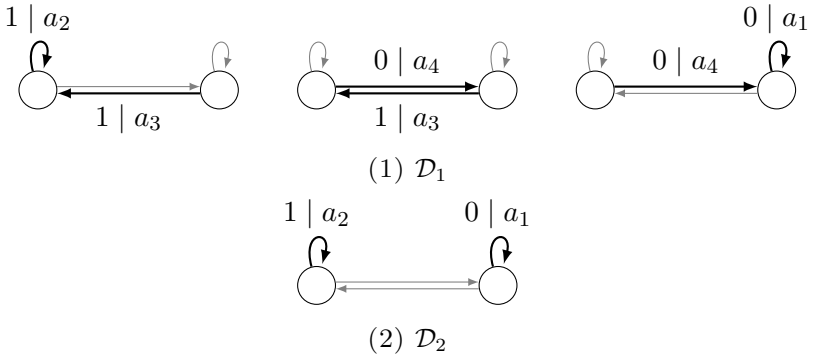


FIGURE 3.4. Functional digraphs of the transducer of Example 3.3.12.

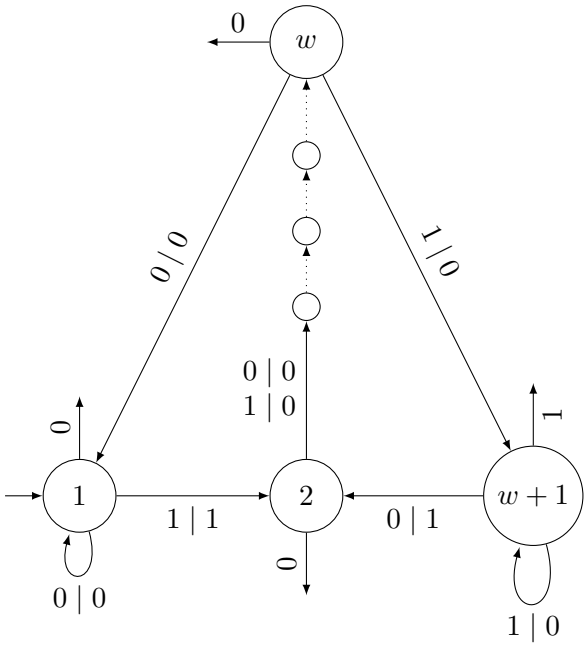


FIGURE 3.5. Transducer to compute the Hamming weight of the width-w non-adjacent form.

3.4. Examples of Transducers

In this section we give various examples to illustrate our theorems: these include both dependent and independent transducers and transducers with both bounded and unbounded variance of the output sum. These examples are also shown in the documentation of the method `FiniteStateMachine.asymptotic_moments()` [51] in SageMath. Especially, example 3.4.6 demonstrates how the combinatorial characterization of transducers with bounded variance can be used in cases where we only have limited information about the transducer.

Example 3.4.1 (Width-w non-adjacent form)**.** The width-w non-adjacent form (cf. [5, 78]) is a digit expansion with base 2, digits $\{0, \pm 1, \pm 3, \ldots, \pm(2^{w-1} - 1)\}$ and the syntactical rule

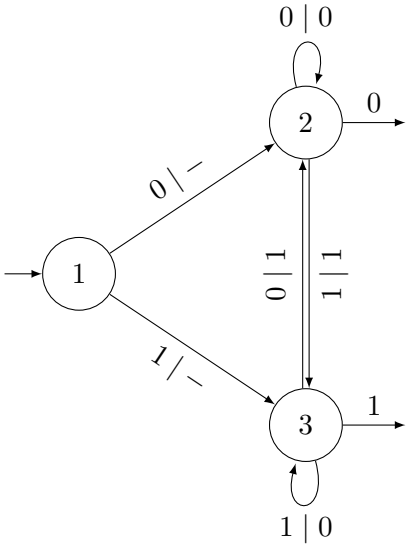


FIGURE 3.6. Transducer to compute the Gray code.

that at most one of any w consecutive digits is non-zero. The transducer in Figure 3.5 computes the Hamming weight of the width-w non-adjacent form when reading the standard binary expansion (cf. [50]). For $w = 2$, this transducer is the same as that in Figure 3.2. The variance of the output is not 0 (Corollary 3.3.4). With Theorem 3.3 or 3.4, we obtain that this transducer is independent for every w. Thus, the Hamming weight of the width-w non-adjacent form and the standard binary expansion are asymptotically independent.

Remark 3.4.2. Example 3.4.1 not only shows that there are infinitely many independent transducers, but also gives the construction of one such infinite family of independent transducers.

Example 3.4.3 (Gray code)**.** The Gray code is an encoding of the positive integers such that the Gray code of n and the Gray code of $n + 1$ differ only at one position. The transducer in Figure 3.6 computes the Gray code of an integer. The output label of the initial state is 0 and, as it does not influence the result, it is not given in the figure. The transducer is finally connected and finally aperiodic. The final component consisting of states 2 and 3 is independent (see Example 3.3.8). Thus, the Hamming weight of the Gray code and the standard binary expansion are asymptotically independent.

Example 3.4.4 (Length 2 blocks in the standard binary expansion)**.** We count the number of patterns of length 2 occurring in the standard binary expansion and compare it to the Hamming weight. By symmetry, it is obviously sufficient to consider the two patterns 01 and 11. The transducers in Figure 3.7 determine the number of 01- and 11-blocks, respectively. The variance of the output weight is not 0 in either case (Corollary 3.3.4), in fact the constant v_2 is $\frac{1}{16}$ (for 01-blocks) and $\frac{5}{16}$ respectively.

By Theorem 3.3 or 3.4, we also find that the transducer for 01-blocks is independent, while the transducer for 11-blocks (unsurprisingly) is not: the number of 11-blocks asymptotically depends on the number of 1's in the standard binary expansion, and the correlation coefficient is $\frac{2}{\sqrt{5}} \approx 0.894$.

(1) 01-blocks (2) 11-blocks

FIGURE 3.7. Transducers to compute the number of 01- and 11-blocks in the standard binary expansion.

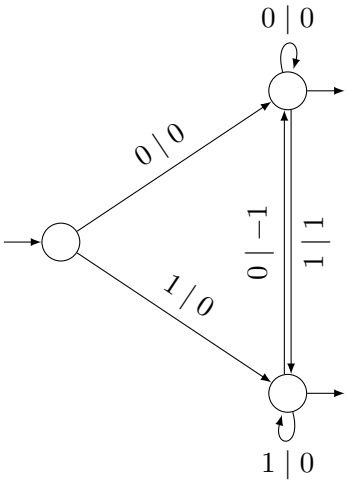


FIGURE 3.8. Transducer to compute the number of 10-blocks minus the number of 01-blocks in the standard binary expansion.

Example 3.4.5. Now, we give an example of a transducer with bounded variance of the output sum. We compute the number of 10-blocks minus the number of 01-blocks in the standard binary digit expansion. In Figure 3.8, we show the corresponding transducer. The output label of the initial state is 0 and, as it does not influence the result, it is not given in the figure. Any of the three cycles has output sum 0. Thus, the asymptotic variance of this random variable is 0. There is, of course, an intuitive explanation: when we read a 1 after a 0 (reading from right to left), the count increases by 1; when we read a 0 after a 1, the count decreases by 1; otherwise, it remains unchanged. Thus the final output value will only depend on the first and last digit.

Example 3.4.6. Finally, we consider the minimal Hamming weight of τ-adic digit expansions for a given algebraic integer τ and a given digit set $\mathcal{D}$. For $z \in \mathbb{Z}[\tau]$, a τ-adic expansion $(d_L \dots d_0)_\tau$ of z with digit set $\mathcal{D} \subset \mathbb{Z}[\tau]$ satisfies $d_i \in \mathcal{D}$ and

$$z = \sum_{i=0}^{L} d_i \tau^i.$$

This can be extended to d-dimensional joint expansions of vectors $z \in \mathbb{Z}[\tau]^d$ with digit set $\mathcal{D} \subset \mathbb{Z}[\tau]^d$.

In [42], a transducer to compute the minimal Hamming weight is constructed. Note that the output alphabet of the transducer need not be $\{0,1\}$ even if we are interested in the Hamming weight. The next theorem is an extension of Theorem 4 in [42].

Theorem 3.5. *Assume that $\mathcal{D} \subset \mathbb{Z}[\tau]^d$, for a positive integer d, and $\mathcal{D} \cap \tau\mathbb{Z}^d = \{0\}$. Let $\mathsf{mw}(z)$ be the minimal Hamming weight of a τ-adic joint digit representation of z with digits in $\mathcal{D}$. Assume further that the digit set $\mathcal{D}$ satisfies*

$$\forall c \in \mathbb{Z}[\tau]^d \quad \exists U \in \mathbb{R} \quad \forall z \in \mathbb{Z}[\tau]^d : |\mathsf{mw}(z+c) - \mathsf{mw}(z)| \le U.$$

Consider the random variable $R_n = \mathsf{mw}(D_n)$, where D_n is a random τ-adic joint digit representation of length n with digits in $\mathcal{A}_I \subset \mathbb{Z}[\tau]^d$. We assume that $(\tau, \mathcal{A}_I)$ is an irredundant digit system with $0 \in \mathcal{A}_I$. The digits of D_n are independent and identically distributed with uniform distribution on $\mathcal{A}_I$.

Then there exist constants E, V, with $V \neq 0$, such that

$$\mathbb{E}R_n = En + \mathcal{O}(1),$$
$$\mathbb{V}R_n = Vn + \mathcal{O}(1)$$

and

$$\frac{R_n - En}{\sqrt{Vn}}$$

is asymptotically normally distributed.

Proof. In [42], the authors give a strongly connected and aperiodic transducer computing $\mathsf{mw}(z)$ if the input is the τ-adic representation of z with digit set $\mathcal{A}_I$ read from left to right. Everything follows from Theorem 4 in [42] if $V \neq 0$.

To prove $V \neq 0$, we use Theorem 3.1, (b). In [42], the authors state that the transducer has a loop at the initial state 1 with input and output digit 0. Thus, in Theorem 3.1, (b), the value of k is 0.

On the other hand, there exists a $z \in \mathbb{Z}[\tau]^d$ with $\mathsf{mw}(z) \neq 0$. The input z leads to a state s. From each state the input 0^l, for some l, leads again to the initial state 1. Thus, the unique path whose input labels are given by the digit representation of $z\tau^l$ is a closed walk visiting 1 at least once. The output sum of this closed walk is $\mathsf{mw}(z\tau^l) = \mathsf{mw}(z) \neq 0$. Thus, there exists a closed walk whose output sum is not 0, which contradicts Theorem 3.1, (b) with $k = 0$. Therefore, we obtain $V \neq 0$. □

3.5. Proofs of the Theorems

In this section, we give the proofs of the theorems and corollaries of Section 3.3. We first prove the algebraic description and the combinatorial characterization in Sections 3.3.2 and 3.3.3. Later we prove the statements in Section 3.3.1 about the bounded variance.

3.5.1. Algebraic Description of Independent Transducers. First, we prove a slight extension of the 2-dimensional Quasi-Power Theorem [44] (a generalization of [66]). This extension will also take into account the case of a singular Hessian matrix.

We write boldface letters for a vector $\boldsymbol{s} = (s_1, s_2)^\top$. Furthermore, we use the notation $e^{\boldsymbol{s}} = (e^{s_1}, e^{s_2})$. We denote by $\mathbf{1}$ a 2- or 3-dimensional vector of ones, depending on the context. By $\|\cdot\|$, we denote the maximum norm $\|\boldsymbol{s}\| = \max(|s_1|, |s_2|)$.

Theorem 3.6. *Let $(\mathbf{\Omega}_n)_{n\geq 1}$ be a sequence of 2-dimensional real random vectors. Suppose that the moment generating function satisfies*

$$\mathbb{E}(e^{\langle \mathbf{\Omega}_n, \boldsymbol{s}\rangle}) = e^{u(\boldsymbol{s})\Phi(n)+v(\boldsymbol{s})}\big(1+\mathcal{O}(\kappa_n^{-1})\big),$$

the $\mathcal{O}$-term being uniform for $\|\boldsymbol{s}\| \leq \tau$, $\boldsymbol{s}\in\mathbb{C}^2$, $\tau>0$, where

(1) $u(\boldsymbol{s})$ and $v(\boldsymbol{s})$ are analytic for $\|s\|\leq\tau$ and independent of n;
(2) $\lim_{n\to\infty}\Phi(n)=\infty$;
(3) $\lim_{n\to\infty}\kappa_n=\infty$.

Then,

$$\begin{aligned}\mathbb{E}(\mathbf{\Omega}_n) &= \Phi(n)\operatorname{grad} u(\mathbf{0}) + \operatorname{grad} v(\mathbf{0}) + \mathcal{O}(\kappa_n^{-1}),\\ \mathbb{V}(\mathbf{\Omega}_n) &= \Phi(n)H_u(\mathbf{0}) + H_v(\mathbf{0}) + \mathcal{O}(\kappa_n^{-1}),\end{aligned} \tag{3.8}$$

where $H_u(\boldsymbol{s})$ is the Hessian matrix of u. Let Σ be the matrix $H_u(\mathbf{0})$.

If $H_u(\mathbf{0})$ is regular, then the standardized random vector

$$\mathbf{\Omega}_n^* = \frac{\mathbf{\Omega}_n - \Phi(n)\operatorname{grad} u(\mathbf{0})}{\sqrt{\Phi(n)}}$$

is asymptotically jointly normally distributed with variance-covariance matrix Σ.

If $H_u(\mathbf{0})$ has rank 1, then the limit distribution of $\mathbf{\Omega}_n^*$ is the direct product of a normal distribution and a degenerate distribution (if one of the variances is $\mathcal{O}(1)$) or a linear transformation thereof. In the first case, the coordinates of $\mathbf{\Omega}_n^*$ are asymptotically independent. In the second case, we have an asymptotically linear relationship between the two coordinates.

If $H_u(\mathbf{0})$ has rank 0, then the limit distribution of $\mathbf{\Omega}_n^*$ is degenerate.

Proof. The expressions (3.8) for expectation and variance-covariance matrix follow from the moment generating function by differentiation.

The case of a regular Hessian matrix $H_u(\mathbf{0})$ is exactly the statement of the 2-dimensional Quasi-Power Theorem [44].

For the case of a singular Hessian matrix, we follow the proof of the Quasi-Power Theorem [44]. We consider the characteristic function

$$f_n(\boldsymbol{s}) = \exp\Big(-\frac{1}{2}\boldsymbol{s}^\top H_u(\mathbf{0})\boldsymbol{s} + \mathcal{O}\Big(\frac{\|s\|^3+\|s\|}{\sqrt{\Phi(n)}}\Big)\Big)\big(1+\mathcal{O}(\kappa_n^{-1})\big)$$

of the standardized random vector $\mathbf{\Omega}_n^*$. Thus the characteristic function tends to

$$f(\boldsymbol{s}) = \exp\Big(-\frac{1}{2}\boldsymbol{s}^\top H_u(\mathbf{0})\boldsymbol{s}\Big).$$

If the Hessian matrix $H_u(\mathbf{0})$ has rank 0, then $f(\boldsymbol{s})$ equals the identity function. Thus, the distribution function is degenerate.

If the Hessian matrix $H_u(\mathbf{0})$ has rank 1 and the variance of the second coordinate $\Omega_{n,2}$ is $\mathcal{O}(1)$, then $H_u(\mathbf{0}) = \left(\begin{smallmatrix} v_1 & 0\\ 0 & 0\end{smallmatrix}\right)$ for a $v_1\in\mathbb{R}$. Thus,

$$f(\boldsymbol{s}) = \exp\Big(-\frac{1}{2}v_1^2 s_1^2\Big)\cdot 1$$

which is the characteristic function of the normal distribution with mean 0 and variance v_1 times the characteristic function of the point mass at 0.

If the Hessian matrix $H_u(\mathbf{0}) = \left(\begin{smallmatrix} v_1 & c\\ c & v_2\end{smallmatrix}\right)$ has rank 1 with $v_1v_2\neq 0$, then we consider the random variables $X=\Omega_{n,1}$, the first coordinate of $\mathbf{\Omega}_n$, and $Z=-\frac{c}{v_1}\Omega_{n,1}+\Omega_{n,2}$. Then, the

main term of the variance-covariance matrix of $(X, Z)^\top$ is $\left(\begin{smallmatrix} v_1 & 0 \\ 0 & 0 \end{smallmatrix}\right)\Phi(n)$. Thus, X is asymptotically normally distributed and Z is an asymptotically constant random variable (see previous case). □

Using this version of the Quasi-Power Theorem, we prove the algebraic description of independent transducers given in Theorem 3.3.

Proof of Theorem 3.3. Let K be the size of the input alphabet $\mathcal{A}_I$. Let a_{kln} be the number of sequences of length n with input sum k such that the corresponding output of the transducer $\mathcal{T}$ has sum l. We define

$$A(x, y, z) = \sum_{k\in\mathbb{R}} \sum_{l\in\mathbb{R}} \sum_{n=0}^{\infty} a_{kln} K^{-n} x^k y^l z^n.$$

Thus, the variable x marks the input sum, y marks the output sum, and z marks the length of the input. Then $[z^n]A(x, y, z)$ is the probability generating function of $\mathbf{\Omega}_n$, where $[z^n]b(z)$ is the coefficient of z^n in the power series $b(z)$.

Due to the block structure of $W_\varepsilon(y)$ in (3.2), we have

$$\begin{aligned} A(x, y, z) &= \boldsymbol{u}^\top \Big(I - \frac{z}{K} \sum_{\varepsilon\in\mathcal{A}_I} x^\varepsilon W_\varepsilon(y)\Big)^{-1} \boldsymbol{v} \\ &= \frac{F_1(x, y, z)}{\det\left(I - \frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^\varepsilon W_\varepsilon(y)\right)} \\ &= \frac{F_1(x, y, z)}{F_2(x, y, z)\det\left(I - \frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^\varepsilon M_\varepsilon(y)\right)}, \end{aligned} \tag{3.9}$$

with $\boldsymbol{u}^\top = (1, 0, \dots, 0)$ for the initial state, $v_s = y^{a(s)}$ for the final output label at state s and $F_1(x, y, z)$ and $F_2(x, y, z)$ "polynomials" in x, y and z. We use quotation marks because exponents of x and y might not be integers. However, only finitely many summands occur. The function $F_2(x, y, z)$ is the determinant corresponding to the non-final components in the upper left corner in (3.2).

The moment generating function of $\mathbf{\Omega}_n$ is

$$\mathbb{E}(e^{\langle \mathbf{\Omega}_n, \boldsymbol{s}\rangle}) = [z^n]A(e^{s_1}, e^{s_2}, z).$$

For extracting the coefficient, we investigate the dominant singularity of $A(x, y, z)$. Since the final component is strongly connected and aperiodic, we have a unique dominant simple eigenvalue of $\sum_{\varepsilon\in\mathcal{A}_I} x^\varepsilon M_\varepsilon(y)$ at $(x, y)^\top = \mathbf{1}$ by the theorem of Perron–Frobenius (cf. [33]). Because the final component is complete, this dominant eigenvalue is K, that is the size of the input alphabet $\mathcal{A}_I$. Thus, the unique dominant singularity of $f(x, y, z)^{-1} = \det\big(I - \frac{z}{|\mathcal{A}_I|}\sum_{\varepsilon\in\mathcal{A}_I} x^\varepsilon M_\varepsilon(y)\big)^{-1}$ at $(x, y)^\top = \mathbf{1}$ is a simple pole at $\rho(\mathbf{1}) = 1$. Therefore, we have $f_z(\mathbf{1}) \neq 0$.

For $(x, y)^\top$ in a small neighborhood of $\mathbf{1}$, there is a unique dominant singularity $\rho(x, y)$ of $f(x, y, z)^{-1}$ due to the continuity of eigenvalues.

Next, we consider the non-final components of the transducer. The corresponding transducer $\mathcal{T}_0$ is not complete. Let $\mathcal{T}_0^+$ be the complete transducer that is obtained from $\mathcal{T}_0$ by adding loops where necessary. The dominant eigenvalue of $\mathcal{T}_0^+$ is K. As the corresponding sums of transition matrices of $\mathcal{T}_0$ and $\mathcal{T}_0^+$ satisfy element-wise inequalities but are not equal (at $(x, y)^\top = \mathbf{1}$), the theorem of Perron–Frobenius (cf. [33, Theorem 8.8.1]) implies that the

dominant eigenvalues of $\mathcal{T}_0$ have absolute value less than K. Thus, the dominant singularities of $F_2(1,1,z)^{-1}$ are at $|z| > 1$. By continuity, this also holds for a small neighborhood of $(x,y)^\top = \mathbf{1}$.

As $A(1,1,z) = (1-z)^{-1}$, we obtain $F_1(\mathbf{1}) \neq 0$ and $F_1(x,y,\rho(x,y)) \neq 0$ for $(x,y)^\top$ in a small neighborhood of $\mathbf{1}$. Therefore, $\rho(x,y)$ is the simple dominant pole of $A(x,y,z)$ in a small neighborhood of $\mathbf{1}$.

The Laurent series of $A(x,y,z)$ at $z = \rho(x,y)$ is

$$A(x,y,z) = (z-\rho(x,y))^{-1}C(x,y) + \text{ power series in } (z-\rho(x,y))$$

for a function $C(x,y)$ which is analytic in a neighborhood of $\mathbf{1}$ with $C(\mathbf{1}) \neq 0$. Thus, by singularity analysis [30], we have

$$\mathbb{E}(e^{\langle \mathbf{\Omega}_n, \boldsymbol{s}\rangle}) = [z^n]A(e^{s_1}, e^{s_2}, z) = e^{u(\boldsymbol{s})n + v(\boldsymbol{s})}(1 + \mathcal{O}(\kappa^n))$$

with

$$\begin{aligned} u(\boldsymbol{s}) &= -\log \rho(e^{\boldsymbol{s}}), \\ v(\boldsymbol{s}) &= \log(-C(e^{\boldsymbol{s}})\rho(e^{\boldsymbol{s}})^{-1}) \end{aligned}$$

and $\kappa < 1$.

Theorem 3.6 yields the expected value, the variance-covariance matrix and the asymptotic normality of $\mathbf{\Omega}_n$. By implicit differentiation, we obtain the stated expressions. The error terms for the input sum are 0 because the input letters are independent and identically distributed. This also yields the explicit constants in (3.4).

Since the input alphabet $\mathcal{A}_I$ has at least two elements, the input sum has non-zero asymptotic variance. Thus, the asymptotic variance-covariance matrix Σ can have rank 1 or 2. Now, we consider these two cases separately and prove the asserted equivalence.

(1) Let Σ have rank 1. Then $\mathbf{\Omega}_n$ converges to a degenerate and a normally distributed random variable if the asymptotic variance of the output sum is 0; or a linear transformation thereof otherwise. Thus, $\mathbf{\Omega}_n$ is asymptotically independent if and only if the asymptotic variance of the sum of the output is 0. As the rank of Σ is 1, the asymptotic variance is 0 if and only if the asymptotic covariance is 0.
(2) Let Σ be invertible. By Theorem 3.6, we obtain an asymptotic joint normal distribution. Thus, $\mathbf{\Omega}_n$ is asymptotically independent if and only if its asymptotic covariance is 0.

□

3.5.2. Combinatorial Characterization of Independent Transducers. To obtain the combinatorial characterization, we use a version of the Matrix-Tree Theorem as proved by Chaiken [16] and Moon [75]. This version does not use trees, but *forests*, i.e., digraphs whose weak components are trees.

Definition 3.5.1. Let A, $B \subseteq \{1,\dots,N\}$. Let $\mathcal{F}_{A,B}$ be the set of all forests which are spanning subgraphs of the final component of the transducer $\mathcal{T}$ with $|A|$ trees such that every tree is rooted at some vertex $a \in A$ and contains exactly one vertex $b \in B$.

Let $A = \{i_1,\dots,i_n\}$ and $B = \{j_1,\dots,j_n\}$ with $i_1 < \cdots < i_n$ and $j_1 < \cdots < j_n$. For $F \in \mathcal{F}_{A,B}$, we define a function $g\colon B \to A$ by $g(j) = i$ if j is in the tree of F which is rooted in vertex i. We further define the function $h\colon A \to B$ by $h(i_k) = j_k$ for $k = 1,\dots,n$. The composition $g \circ h\colon A \to A$ is a permutation of A. We define $\operatorname{sgn} F = \operatorname{sgn} g \circ h$.

If $|A| \neq |B|$, then $\mathcal{F}_{A,B} = \emptyset$. If $|A| = |B| = 1$, then $\operatorname{sgn} F = 1$ and $\mathcal{F}_{A,B}$ consists of all spanning trees rooted in $a \in A$.

Theorem (All-Minors-Matrix-Tree Theorem [16, 75])**.** *For a directed graph with loops, let $L = (l_{ij})_{1\leq i,j\leq N}$ be the Laplacian matrix, that is $\sum_{j=1}^{N} l_{ij} = 0$ for every $i = 1, \dots, N$ and $-l_{ij}$ is the number of edges from i to j for $i \neq j$. Then, for $|A| = |B|$, the minor $\det L_{A,B}$ satisfies*

$$\det L_{A,B} = (-1)^{\sum_{i\in A} i + \sum_{j\in B} j} \sum_{F\in\mathcal{F}_{A,B}} \operatorname{sgn} F$$

where $L_{A,B}$ is the matrix L whose rows with index in A and columns with index in B are deleted.

The All-Minors-Matrix-Tree Theorem is still valid for $|A| \neq |B|$ if we assume that the determinant of a non-square matrix is 0. For notational simplicity, we use this convention in the rest of this section.

The next lemma connects the derivatives of $f(x, y, z)$ with weighted sums of functional digraphs. Theorem 3.4 follows immediately from this lemma and Theorem 3.3.

Lemma 3.5.2. *Let K be the size of the input alphabet $\mathcal{A}_I$. For $f(x, y, z) = \det\big(I - \frac{z}{|\mathcal{A}_I|}\sum_{\varepsilon\in\mathcal{A}_I} x^{\varepsilon} M_{\varepsilon}(y)\big)$, we have*

$$\begin{aligned}
f_x(1,1,1) &= -K^{-N}\varepsilon(\mathcal{D}_1), & f_{xy}(1,1,1) &= K^{-N}(\varepsilon\delta(\mathcal{D}_2) - \varepsilon\delta(\mathcal{D}_1)),\\
f_y(1,1,1) &= -K^{-N}\delta(\mathcal{D}_1), & f_{xz}(1,1,1) &= K^{-N}(\varepsilon\mathbb{1}(\mathcal{D}_2) - \varepsilon\mathbb{1}(\mathcal{D}_1)),\\
f_z(1,1,1) &= -K^{-N}\mathbb{1}(\mathcal{D}_1), & f_{yz}(1,1,1) &= K^{-N}(\delta\mathbb{1}(\mathcal{D}_2) - \delta\mathbb{1}(\mathcal{D}_1)),\\
f_{xx}(1,1,1) + f_x(1,1,1) &= K^{-N}(\varepsilon\varepsilon(\mathcal{D}_2) - \varepsilon\varepsilon(\mathcal{D}_1)),\\
f_{yy}(1,1,1) + f_y(1,1,1) &= K^{-N}(\delta\delta(\mathcal{D}_2) - \delta\delta(\mathcal{D}_1)),\\
f_{zz}(1,1,1) + f_z(1,1,1) &= K^{-N}(\mathbb{1}\mathbb{1}(\mathcal{D}_2) - \mathbb{1}\mathbb{1}(\mathcal{D}_1)).
\end{aligned}$$

Proof. The idea of the proof is as follows: First, we compute the derivatives and write them as sums over all states. Using the All-Minors-Matrix-Tree Theorem, we change the summation to a sum over forests. In the next step, we again change to a sum over functional digraphs.

Let u_1, u_2 be any of the variables x, y or z. For a matrix $M = (m_{ij})_{1\leq i,j\leq N}$, we define the matrix $M_{k:u_1} = (\hat{m}_{ij})_{1\leq i,j\leq N}$ with $\hat{m}_{ij} = m_{ij}$ for $i \neq k$ and $\hat{m}_{kj} = \frac{\partial}{\partial u_1} m_{kj}$. Thus $M_{k:u_1}$ is the matrix M where row k is differentiated with respect to u_1.

We further define the derivatives at $\mathbf{1}$ as

$$D_{u_1}(\,\cdot\,) = \frac{\partial}{\partial u_1}(\,\cdot\,)\Big|_{\mathbf{1}}$$

and

$$D_{u_1u_2}(\,\cdot\,) = \frac{\partial^2}{\partial u_1 \partial u_2}(\,\cdot\,)\Big|_{\mathbf{1}}.$$

Applying the product rule to the definition of the determinants gives us

$$D_{u_1}(f) = \sum_{j=1}^{N} \det\Big(I - \frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^{\varepsilon} M_{\varepsilon}(y)\Big)_{j:u_1}\Big|_{\mathbf{1}},$$

$$D_{u_1u_2}(f) = \sum_{i=1}^{N}\sum_{j=1}^{N} \det\Big(I - \frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^{\varepsilon} M_\varepsilon(y)\Big)_{i:u_1,\,j:u_2}\Big|_{\mathbf{1}}.$$

In these equations, we have a sum over all states.

Since our original matrix $I-\frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^{\varepsilon}M_\varepsilon(y)$ is sparse, and $(I-\frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^{\varepsilon}M_\varepsilon(y))_{j:u_1}$ is even sparser, we use Laplace expansion along row j to determine these determinants. If $i\neq j$, we use Laplace expansion along rows i and j to determine $\det(I-\frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} x^{\varepsilon}M_\varepsilon(y))_{i:u_1,\,j:u_2}$ for the second derivatives. If $i=j$, we only expand along row j. Depending on the variable of differentiation, there are at most K non-zero values in row j after differentiation.

For a transition e, we denote by $t(e)$, $h(e)$, $\varepsilon(e)$ and $\delta(e)$ the tail, the head, the input and the output of the transition e, respectively. Furthermore, let $w_e = \frac{1}{K}x^{\varepsilon(e)}y^{\delta(e)}z$ be the weight of the transition e.

If we use Laplace expansion along two different rows, we must be careful with the sign. Therefore, we define

$$\sigma_{de} = (-1)^{[t(e)>t(d)]+[h(e)>h(d)]}$$

for two transitions d and e. Here, we use Iverson's notation, that is [*expression*] is 1 if *expression* is true and 0 otherwise (cf. [41]).

Let L be the Laplacian matrix of the underlying graph, that is

$$L = KI - \sum_{\varepsilon\in\mathcal{A}_I} M_\varepsilon(1).$$

Recall the notation $L_{A,B}$ for the matrix where the rows corresponding to A and the columns corresponding to B have been removed. Laplace expansion yields

$$D_{u_1}(f) = -K^{-N+1}\sum_{j=1}^{N}\sum_{\substack{e\in\mathcal{E}\\ t(e)=j}} (-1)^{t(e)+h(e)} D_{u_1}(w_e)\det(L_{\{t(e)\},\{h(e)\}}),$$

$$\begin{aligned} D_{u_1u_2}(f) = {} & -K^{-N+1}\sum_{j=1}^{N}\sum_{\substack{e\in\mathcal{E}\\ t(e)=j}} (-1)^{t(e)+h(e)} D_{u_1u_2}(w_e)\det(L_{\{t(e)\},\{h(e)\}}) \\ & + K^{-N+2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{d\in\mathcal{E}\\ t(d)=i}}\sum_{\substack{e\in\mathcal{E}\\ t(e)=j}} \big((-1)^{t(d)+h(d)+t(e)+h(e)}\sigma_{de} \\ & \qquad \times D_{u_1}(w_d)D_{u_2}(w_e)\det(L_{\{t(d),t(e)\},\{h(d),h(e)\}})\big). \end{aligned}$$

Next, we use the All-Minors-Matrix-Tree Theorem and change the summation over all rows to a summation over forests. We obtain

$$D_{u_1}(f) = -K^{-N+1}\sum_{e\in\mathcal{E}} D_{u_1}(w_e) \sum_{F\in\mathcal{F}_{\{t(e)\},\{h(e)\}}} \operatorname{sgn} F,$$

$$\begin{aligned} D_{u_1u_2}(f) = {} & -K^{-N+1}\sum_{e\in\mathcal{E}} D_{u_1u_2}(w_e) \sum_{F\in\mathcal{F}_{\{t(e)\},\{h(e)\}}} \operatorname{sgn} F \\ & + K^{-N+2}\sum_{d\in\mathcal{E}}\sum_{\substack{e\in\mathcal{E}\\ e\neq d}} \Big(\sigma_{de}D_{u_1}(w_d)D_{u_2}(w_e) \sum_{F\in\mathcal{F}_{\{t(d),t(e)\},\{h(d),h(e)\}}} \operatorname{sgn} F\Big). \end{aligned}$$

Let $F \in \mathcal{F}_{\{t(e)\},\{h(e)\}}$ be a forest for a transition $e \in \mathcal{E}$. Then $F+e$ is a spanning functional digraph with one component. Let $F \in \mathcal{F}_{\{t(d),t(e)\},\{h(d),h(e)\}}$ be a forest for transitions $d, e \in \mathcal{E}$. Then $F + d + e$ is a spanning functional digraph with one or two components, depending on $\sigma_{de} \operatorname{sgn} F$. If $\sigma_{de} \operatorname{sgn} F = 1$, then it has two components. Otherwise, it has one component. Now we can change the summation into a sum over functional digraphs and obtain

$$\begin{aligned}
D_{u_1}(f) &= -K^{-N+1} \sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} \sum_{e\in C} D_{u_1}(w_e),\\
D_{u_1u_2}(f) &= -K^{-N+1} \sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} \sum_{e\in C} D_{u_1u_2}(w_e)\\
&\quad + K^{-N+2} \sum_{D\in\mathcal{D}_2} \sum_{C_1\in\mathcal{C}_D} \sum_{\substack{C_2\in\mathcal{C}_D\\ C_2\neq C_1}} \sum_{d\in C_1} \sum_{e\in C_2} D_{u_1}(w_d)D_{u_2}(w_e)\\
&\quad - K^{-N+2} \sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} \sum_{d\in C} \sum_{\substack{e\in C\\ e\neq d}} D_{u_1}(w_d)D_{u_2}(w_e).
\end{aligned}$$

For a transition e, we know the first derivatives

$$D_x(w_e) = \frac{1}{K}\varepsilon(e), \qquad D_y(w_e) = \frac{1}{K}\delta(e), \qquad D_z(w_e) = \frac{1}{K}\mathbb{1}(e),$$

and the second derivatives

$$\begin{aligned}
D_{xy}(w_e) &= \frac{1}{K}\varepsilon(e)\delta(e), & D_{xx}(w_e) &= \frac{1}{K}\varepsilon(e)(\varepsilon(e)-1),\\
D_{xz}(w_e) &= \frac{1}{K}\varepsilon(e)\mathbb{1}(e), & D_{yy}(w_e) &= \frac{1}{K}\delta(e)(\delta(e)-1),\\
D_{yz}(w_e) &= \frac{1}{K}\delta(e)\mathbb{1}(e), & D_{zz}(w_e) &= 0.
\end{aligned}$$

Thus, we obtain the formulas stated in the lemma. □

3.5.3. Bounded Variance and Singular Asymptotic Variance-Covariance Matrix. We next give the proof of the equivalence of the three statements in Theorem 3.1, including the bounded variance.

Proof of Theorem 3.1. We first prove (a) ⇔ (b) by giving an alternative representation of the generating function $A(x, y, z)$ from the proof of Theorem 3.3. Then we prove the equivalence (b) ⇔ (c).

(a) ⇔ (b): WLOG, we assume that the expected value $\mathbb{E}(\mathsf{Output}(X_n))$ is a $\mathcal{O}(1)$. Otherwise, we have $\mathbb{E}(\mathsf{Output}(X_n)) = e_2 n + \mathcal{O}(1)$ for some constant e_2 (see Theorem 3.3). Then we subtract e_2 from the output of every transition, as for Corollary 3.3.11. Under this assumption, Theorem 3.4 implies that (b) can only hold with $k = 0$.

As the input sum is inconsequential, we consider $A(1, y, z)$. For brevity, we write $A(y, z)$ instead. We obtain

$$A(y,z) = \boldsymbol{u}^\top \Big(I - \frac{z}{K} \sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(y)\Big)^{-1} \boldsymbol{v}$$

where W_ε for $\varepsilon \in \{0, \dots, q-1\}$ are the transition matrices of $\mathcal{T}$ and $K = |\mathcal{A}_I|$.

Since $\mathcal{T}$ is complete, finally connected and finally aperiodic, $A(1,z)$ has a simple dominant pole at $z=1$ (see the proof of Theorem 3.3). We know that

$$\begin{aligned}\mathbb{E}(\mathsf{Output}(X_n)) &= [z^n]A_y(1,z) = \mathcal{O}(1),\\ \mathbb{V}(\mathsf{Output}(X_n)) &= [z^n]A_{yy}(1,z) + \mathcal{O}(1).\end{aligned} \tag{3.10}$$

Let s be any state of the final component. Each path starting at state 1 either does or does not visit state s. In the first case, this path can be decomposed into a path leading to state s and visiting s only once, followed by a sequence of closed walks visiting state s exactly once, and a path starting in s and not returning to s. We translate this decomposition into an equation for the corresponding generating functions.

Let $\mathcal{P}^s$ be the set of all walks in $\mathcal{T}$ which start at state s but never return to state s. All other states can be visited arbitrarily often. We define the corresponding generating function $P^s(y,z) = \sum_{P\in\mathcal{P}^s} y^{\delta(P)} z^{\mathbb{1}(P)} K^{-\mathbb{1}(P)}$. Then $[z^n]P^s(y,z)$ is the probability generating function of the output sum over walks in $\mathcal{P}^s$ of length n.

Let $\mathcal{P}^{1s}$ be the set of all walks in $\mathcal{T}$ which start at state 1 and lead to state s, visiting s exactly once. If $s=1$, this set consists only of the path of length 0. The corresponding generating function is called $P^{1s}(y,z)$.

Let $\mathcal{P}^1$ be the set of all walks in $\mathcal{T}$ which start at state 1 and never visit state s. If $s=1$, this set is empty. The corresponding generating function is called $P^1(y,z)$.

Let $\mathcal{C}^s$ be the set of all closed walks in $\mathcal{T}$ which visit state s exactly once. All other states can be visited arbitrarily often. The corresponding generating function is called $C^s(y,z)$.

Thus, we have

$$A(y,z) = P^1(y,z) + \frac{P^{1s}(y,z)P^s(y,z)}{1-C^s(y,z)}. \tag{3.11}$$

Let α be any of the superscripts 1, $1s$ or s. By deleting the transitions leading to s, we have

$$P^\alpha(y,z) = (\boldsymbol{u}^\alpha)^\top \Big(I - \frac{z}{K}\sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(y)E\Big)^{-1}\boldsymbol{v}^\alpha,$$

where $E = \operatorname{diag}(1,\dots,1,0,1,\dots,1)$ and $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ are fixed vectors. The position of the zero on the diagonal of E corresponds to the state s. The vectors $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ depend on α and may include the output of the transitions leading to s, but E is independent of α. Since we have the element-wise inequalities

$$0 \le \sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(1)E \le \sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(1)$$

and $\sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(1)E \ne \sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(1)$, we know that the spectral radii satisfy

$$\rho\Big(\sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(1)E\Big) < \rho\Big(\sum_{\varepsilon\in\mathcal{A}_I} W_\varepsilon(1)\Big) = K$$

due to the theorem of Perron–Frobenius (cf. [33, Theorem 8.8.1]). Here, it is important that s lies in the final component. Thus, the dominant singularities of $P^\alpha(1,z)$ are at $|z|>1$. Furthermore, we know that $P^s(1,1)>0$ and $P^{1s}(1,1)>0$ by the definition as generating functions.

Because $z = 1$ is a simple pole of $A(1, z)$, no pole of $P^1(1, z)$ and $P^{1s}(1, z)P^s(1, z)$, and $P^{1s}(1, 1)P^s(1, 1) \neq 0$, it is a simple root of $1 - C^s(1, z)$ by (3.11). Thus, we can write $1 - C^s(1, z) = (z - 1)g(z)$ for a suitable function $g(z)$ with $g(1) \neq 0$.

By (3.10), (3.11) and singularity analysis [30], we obtain

$$\mathcal{O}(1) = \mathbb{E}(\mathsf{Output}(X_n)) = P^{1s}(1, 1)P^s(1, 1)C^s_y(1, 1)g(1)^{-2}n + \mathcal{O}(1).$$

Therefore, $C^s_y(1, 1) = 0$.

Similarly, we have

$$\mathbb{V}(\mathsf{Output}(X_n)) = P^{1s}(1, 1)P^s(1, 1)C^s_{yy}(1, 1)g(1)^{-2}n + \mathcal{O}(1), \tag{3.12}$$

taking into account that $C^s_y(1, 1) = 0$.

By (3.12), $\mathbb{V}(\mathsf{Output}(X_n)) = \mathcal{O}(1)$ is equivalent to $C^s_{yy}(1, 1) = 0$, and thus, $C^s_{yy}(1, 1) + C^s_y(1, 1) = 0$ as $C^s_y(1, 1) = 0$. By the definition of $C^s(y, z)$, this is equivalent to

$$\sum_{C \in \mathcal{C}^s} \delta(C)^2 K^{-\mathbb{1}(C)} = 0,$$

and thus $\delta(C) = 0$ for all $C \in \mathcal{C}^s$.

(b) ⇒ (c): Let $\mathcal{C}^s$ be the set of all closed walks in the final component of $\mathcal{T}$ which visit state s exactly once. If D is any cycle of the final component of the transducer, then one of the following occurs.

- No visits of state s: Let i be a vertex of D. Because the final component is strongly connected, there exists a closed walk $C \in \mathcal{C}^s$ with $s, i \in C$. Let D' be the combined closed walk of D and C. Then, $D' \in \mathcal{C}^s$, and so we have
$$\delta(D) = \delta(D') - \delta(C) = k\mathbb{1}(D') - k\mathbb{1}(C) = k\mathbb{1}(D).$$
- One visit of state s: Then we have $D \in \mathcal{C}^s$ and $\delta(D) = k\mathbb{1}(D)$.

(c) ⇒ (b): As a closed walk visiting s exactly once can be decomposed into cycles, this is obvious.

□

Next, we prove the equivalence for the quasi-deterministic output sum.

Proof of Theorem 3.2.

(d) ⇒ (e): Let C be an arbitrary cycle of the transducer and P be a path from the initial state 1 to any state of the cycle. Let z_n be the input sequence along the combined walk consisting of P and n times C. Then, by quasi-determinism and the definition of the output, we have

$$k(\mathbb{1}(P) + n\mathbb{1}(C)) + \mathcal{O}(1) = \mathsf{Output}(z_n) = \delta(P) + n\delta(C) + \mathcal{O}(1).$$

Thus, $n(\delta(C) - k\mathbb{1}(C))$ is bounded by a constant depending on P and C, but independent of n. Therefore, we know that $\delta(C) = k\mathbb{1}(C)$.

(e) ⇒ (d): WLOG, we assume $k = 0$ (replace $\delta(e)$ by $\delta(e) - k$ for all transitions e). For every $z \in \mathcal{A}^*_I$, we have $|\mathsf{Output}(z)| \leq \sum_{e \in \mathcal{E}} |\delta(e)| + \max_{s \in \{1,\dots,S\}} |a(s)|$ because all cycles have output sum 0 so that every transition contributes at most once to $\mathsf{Output}(z)$. Therefore, we have a quasi-deterministic random variable $\mathsf{Output}(X_n) = \mathcal{O}(1)$.

□

Now, we consider transducers whose output alphabet is $\{0, 1\}$ and prove that there are only trivial cases with a bounded variance.

PROOF OF COROLLARY 3.3.4. We know that the output digits $(0,\dots,0)$ and $(1,\dots,1)$ have asymptotic variance 0.

Assume that the asymptotic variance is 0. Let k be the constant given in Theorem 3.1. Then, we know $k \in [0,1]$. By the aperiodicity of the final component, there exist cycles $C_1,\dots,C_n$ of coprime length and therefore integers $b_1,\dots,b_n$ with

$$1 = b_1\mathbb{1}(C_1) + \cdots + b_n\mathbb{1}(C_n).$$

Thus,

$$k = b_1\delta(C_1) + \cdots + b_n\delta(C_n) \in \mathbb{Z}$$

and hence, $k \in \{0,1\}$. Therefore, $(0,\dots,0)$ and $(1,\dots,1)$ are the only output digits with asymptotic variance 0. □

This last proof shows the equivalence of the statements in Corollary 3.3.5, including a transducer with a singular asymptotic variance-covariance matrix.

PROOF OF COROLLARY 3.3.5. WLOG, we can assume that both expected values

$$\mathbb{E}(\mathsf{Output}(X_n)) = \mathbb{E}(\mathsf{Input}(X_n)) = \mathcal{O}(1).$$

We know that the asymptotic variance v_1 of the input is non-zero because $\mathcal{A}_I$ consists of at least two elements. As in the last paragraph of the proof of Theorem 3.6, we consider the random variables $Y_n = \mathsf{Input}(X_n)$ and $Z_n = -\frac{c}{v_1}\mathsf{Input}(X_n) + \mathsf{Output}(X_n)$ and their variance-covariance matrix $\left(\begin{smallmatrix} v_1 & 0 \\ 0 & v_2 - \frac{c^2}{v_1} \end{smallmatrix}\right)$. The matrix Σ is singular if and only if the asymptotic variance of Z_n is 0.

Thus, we consider a transducer with the same input as the original transducer $\mathcal{T}$ for which the output of a transition e is $-\frac{c}{v_1}\varepsilon(e) + \delta(e)$. By Theorem 3.1, the output sum of this new transducer has asymptotic variance 0 if and only if there exists an $m \in \mathbb{R}$ such that

$$-\frac{c}{v_1}\varepsilon(C) + \delta(C) = m\mathbb{1}(C)$$

for every cycle C of the final component. Since the expected value of Z_n is $\mathcal{O}(1)$, we have $m = 0$.

The second statement follows from Theorem 3.1. □

CHAPTER 4

Variance and Covariance of Several Simultaneous Outputs of a Markov Chain

In this chapter, we give combinatorial characterizations of transducers with several simultaneously considered output sums which have a singular variance-covariance matrix and of output sums of transducers which are asymptotically independent.

This generalizes the results of the previous chapter to more than one output sum and input sequences which are not necessarily independently identically distributed. This generalization can be described by using Markov chains. While writing this thesis, this generalization was considered after [56] has appeared, thus the results are presented separately in two different chapters.

4.1. Introduction

In this chapter, every transition of a transducer has not only one output, but several outputs $k_1, \ldots, k_m$. We investigate the random variables $K_n^{(1)}, \ldots, K_n^{(m)}$ which are the sums of the outputs $k_1, \ldots, k_m$, respectively, along a random path of length n in the transducer.

The m different output sums of the transducer turn out to be asymptotically jointly normally distributed in the case of a non-singular variance-covariance matrix by the Quasi-Power Theorem [24, Theorem 2.22]. This will be proved to be equivalent to the linear independence of certain functions of cycles of the underlying graph of the transducer. Furthermore, we give a combinatorial characterization of transducers with two output sums which are asymptotically independent.

In contrast to Chapter 3, we allow the input sequence of the transducer to be generated by a Markov source. This is equivalent to choosing transition probabilities for the transitions of the transducer. As the input of the transducer is then no longer needed, we define our setting in terms of a Markov chain with m output functions mapping the transitions of the Markov chain to real numbers. This allows us to model an input sequence for a transducer whose letters do not occur with equal probabilities and/or have dependencies between the letters.

As an example, we prove that the Hamming weight of the width-w non-adjacent form is asymptotically jointly normally distributed for two different values of $w \geq 2$.

The outline of this chapter is as follows: In Section 4.2, we define our setting and the types of graphs we use to state the combinatorial characterization of independent output sums and singular variance-covariance matrices. These characterizations are given in Section 4.3 and examples are given in Section 4.4. In Section 4.5, we finally prove the results of Section 4.3.

4.2. Preliminaries

In this chapter, a *finite Markov chain* consists of a finite state space $\{1, \ldots, M\}$, a finite set of transitions $\mathcal{E}$ between the states, each with a positive transition probability, and a

unique[1] initial state 1. We denote the transition probability for a transition e by p_e. Then we have

$$\sum_{\substack{e\in\mathcal{E}\\ e \text{ starts in } i}} p_e = 1$$

for all states i. Note that for all transitions $e \in \mathcal{E}$, we require $p_e > 0$. Further note that there may be multiple transitions between two states but always only a finite number of them. This may be useful for different outputs later on.

The transition probabilities induce a probability distribution on the paths of length n starting in the initial state 1. Let X_n be a random path of length n according to this model.

All states of the underlying digraph of the Markov chain are assumed to be accessible from the initial state. Contracting each strongly connected component of the underlying digraph gives an acyclic digraph, the so-called condensation. We assume that this condensation has only one leaf (i.e., one vertex with out-degree 0). The strongly connected component corresponding to this leaf is called *final component*. We assume that the period (i.e., the greatest common divisor of the lengths of all cycles) of this final component is 1. We call such Markov chains *finally connected* and *finally aperiodic*.

Additionally we use several *output functions* $k\colon \mathcal{E} \to \mathbb{R}$. The corresponding random variable K_n is the sum of all values of k along a random path X_n. We call K_n the *output sum* of the Markov chain with respect to k. We use several output function $k_1, \dots, k_m$ and the corresponding random variables $K_n^{(1)}, \dots, K_n^{(m)}$ simultaneously for one Markov chain.

Remark 4.2.1. Usually, one is interested in a function evaluated at the sequence of random states of the Markov chain. This is equivalent to this setting with an output function of the transitions: For the one direction, the restriction of the output function to the outgoing transitions of one state is constant for every state. For the other direction, we use the standard construction of the Markov chain with state space $\{(i, j) \mid 1 \le i, j \le M\}$.

Thus, our setting can be seen as a Markov source with a finite set of m-dimensional vectors as alphabet.

In this chapter, we are interested in the joint distribution of the random variables $K_n^{(1)}$, $\dots$, $K_n^{(m)}$. For one coordinate, we will prove that the expected value of $K_n^{(i)}$ is $e_i n + \mathcal{O}(1)$ for constants e_i. The variance-covariance matrix of $K_n^{(1)}, \dots, K_n^{(m)}$ will turn out to be $\Sigma n + \mathcal{O}(1)$ for a matrix Σ. We call Σ the asymptotic variance-covariance matrix and its entries the asymptotic variances and covariances.

We will combinatorically characterize Markov chains with output functions such that the variance-covariance matrix is regular. Furthermore, we give a combinatorial characterization of the case that the asymptotic covariance is zero. As this is only influenced by two output functions, we restrict ourselves to $K_n^{(1)}$ and $K_n^{(2)}$ in this case.

Remark 4.2.2. A Markov chain with one output function can be obtained by a transducer with additional probability distributions for the outgoing transitions of each state and by deleting the input labels of the transducer.

If we have two transducers where only the outputs of the transitions are different, we can choose probability distributions for the outgoing transitions of each state. Then we obtain a

[1]This is no restriction as we can always add an additional state and the transitions starting in this state with probabilities corresponding to the non-degenerate initial distribution. The output functions are then extended by mapping these transitions to 0.

Markov chain with two output functions. Thus, we can use the results of this chapter for two output functions (see Examples 4.4.2 and 4.4.3).

Remark 4.2.3. We can additionally have *final output functions* $f\colon \{1,\dots,M\} \to \mathbb{R}$ for each output function k and redefine the random variable K_n as the sum of the values of the output function k along a random path X_n plus the final output f of the final state of this path. We will see that this does not change the main terms of the asymptotic behavior. Thus, the results in Section 4.3 are still valid (see also Remark 4.5.5).

Remark 4.2.4. The Parry measure are probabilities p_e such that every path of length n has the same weight up to a constant factor (cf. [83, 93]). If we are interested in probabilities such that every path of length n starting in the initial state 1 has exactly the same weight, we have to use the Parry measure with additional *exit weights*: Each path is additionally weighted by these exit weights according to the final state of the path (cf. [53, Lemma 4.1] and Lemma 5.4.1).

However, the sum of the weights of all paths of length n is no longer normalized: It differs from 1 by an exponentially small error term for $n \to \infty$. This gives an approximate equidistribution of all paths of length n. As we are interested in the asymptotic behavior for $n \to \infty$, the expected value and the variance of the corresponding measurable function K_n can still be defined as usual.

If we use these exit weights w_s in our setting, the main terms of the asymptotic behavior are not changed. Thus, the theorems in Section 4.3 are still valid (see also Remark 4.5.5).

These exit weights can also be used to simulate final and non-final states of a transducer by setting the weights of non-final states to 0. However, not all exit weights of the final component are allowed to be zero.

Next, we define some subgraphs of the underlying graph of the final component and extend the probabilities and the output functions to these subgraphs.

Definition 4.2.5. We define the following types of directed graphs as subgraphs of the final component of the Markov chain.

- A *rooted tree* is a weakly connected digraph with one vertex which has out-degree 0, while all other vertices have out-degree 1. The vertex with out-degree 0 is called the *root* of the tree.
- A *functional digraph* is a digraph whose vertices have out-degree 1. Each component of a functional digraph consists of a directed cycle and some trees rooted at vertices of the cycle. For a functional digraph D, let $\mathcal{C}_D$ be the set of all cycles of D.

The probabilities p_e can be multiplicatively extended to a weight function for arbitrary subgraphs of the Markov chain: Let D be any subgraph of the underlying graph of the Markov chain, then define the weight of D by

$$p_D = \prod_{e \in D} p_e.$$

For a path P of length n, this is exactly the probability $\mathbb{P}(X_n = P)$.

However, the output function k is additively extended to cycles C of the underlying graph of the Markov chain by

$$k(C) = \sum_{e \in C} k(e).$$

This can further be extended to functional digraphs:

Definition 4.2.6. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be the sets of all spanning subgraphs of the final component of the Markov chain $\mathcal{M}$ which are functional digraphs and have one and two components, respectively.

For functions g and $h\colon \mathcal{E} \to \mathbb{R}$, we define

$$\begin{aligned} g(\mathcal{D}_1) &= \sum_{D\in\mathcal{D}_1} p_D \sum_{C\in\mathcal{C}_D} g(C), \\ (g,h)(\mathcal{D}_1) &= \sum_{D\in\mathcal{D}_1} p_D \sum_{C\in\mathcal{C}_D} g(C)h(C), \\ (g,h)(\mathcal{D}_2) &= \sum_{D\in\mathcal{D}_2} p_D \sum_{C_1\in\mathcal{C}_D} \sum_{\substack{C_2\in\mathcal{C}_D \\ C_2\neq C_1}} g(C_1)h(C_2). \end{aligned}$$

As functions g and h, we use the output functions $k_1, \ldots, k_m$ and the constant function $\mathbb{1}(e) = 1$.

4.3. Main Results

In this section, we present the combinatorial characterization of output functions of Markov chains which are asymptotically independent and of Markov chains with output functions with a singular variance-covariance matrix.

If the underlying directed graph of the Markov chain is j-regular, every transition has probability $1/j$ and the output function $k_1\colon \mathcal{E} \to \{0, 1, \ldots, j-1\}$ is such that the restrictions of k_1 to the outgoing transitions of one state is bijective for every state, then these results are stated in [56] and Chapter 3 (see also Remark 4.2.2).

The next definition describes a sequence of random variables whose difference from its expected value is bounded for all elements.

Definition 4.3.1. The output sum K_n of a Markov chain is called *quasi-deterministic* if there is a constant $a \in \mathbb{R}$ such that

$$K_n = an + \mathcal{O}(1)$$

holds for all n.

Next we give the combinatorial characterization of output sums with bounded variance in the case of a not necessarily independent identically distributed input sequence.

Theorem 4.1. *For a finite, finally connected and finally aperiodic Markov chain $\mathcal{M}$ with an output function k, the following assertions are equivalent:*

(a) The asymptotic variance v of the output sum is 0.

(b) There exists a state s of the final component and a constant $a \in \mathbb{R}$ such that

$$k(C) = a\mathbb{1}(C)$$

holds for every closed walk C of the final component visiting the state s exactly once.

(c) There exists a constant $a \in \mathbb{R}$ such that

$$k(C) = a\mathbb{1}(C)$$

holds for every directed cycle C of the final component of $\mathcal{M}$.

In that case, $an + \mathcal{O}(1)$ is the expected value of the output sum and Statement (b) holds for all states s of the final component.

If $\mathcal{M}$ is furthermore strongly connected, the following assertion is also equivalent:

(d) The random variable K_n is quasi-deterministic with constant a.

This theorem is proved in Section 4.5.

In the case that the value of the output function is 0 or 1 for each transition, there are only two trivial output functions with asymptotic variance zero. This corollary is proved in Section 4.5.

Corollary 4.3.2. *Let $k\colon \mathcal{E} \to \{0,1\}$. Then the asymptotic variance v is zero if and only if the output function k is constant on the final component.*

The next theorem extends Theorem 4.1 to the joint distribution of several simultaneous output sums by combinatorically describing the case of a singular variance-covariance matrix.

Theorem 4.2. *Let $\mathcal{M}$ be a finite, finally connected, finally aperiodic Markov chain with m output functions $k_1, \dots, k_m$. Then the variance-covariance matrix Σ is regular if and only if the functions $\mathbb{1}, k_1, \dots, k_m$ are linearly independent as functions from the vector space of cycles of the final component to the real numbers, i.e. there do not exist real constants $a_0, \dots, a_m$, not all zero, such that*

$$a_0\mathbb{1}(C) + a_1k_1(C) + \cdots + a_mk_m(C) = 0 \tag{4.1}$$

holds for all cycles (or equivalently, for all closed walks) C of the final component.

The random variables $K_n^{(1)}, \dots, K_n^{(m)}$ are asymptotically jointly normally distributed if and only if Σ is regular.

This theorem is proved in Section 4.5.

Remark 4.3.3. Theorems 4.1 and 4.2 and Corollary 4.3.2 are independent of the choice of the probabilities of the transitions. Only the structure of the underlying graph of the Markov chain and the output functions influence the result. Note, however, that according to our general assumptions, all transitions have *positive* probability.

The next theorem gives a combinatorial characterization of output functions of a Markov chain which are asymptotically independent. As this characterization is given by the covariance, we can restrict ourselves to two output functions without loss of generality.

Theorem 4.3. *Let $\mathcal{M}$ be a finite, finally connected, finally aperiodic Markov chain with two output functions k_1 and k_2.*

Then the random variables $K_n^{(1)}$ and $K_n^{(2)}$ have the expected values $e_1n + \mathcal{O}(1)$ and $e_2n + \mathcal{O}(1)$, respectively, where the constants are

$$\begin{aligned} e_1 &= \frac{k_1(\mathcal{D}_1)}{\mathbb{1}(\mathcal{D}_1)}, \\ e_2 &= \frac{k_2(\mathcal{D}_1)}{\mathbb{1}(\mathcal{D}_1)}. \end{aligned} \tag{4.2}$$

The variances and the covariance are $v_1n + \mathcal{O}(1)$, $v_2n + \mathcal{O}(1)$ and $cn + \mathcal{O}(1)$, with the constants

$$\begin{aligned} v_1 &= \frac{1}{\mathbb{1}(\mathcal{D}_1)}\big((k_1 - e_1\mathbb{1}, k_1 - e_1\mathbb{1})(\mathcal{D}_1) - (k_1 - e_1\mathbb{1}, k_1 - e_1\mathbb{1})(\mathcal{D}_2)\big), \\ v_2 &= \frac{1}{\mathbb{1}(\mathcal{D}_1)}\big((k_2 - e_2\mathbb{1}, k_2 - e_2\mathbb{1})(\mathcal{D}_1) - (k_2 - e_2\mathbb{1}, k_2 - e_2\mathbb{1})(\mathcal{D}_2)\big), \end{aligned}$$

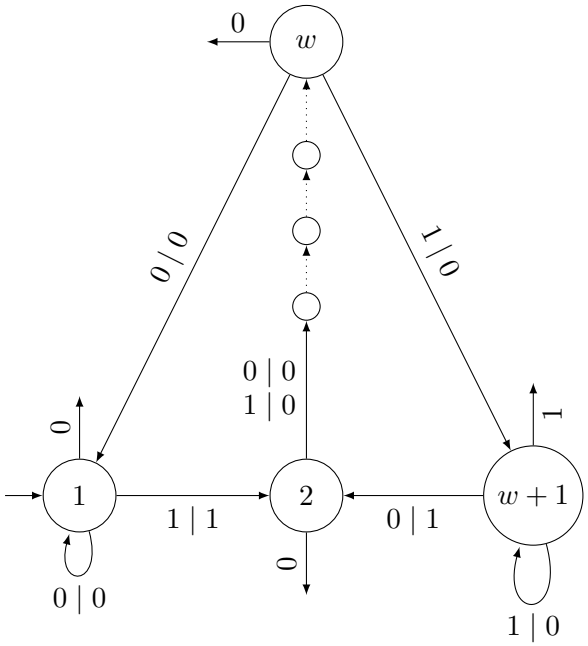


FIGURE 4.1. Transducer $\mathcal{T}(w)$ to compute the Hamming weight of the width-w non-adjacent form.

$$c = \frac{1}{\mathbb{1}(\mathcal{D}_1)}\big((k_1 - e_1\mathbb{1}, k_2 - e_2\mathbb{1})(\mathcal{D}_1) - (k_1 - e_1\mathbb{1}, k_2 - e_2\mathbb{1})(\mathcal{D}_2)\big).$$

The random variables $K_n^{(1)}$ and $K_n^{(2)}$ are asymptotically independent if and only if

$$(k_1 - e_1\mathbb{1}, k_2 - e_2\mathbb{1})(\mathcal{D}_1) = (k_1 - e_1\mathbb{1}, k_2 - e_2\mathbb{1})(\mathcal{D}_2).$$

This theorem is proved in Section 4.5.

In the case that the expected values of $K_n^{(1)}$ and $K_n^{(2)}$ are both bounded, i.e. $e_1 = e_2 = 0$, these random variables are asymptotically independent if and only if

$$(k_1, k_2)(\mathcal{D}_1) = (k_1, k_2)(\mathcal{D}_2).$$

4.4. Examples

In this section, we first prove the asymptotic joint normal distribution of two Hamming weights of different digit expansions by using Theorem 4.2. Then we investigate the independence of length 2 blocks of 0-1-sequences by using Theorem 4.3. In both cases we start with two transducers to construct a Markov chain with two output functions, once as a Cartesian product, once via Remark 4.2.2.

Example 4.4.1 (Width-w non-adjacent forms)**.** Let $2 \le w_1 < w_2$ be integers. We consider the asymptotic joint distribution of the Hamming weight of the width-w_1 non-adjacent form (w_1-NAF) and the Hamming weight of the w_2-NAF. It will turn out that this distribution is normal if and only if the variance-covariance matrix is regular. Using Theorem 4.2, we have to find closed walks in the corresponding Markov chain such that all coefficients in (4.1) have to be zero.

The transducer $\mathcal{T}(w)$ in Figure 4.1 computes the Hamming weight of the w-NAF of the integer n when the input is the binary expansion of n (cf. [50]). It has $w+1$ states. Next, we construct the Cartesian product of the transducers for w_1 and w_2 and choose any non-degenerate probability distribution, i.e. with all probabilities non-zero, for the outgoing transitions of a state. Thus, we obtain a Markov chain $\mathcal{M}$ with $(w_1+1)(w_2+1)$ states with two different output functions h_1 and h_2 corresponding to the outputs of the transducers for w_1 and w_2, respectively. We can now use Theorem 4.2 to prove that these two Hamming weights are asymptotically jointly normally distributed.

The Cartesian product of two closed walks in $\mathcal{T}(w_1)$ and $\mathcal{T}(w_2)$ with the same input sequence is a closed walk in $\mathcal{M}$. We construct three different closed walks and prove that all three coefficients in (4.1) have to be zero. For brevity, we denote a closed walk in the Cartesian product $\mathcal{M}$ and its projections to $\mathcal{T}(w_1)$ and $\mathcal{T}(w_2)$ by the same letter.

First, we choose the closed walk C_1 starting in state 1 with input sequence 0. We obtain $h_1(C_1)=0$ in $\mathcal{T}(w_1)$, $h_2(C_1)=0$ in $\mathcal{T}(w_2)$ and $\mathbb{1}(C_1)=1$. Second, we choose the closed walk C_2 starting in 1 with input sequence 10^{w_2-1}. Because $w_1<w_2$ and the loop at state 1, C_2 is a closed walk in $\mathcal{T}(w_1)$ and $\mathcal{T}(w_2)$. We obtain $h_1(C_2)=1$ in $\mathcal{T}(w_1)$, $h_2(C_2)=1$ in $\mathcal{T}(w_2)$ and $\mathbb{1}(C_2)=w_2$. The third choice depends on whether $w_1=w_2-1$ or not:

- $w_1 \neq w_2-1$: We choose the closed walk C_3 starting in 1 with input sequence $10^{w_1-1}10^{w_1-1}0^{\alpha}$ where $\alpha=\max(w_2-2w_1,0)$. On the one hand, this is a closed walk in $\mathcal{T}(w_1)$ consisting of two times the cycle $1\to w_1\to 1$ and α times the loop at state 1. On the other hand, this is a closed walk in $\mathcal{T}(w_2)$ consisting of the cycle $1\to w_2\to 1$ and the correct number of loops at state 1. We obtain $h_1(C_3)=2$ in $\mathcal{T}(w_1)$, $h_2(C_3)=1$ in $\mathcal{T}(w_2)$ and $\mathbb{1}(C_3)=\max(w_2,2w_1)$.
- $w_1=w_2-1$: We choose the closed walk C_3 starting in 1 with input sequence $10^{w_1-1}10^{w_1-1}10^{w_1-1}$. On the one hand, this is a closed walk in $\mathcal{T}(w_1)$ consisting of three times the cycle $1\to w_1\to 1$. On the other hand, this is a closed walk in $\mathcal{T}(w_2)$ consisting of the closed walk $1\to w_2\to w_2+1\to w_2\to 1$ and the correct number of loops at state 1. We obtain $h_1(C_3)=3$ in $\mathcal{T}(w_1)$, $h_2(C_3)=2$ in $\mathcal{T}(w_2)$ and $\mathbb{1}(C_3)=3w_1$.

This yields a system of linear equations for the coefficients a_0, a_1 and a_2 with coefficient matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ w_2 & 1 & 1 \\ \max(w_2,2w_1) & 2 & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 1 & 0 & 0 \\ w_2 & 1 & 1 \\ 3w_1 & 3 & 2 \end{pmatrix},$$

which only has the trivial solution. Thus, the Hamming weights of the w_1-NAF and the w_2-NAF are asymptotically jointly normally distributed, independently of the choice of the distributions for the Markov chain.

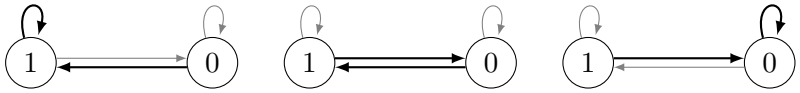
The next two examples investigate the asymptotic independence of length two blocks of 0-1-sequences.

Example 4.4.2 (10- and 11-blocks)**.** The two transducers in Figure 4.2 count the number of 10- and 11-blocks in 0-1-sequences. After deleting the outputs, both transducers are the same. Thus, any non-degenerate probability distribution on the outgoing edges of the states gives a Markov chain with two output functions k_{10} (for the 10-blocks) and k_{11} (for the 11-blocks).

Because of the two loops and the cycle $0\to 1\to 0$, Theorem 4.2 implies that the number of 10- and 11-blocks is asymptotically normally distributed.

(1) 10-blocks (2) 11-blocks

FIGURE 4.2. Transducers to compute the number of 10- and 11-blocks.



(1) $\mathcal{D}_1$

(2) $\mathcal{D}_2$

FIGURE 4.3. Functional digraphs of the transducers of Examples 4.4.2 and 4.4.3.

The next question is: For which choices of probability distributions is the number of 10- and 11-blocks asymptotically independent? All functional digraphs with one or two components are given in Figure 4.3. Using Theorem 4.3, we obtain the following system of equations for the values of the probabilities such that the numbers of 11-blocks and 10-blocks are asymptotically independent: first by definition

$$\begin{aligned} 1 &= p_{0\to 0} + p_{0\to 1}, \\ 1 &= p_{1\to 0} + p_{1\to 1}, \end{aligned}$$

then by (4.2)

$$\begin{aligned} e_{10} &= \frac{p_{0\to 1}p_{1\to 0}}{p_{0\to 1}p_{1\to 1} + 2p_{0\to 1}p_{1\to 0} + p_{0\to 0}p_{1\to 0}}, \\ e_{11} &= \frac{p_{0\to 1}p_{1\to 1}}{p_{0\to 1}p_{1\to 1} + 2p_{0\to 1}p_{1\to 0} + p_{0\to 0}p_{1\to 0}}, \end{aligned}$$

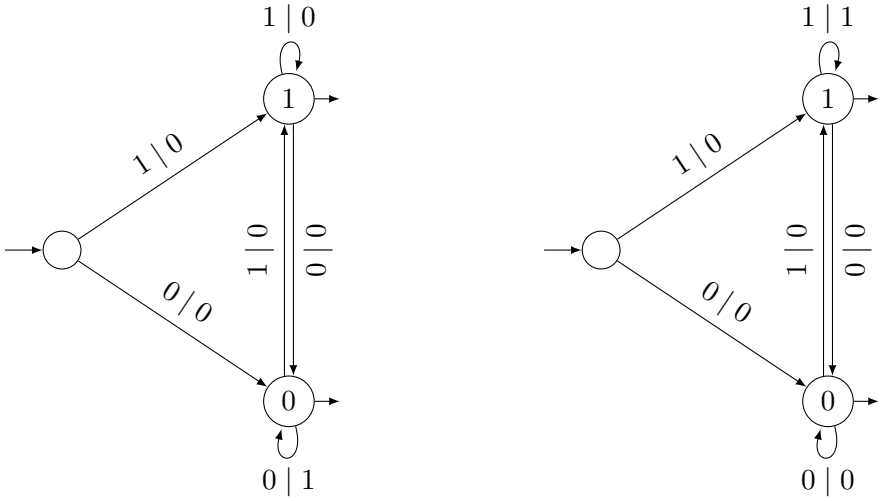
and finally for the independence

$$\begin{aligned} &p_{0\to 1}p_{1\to 1}(-e_{10})(1-e_{11}) + p_{0\to 1}p_{1\to 0}(1-2e_{10})(-2e_{11}) + p_{0\to 0}p_{1\to 0}(-e_{10})(-e_{11}) \\ &\quad = p_{0\to 0}p_{1\to 1}(-e_{10})(-e_{11}) + p_{0\to 0}p_{1\to 1}(-e_{10})(1-e_{11}). \end{aligned}$$

This system has non-trivial real solutions, i.e. solutions where all probabilities are non-zero, with

$$p_{0\to 0} = -\frac{1}{2}p_{1\to 1} + 2 - \frac{1}{2}\sqrt{p_{1\to 1}^2 - 8p_{1\to 1} + 8}$$

for all $0 < p_{1\to 1} < 1$. Then we have $2 - \sqrt{2} < p_{0\to 0} < 1$.



(1) 00-blocks (2) 11-blocks

FIGURE 4.4. Transducers to compute the number of 00- and 11-blocks.

Thus, for these transition probabilities, the number of 10-blocks and the number of 11-blocks are asymptotically independent.

One such example of a non-trivial solution is $p_{1\to1} = p_{1\to0} = 0.5$, $p_{0\to0} \approx 0.7192$ and $p_{0\to1} \approx 0.2808$. Note that for the symmetric distributions $p_{0\to0} = p_{0\to1} = p_{1\to1} = p_{1\to0} = 0.5$, we obtain asymptotic dependence of the number of 10- and 11-blocks.

Example 4.4.3 (00- and 11-blocks)**.** The two transducers in Figure 4.4 count the number of 00- and 11-blocks in 0-1-sequences. They have the same underlying graph and the same input labels. Thus, choosing any non-degenerate probability distribution of the outgoing edges of the states yields a Markov chain with two output functions.

Because of the two loops and the cycle $0 \to 1 \to 0$, Theorem 4.2 implies that the number of 00- and 11-blocks is asymptotically normally distributed.

The next question is: For which choices of probability distributions is the number of 00- and 11-blocks asymptotically independent? The functional digraphs of the final component are the same as in Example 4.4.2, see again Figure 4.3. By Theorem 4.3, the system of equations for the transition probabilities p_e such that the two output functions are asymptotically independent are: first by definition

$$1 = p_{0\to0} + p_{0\to1},$$
$$1 = p_{1\to0} + p_{1\to1},$$

then by (4.2)

$$e_{00} = \frac{p_{0\to0}p_{1\to0}}{p_{0\to1}p_{1\to1} + 2p_{0\to1}p_{1\to0} + p_{0\to0}p_{1\to0}},$$
$$e_{11} = \frac{p_{0\to1}p_{1\to1}}{p_{0\to1}p_{1\to1} + 2p_{0\to1}p_{1\to0} + p_{0\to0}p_{1\to0}},$$

and finally for the independence

$$p_{0\to1}p_{1\to1}(-e_{00})(1-e_{11}) + p_{0\to1}p_{1\to0}(-2e_{00})(-2e_{11}) + p_{0\to0}p_{1\to0}(1-e_{00})(-e_{11})$$

$$= p_{0\to0}p_{1\to1}(1-e_{00})(1-e_{11}) + p_{0\to0}p_{1\to1}(-e_{00})(-e_{11}).$$

These equations have no solution with $0 < p_e < 1$ for all transitions e. Thus, the numbers of 00- and 11-blocks are asymptotically dependent for all choices of the input distributions, as expected.

4.5. Proofs

In this section, we prove the results from Section 4.3. The proofs follow along the same ideas as in [56] and Chapter 3. The main differences are that one has to replace "complete transducer" by "Markov chain" and the input sum by the output sum $K_n^{(1)}$. For the sake of completeness, we state the proofs here.

We first prove Theorem 4.3 with the help of two lemmas. For one of these lemmas, we use a version of the Matrix-Tree Theorem for weighted directed forests proved in [16, 75]. At the end of this section, we prove Theorems 4.1 and 4.2.

Definition 4.5.1. Let A, $B \subseteq \{1,\dots,N\}$. Let $\mathcal{F}_{A,B}$ be the set of all forests which are spanning subgraphs of the final component of the Markov chain $\mathcal{M}$ with $|A|$ trees such that every tree is rooted at some vertex $a \in A$ and contains exactly one vertex $b \in B$.

Let $A = \{i_1,\dots,i_n\}$ and $B = \{j_1,\dots,j_n\}$ with $i_1 < \cdots < i_n$ and $j_1 < \cdots < j_n$. For $F \in \mathcal{F}_{A,B}$, we define a function $g\colon B \to A$ by $g(j) = i$ if j is in the tree of F which is rooted in vertex i. We further define the function $h\colon A \to B$ by $h(i_k) = j_k$ for $k = 1,\dots,n$. The composition $g \circ h\colon A \to A$ is a permutation of A. We define $\operatorname{sgn} F = \operatorname{sgn} g \circ h$.

If $|A| \neq |B|$, then $\mathcal{F}_{A,B} = \emptyset$. If $|A| = |B| = 1$, then $\operatorname{sgn} F = 1$ and $\mathcal{F}_{A,B}$ consists of all spanning trees rooted in $a \in A$.

Theorem (All-Minors-Matrix-Tree Theorem [16, 75])**.** *For a directed, weighted graph with loops and multiple edges, let $L = (l_{ij})_{1\le i,j\le N}$ be the Laplacian matrix, that is $\sum_{j=1}^{N} l_{ij} = 0$ for every $i = 1,\dots,N$ and $-l_{ij}$ is the sum of the weights p_e of all edges e from i to j for $i \neq j$. Then, for $|A| = |B|$, the minor $\det L_{A,B}$ satisfies*

$$\det L_{A,B} = (-1)^{\sum_{i\in A} i + \sum_{j\in B} j} \sum_{F\in\mathcal{F}_{A,B}} p_F \operatorname{sgn} F$$

where $L_{A,B}$ is the matrix L whose rows with index in A and columns with index in B are deleted.

The All-Minors-Matrix-Tree Theorem is still valid for $|A| \neq |B|$ if we assume that the determinant of a non-square matrix is 0. For notational simplicity, we use this convention in the rest of this section.

Definition 4.5.2. The transition matrix $W(x_1,\dots,x_m)$ of a Markov chain with M states and m output functions $k_1, \dots, k_m$ is a $M \times M$ matrix whose (i,j)-th entry is

$$\sum_{e\colon i\to j} p_e x_1^{k_1(e)} \cdots x_m^{k_m(e)}$$

where p_e is the probability of the transition e.

Let $A(x_1,\dots,x_m)$ be the $N \times N$ transition matrix of the final component of the Markov chain. Let the order of the states be such that the transition matrix of the whole Markov

chain $W(x_1, \dots, x_m)$ has the block structure

$$W(x_1, \dots, x_m) = \begin{pmatrix} * & * \\ 0 & A(x_1, \dots, x_m) \end{pmatrix} \tag{4.3}$$

where $*$ denotes any matrix. If the Markov chain is strongly connected, the matrices $*$ are not present (they have 0 rows).

We first use the All-Minors-Matrix-Tree Theorem to connect the derivatives of the characteristic polynomial of the transition matrix with a sum of weighted digraphs in the next lemma.

Lemma 4.5.3. *For $f(x_1, x_2, z) = \det(I - zA(x_1, x_2))$, we have*

$$\begin{aligned} f_{x_1}(1,1,1) &= -k_1(\mathcal{D}_1), & f_{x_1x_2}(1,1,1) &= (k_1,k_2)(\mathcal{D}_2) - (k_1,k_2)(\mathcal{D}_1), \\ f_{x_2}(1,1,1) &= -k_2(\mathcal{D}_1), & f_{x_1z}(1,1,1) &= (k_1,\mathbb{1})(\mathcal{D}_2) - (k_1,\mathbb{1})(\mathcal{D}_1), \\ f_z(1,1,1) &= -\mathbb{1}(\mathcal{D}_1), & f_{x_2z}(1,1,1) &= (k_2,\mathbb{1})(\mathcal{D}_2) - (k_2,\mathbb{1})(\mathcal{D}_1), \\ f_{x_1x_1}(1,1,1) + f_{x_1}(1,1,1) &= (k_1,k_1)(\mathcal{D}_2) - (k_1,k_1)(\mathcal{D}_1), \\ f_{x_2x_2}(1,1,1) + f_{x_2}(1,1,1) &= (k_2,k_2)(\mathcal{D}_2) - (k_2,k_2)(\mathcal{D}_1), \\ f_{zz}(1,1,1) + f_z(1,1,1) &= (\mathbb{1},\mathbb{1})(\mathcal{D}_2) - (\mathbb{1},\mathbb{1})(\mathcal{D}_1). \end{aligned}$$

Proof. The idea of the proof is as follows: We start writing the derivatives as sums over all states. Using the All-Minors-Matrix-Tree Theorem, we rewrite this into a weighted sum over forests and then into a weighted sum over functional digraphs.

Let u_1, u_2 be any of the variables x_1, x_2 or z. For a matrix $M = (m_{ij})_{1 \le i,j \le N}$, we define matrices $M_{a:u_1} = (\hat{m}_{ij})_{1 \le i,j \le N}$ with $\hat{m}_{ij} = m_{ij}$ for $i \neq a$ and $\hat{m}_{aj} = \frac{\partial}{\partial u_1} m_{aj}$ otherwise. Thus $M_{a:u_1}$ is the matrix M where row a is differentiated with respect to u_1.

We further define the derivatives at $\mathbf{1} = (1,1,1)$ as

$$D_{u_1}(\,\cdot\,) = \frac{\partial}{\partial u_1}(\,\cdot\,)\Big|_{\mathbf{1}}$$

and

$$D_{u_1u_2}(\,\cdot\,) = \frac{\partial^2}{\partial u_1 \partial u_2}(\,\cdot\,)\Big|_{\mathbf{1}}.$$

Applying the product rule to the definition of the determinants gives us

$$\begin{aligned} D_{u_1}(f) &= \sum_{j=1}^{N} \det(I - zA(x_1,x_2))_{j:u_1}\Big|_{\mathbf{1}}, \\ D_{u_1u_2}(f) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \det(I - zA(x_1,x_2))_{i:u_1,\, j:u_2}\Big|_{\mathbf{1}}. \end{aligned}$$

In these equations, we have sums over all states.

Later, we will use Laplace expansion along row j to determine the determinants $\det(I - zA(x_1,x_2))_{j:u_1}$. If $i \neq j$, we will use Laplace expansion along rows i and j to determine $\det(I - zA(x_1,x_2))_{i:u_1,\, j:u_2}$. If $i = j$, we will only expand along row j.

For a transition e, we denote by $t(e)$ and $h(e)$ the tail and the head of the transition e, respectively. For brevity, we write $w_e = p_e x_1^{k_1(e)} x_2^{k_2(e)} z$ for a transition e.

If we use Laplace expansion along two different rows, we must be careful with the sign. Therefore, we define

$$\sigma_{de} = (-1)^{[t(e)>t(d)]+[h(e)>h(d)]}$$

for two transitions d and e. Here, we use Iverson's notation, that is [*expression*] is 1 if *expression* is true and 0 otherwise (cf. [41]).

Let L be the Laplacian of the underlying graph of the final component of the Markov chain with the probabilities as weights, that is $L = I - A(1,1)$.

Recall the notation $L_{A,B}$ for the matrix where the rows corresponding to A and the columns corresponding to B have been removed. Laplace expansion yields

$$\begin{aligned}
D_{u_1}(f) &= -\sum_{j=1}^{N} \sum_{\substack{e\in\mathcal{E}\\ t(e)=j}} (-1)^{t(e)+h(e)} D_{u_1}(w_e) \det(L_{\{t(e)\},\{h(e)\}}),\\
D_{u_1u_2}(f) &= \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} \sum_{\substack{e\in\mathcal{E}\\ t(e)=j}} \sum_{\substack{d\in\mathcal{E}\\ t(d)=i}} \Big((-1)^{t(e)+h(e)+t(d)+h(d)} \sigma_{de} D_{u_1}(w_e) D_{u_2}(w_d)\\
&\qquad\qquad \times \det(L_{\{t(e),t(d)\},\{h(e),h(d)\}})\Big)\\
&\quad -\sum_{j=1}^{N} \sum_{\substack{e\in\mathcal{E}\\ t(e)=j}} (-1)^{t(e)+h(e)} D_{u_1u_2}(w_e) \det(L_{\{t(e)\},\{h(e)\}}).
\end{aligned}$$

Next, we use the All-Minors-Matrix-Tree Theorem and change the summation over all states to a summation over forests. We obtain

$$\begin{aligned}
D_{u_1}(f) &= -\sum_{e\in\mathcal{E}} D_{u_1}(w_e) \sum_{F\in\mathcal{F}_{\{t(e)\},\{h(e)\}}} p_F \operatorname{sgn} F,\\
D_{u_1u_2}(f) &= \sum_{e\in\mathcal{E}} \sum_{\substack{d\in\mathcal{E}\\ d\neq e}} \sigma_{de} D_{u_1}(w_e) D_{u_2}(w_d) \sum_{F\in\mathcal{F}_{\{t(d),t(e)\},\{h(d),h(e)\}}} p_F \operatorname{sgn} F\\
&\quad -\sum_{e\in\mathcal{E}} D_{u_1u_2}(w_e) \sum_{F\in\mathcal{F}_{\{t(e)\},\{h(e)\}}} p_F \operatorname{sgn} F.
\end{aligned}$$

Let $F \in \mathcal{F}_{\{t(e)\},\{h(e)\}}$ be a forest for a transition $e \in \mathcal{E}$. Then $F+e$ is a spanning functional digraph with one component. Let $F \in \mathcal{F}_{\{t(d),t(e)\},\{h(d),h(e)\}}$ be a forest for transitions $d, e \in \mathcal{E}$. Then $F + d + e$ is a spanning functional digraph with one or two components, depending on $\sigma_{de} \operatorname{sgn} F$. If $\sigma_{de} \operatorname{sgn} F = 1$, then it has two components. Otherwise, it has one component. Now we can change the summation into a sum over functional digraphs and obtain

$$\begin{aligned}
D_{u_1}(f) &= -\sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} \sum_{e\in C} p_{D\setminus\{e\}} D_{u_1}(w_e),\\
D_{u_1u_2}(f) &= \sum_{D\in\mathcal{D}_2} \sum_{C_1\in\mathcal{C}_D} \sum_{\substack{C_2\in\mathcal{C}_D\\ C_2\neq C_1}} \sum_{e\in C_1} \sum_{d\in C_2} p_{D\setminus\{e,d\}} D_{u_1}(w_e) D_{u_2}(w_d)\\
&\quad -\sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} \sum_{e\in C} \sum_{\substack{d\in C\\ d\neq e}} p_{D\setminus\{d,e\}} D_{u_1}(w_e) D_{u_2}(w_d)
\end{aligned}$$

$$- \sum_{D\in\mathcal{D}_1} \sum_{C\in\mathcal{C}_D} \sum_{e\in C} p_{D\setminus\{e\}} D_{u_1u_2}(w_e).$$

For a transition e, we know the first derivatives

$$D_{x_1}(w_e) = p_e k_1(e), \qquad D_{x_2}(w_e) = p_e k_2(e), \qquad D_z(w_e) = p_e \mathbb{1}(e),$$

and the second derivatives

$$\begin{aligned} D_{x_1x_2}(w_e) &= p_e k_1(e)k_2(e), & D_{x_1x_1}(w_e) &= p_e k_1(e)(k_1(e)-1),\\ D_{x_1z}(w_e) &= p_e k_1(e)\mathbb{1}(e), & D_{x_2x_2}(w_e) &= p_e k_2(e)(k_2(e)-1),\\ D_{x_2z}(w_e) &= p_e k_2(e)\mathbb{1}(e), & D_{zz}(w_e) &= 0. \end{aligned}$$

As the probabilities $p_D = p_{D\setminus\{e\}}p_e$ are multiplicative, we obtain the formulas stated in the lemma. □

The following lemma will be used for $m \geq 2$ output functions later on.

Lemma 4.5.4. *Let $f(x_1,\dots,x_m,z) = \det(I - zA(x_1,\dots,x_m))$. Then there is a unique dominant root $z = \rho(x_1,\dots,x_m)$ of f in a neighborhood of $(1,\dots,1)$.*

The moment generating function of $(K_n^{(1)},\dots,K_n^{(m)})$ has the asymptotic expansion

$$\mathbb{E}(\exp(s_1K_n^{(1)} + \cdots + s_mK_n^{(m)})) = e^{u(s_1,\dots,s_m)n+v(s_1,\dots,s_m)}(1+\mathcal{O}(\kappa^n))$$

where $\kappa < 1$,

$$u(s_1,\dots,s_m) = -\log\rho(e^{s_1},\dots,e^{s_m}),$$

and $v(s_1,\dots,s_m)$ are analytic functions in a small neighborhood of $(0,\dots,0)$.

Proof. The moment generating function of $(K_n^{(1)},\dots,K_n^{(m)})$ is

$$\mathbb{E}(\exp(s_1K_n^{(1)} + \cdots + s_mK_n^{(m)})) = [z^n]v_1^t(I - zW(e^{s_1},\dots,e^{s_m}))^{-1}v_2(e^{s_1},\dots,e^{s_m})$$

for the initial vector v_1, and a vector $v_2(x_1,\dots,x_m)$ encoding all the final information of the states[2] where we write $[z^n]b(z)$ for the coefficient of z^n in the power series b. Because of the block structure of the transition matrix W of the whole Markov chain in (4.3), we obtain

$$\begin{aligned} \mathbb{E}(x_1^{K_n^{(1)}}\cdots x_m^{K_n^{(m)}}) &= [z^n]\frac{F_1(x_1,\dots,x_m,z)}{\det(I - zW(x_1,\dots,x_m))}\\ &= [z^n]\frac{F_1(x_1,\dots,x_m,z)}{F_2(x_1,\dots,x_m,z)f(x_1,\dots,x_m,z)} \end{aligned}$$

for "polynomials" F_1 and F_2 , i.e. finite linear combinations of $x_1^{\alpha_1}\cdots x_m^{\alpha_m}z^\beta$ for $\alpha_i \in \mathbb{R}$ and β a non-negative integer. The function F_2 corresponds to the determinant of the non-final part of the Markov chain.

We obtain the coefficient of z^n by singularity analysis (cf. [30]). Since the final component of $\mathcal{M}$ is again a Markov chain, the dominant singularity of $1/f(1,\dots,1,z)$ is 1 by the theorem of Perron–Frobenius (cf. [33]). By the aperiodicity of the final component, this dominant singularity is unique and it is $\rho(1,\dots,1) = 1$.

Next, we consider the non-final components of the Markov chain using the same arguments as in [56] and Chapter 3. The corresponding non-final component $\mathcal{M}_0$ is not a Markov chain

[2]This information is the final output (see Remark 4.2.3) and the exit weight (see Remark 4.2.4) included as $w_i x_1^{f_1(i)}\cdots x_m^{f_m(i)}$ in the i-th coordinate of $v_2(x_1,\dots,x_m)$. This does not change the asymptotic behavior (see Remark 4.5.5).

as the transition matrix is not stochastic. Let $\mathcal{M}_0^+$ be the Markov chain that is obtained from $\mathcal{M}_0$ by adding loops with the missing probabilities where necessary. The dominant eigenvalue of the transition matrix of $\mathcal{M}_0^+$ is 1. As the transition matrices of $\mathcal{M}_0$ and $\mathcal{M}_0^+$ satisfy element-wise inequalities but are not equal (at $(x_1, \dots, x_m) = (1, \dots, 1)$), the theorem of Perron–Frobenius (cf. [33, Theorem 8.8.1]) implies that the dominant eigenvalues of $\mathcal{M}_0$ have absolute value less than 1. Thus, the dominant singularities of $F_2(1, \dots, 1, z)^{-1}$ are at $|z| > 1$.

As $A(1, \dots, 1, z) = (1 - z)^{-1}$, we obtain $F_1(1, \dots, 1) \neq 0$.

Thus, there is a is the unique, dominant singularity of

$$\frac{F_1(1, \dots, 1, z)}{F_2(1, \dots, 1, z) f(1, \dots, 1, z)},$$

which is $\rho(1, \dots, 1) = 1$. This also holds for $(x_1, \dots, x_m)$ in a small neighborhood of $(1, \dots, 1)$ by the continuity of the eigenvalues of the transition matrices. Thus, $\rho(x_1, \dots, x_m)$ is this unique dominant singularity.

Now, singularity analysis (cf. [30]) implies the statement of this lemma. □

Remark 4.5.5. The main term of the asymptotic expansion of the moment generating function only depends on $\rho(x_1, \dots, x_m)$ and therefore on $f(x_1, \dots, x_m, z)$. It does not depend on the "polynomials" $F_1(x_1, \dots, x_m, z)$ and $F_2(x_1, \dots, x_m, z)$. Thus, only the final component influences the main term. Neither the states in the non-final part of the Markov chain nor the final outputs and exit weights influence the main term.

Now, we can use the previous two lemmas to prove Theorem 4.3.

Proof of Theorem 4.3. By Lemma 4.5.4 for two output functions k_1 and k_2, the moment generating function satisfies the conditions of the Quasi-Power Theorem [56, Theorem 5.1] or Theorem 3.6, which yields the expected value

$$\mathbb{E}(K_n^{(1)}, K_n^{(2)}) = n \operatorname{grad} u(\mathbf{0}) + \mathcal{O}(1)$$

and the variance

$$\mathbb{V}(K_n^{(1)}, K_n^{(2)}) = n H_u(\mathbf{0}) + \mathcal{O}(1)$$

with $\operatorname{grad} u(\mathbf{0})$ and $H_u(\mathbf{0})$ the gradient and the Hessian of u at $\mathbf{0}$, respectively. Furthermore, we obtain an asymptotic joint normal distribution of the standardized random vector if the Hessian is not singular by [56, Theorem 3.9] or Theorem 3.6. Otherwise, the limiting random vector is either a pair of degenerate random variables, or a degenerate and normally distributed one, or a linear transformation thereof. Thus, the random variables $K_n^{(1)}$ and $K_n^{(2)}$ are asymptotically independent if and only if the covariance is zero.

By implicit differentiation, we obtain the following formulas for the constants of the moments in terms of the partial derivatives of f:

$$\begin{aligned}
e_1 &= \frac{f_{x_1}}{f_z}\Big|_{\mathbf{1}}, \\
e_2 &= \frac{f_{x_2}}{f_z}\Big|_{\mathbf{1}}, \\
v_1 &= \frac{1}{f_z^3}(f_{x_1}^2(f_{zz} + f_z) + f_z^2(f_{x_1x_1} + f_{x_1}) - 2f_{x_1}f_z f_{x_1z})\Big|_{\mathbf{1}}, \\
v_2 &= \frac{1}{f_z^3}(f_{x_2}^2(f_{zz} + f_z) + f_z^2(f_{x_2x_2} + f_{x_2}) - 2f_{x_2}f_z f_{x_2z})\Big|_{\mathbf{1}},
\end{aligned}$$

$$c = \frac{1}{f_z^3}(f_{x_1}f_{x_2}(f_{zz}+f_z) + f_z^2 f_{x_1x_2} - f_{x_2}f_z f_{x_1z} - f_{x_1}f_z f_{x_2z})\Big|_{\mathbf{1}}.$$

Now, Lemma 4.5.3 implies the results as stated in the theorem. □

Proof of Theorem 4.1. The equivalence of (b) and (c) is the same as in [56, Theorem 3.1] and Theorem 3.1.

(a) ⇔ (b): WLOG, we assume that the expected value of K_n is bounded (otherwise replace $k(e)$ by $k(e) - e_1$ for all transitions e and e_1 the constant in (4.2)). Under this assumption, Theorem 4.3 implies that (b) can only hold with $k = 0$.

As in the proof in [56, Theorem 3.1] and Theorem 3.1, (a) is equivalent to $C^s_{xx}(1,1) + C^s_x(1,1) = 0$ where

$$C^s(x,z) = \sum_{C\in\mathcal{C}^s} p_C x^{k(C)} z^{\mathbb{1}(C)}$$

is the generating function of the set $\mathcal{C}^s$ of all closed walks in the final component of $\mathcal{M}$ which visit state s exactly once where x marks the output sum of the function k and z marks the length of the walk.

This is equivalent to

$$\sum_{C\in\mathcal{C}^s} p_C k(C)^2 = 0$$

which is equivalent to $k(C) = 0$ for all $C \in \mathcal{C}^s$.

To prove the remaining equivalence, we prove the equivalence of the following two assertions for not necessarily strongly connected Markov chains.

(d) The random variable K_n is quasi-deterministic with constant a.

(e) There exists a constant $a \in \mathbb{R}$ such that

$$k(C) = a\mathbb{1}(C)$$

holds for every directed cycle C of the whole Markov chain $\mathcal{M}$.

Then the theorem follows immediately because the strongly connected underlying graph of the Markov chain implies (e) ⇔ (c).

(d) ⇒ (e): Let C be an arbitrary cycle of the Markov chain and P be a path from the initial state 1 to any state of the cycle. For some n, consider the path consisting of P and n times C. Its output sum with respect to k is then $k(P)+nk(C)$. This is a realization of the quasi-deterministic random variable $K_{\mathbb{1}(P)+n\mathbb{1}(C)}$ and thus fulfills

$$a(\mathbb{1}(P) + n\mathbb{1}(C)) + \mathcal{O}(1) = k(P) + nk(C).$$

Thus, $n(k(C) - a\mathbb{1}(C))$ is bounded by a constant depending on P and C, but independent of n. Therefore, we know that $k(C) = a\mathbb{1}(C)$.

(e) ⇒ (d): WLOG, we assume $a = 0$ (otherwise replace $k(e)$ by $k(e) - a$ for all transitions e). All cycles have output sum 0 so that every transition contributes at most once to K_n. Thus, for every path X_n, we have $|K_n| \le \sum_{e\in\mathcal{E}}|k(e)| + \max_{s\in\{1,\dots,M\}}|f(s)|$ where f is the final output function (see Remark 4.2.3). Therefore, we have a quasi-deterministic random variable $K_n = \mathcal{O}(1)$.

□

Proof of Corollary 4.3.2. This follows by the same arguments as in [56, Corollary 3.6] and Corollary 3.3.4. □

PROOF OF THEOREM 4.2. WLOG, we assume that $\mathbb{E}K_n^{(i)} = \mathcal{O}(1)$ for $i = 1, \dots, m$ by subtracting the corresponding constant of the expected value from each output function. There exists a unitary matrix $T = (t_{ji})_{1 \le j,i \le m}$ such that the variance-covariance matrix Σ can be diagonalized as $T\Sigma T^\top = D$. The diagonal matrix D is the variance-covariance matrix of the linearly transformed random vector $\boldsymbol{Y}_n = T\boldsymbol{K}_n$.

Then Σ is singular if and only if the diagonal matrix D is singular. This is equivalent to

$$\mathbb{V}(t_{j1}K_n^{(1)} + \cdots + t_{jm}K_n^{(m)}) = \mathcal{O}(1) \tag{4.4}$$

holds for a $j \in \{1, \dots, m\}$. Now consider the output function $t_{j1}k_1 + \cdots + t_{jm}k_m$. By Theorem 4.1, (4.4) is equivalent to

$$t_{j1}k_1(C) + \cdots + t_{jm}k_m(C) = 0$$

holds for all cycles of the final component (since the expected value of this output function is $\mathcal{O}(1)$).

If we shift back the output function such that the expected value is no longer bounded, we obtain an additional summand $a_0\mathbb{1}(C)$.

The asymptotic joint normal distribution follows from Lemma 4.5.4 and the multidimensional Quasi-Power Theorem [24, Theorem 2.22]. □

CHAPTER 5

Analysis of Carries in Signed Digit Expansions

In this chapter, the number of positive and negative carries in the addition of two independent random signed digit expansions of given length is analyzed asymptotically for the (q, d)-system and the symmetric signed digit expansion. The number of iterations in von Neumann's parallel addition method for the symmetric signed digit expansion is also analyzed. One purpose of this chapter is to provide the theoretical foudations such that the actual analysis can be performed algorithmically.

Obtaining the values of the constants occuring in the asymptotic analysis of standard and von Neumann's addition requires computations involving finite state machines and determinants of matrices in several variables. These computations are performed using the mathematical software system SageMath [96]. Notebooks containing all the computations can be found at [54]. However, the existence of these constants follows from the theoretical results.

This chapter corresponds to [53], which is submitted for publication. This is joint work with Clemens Heuberger and Helmut Prodinger.

5.1. Introduction

We consider two types of digit expansions: On the one hand, we investigate (q, d)-expansions, that are q-ary digit expansions with digit set $\{d, \dots, q + d - 1\}$. With $d = 0$, this includes the case of the standard q-ary expansion. Consecutive digits are independent in this case. On the other hand, the symmetric signed digit expansion [58] has an even base q and the redundant digit set $\{-q/2, \dots, q/2\}$. To remove the redundancy, there is a syntactical rule to decide which of the digits $-q/2$ and $q/2$ is used. This rule introduces dependencies between consecutive digits.

Two different addition algorithms are investigated. The first one is the standard addition: We add two digits starting at the least significant position. If the result is not in the given digit set or does not fulfill the syntactical conditions, then a non-zero carry is produced. This carry is added to the sum of the two digits at the next position. An example for this standard addition of two decimal expansions is given in Table 5.1. In the case of positive and negative digits, positive and negative carries occur.

	2	1	4	6
	1_0	2_1	5_1	5_0
	3	4	0	1

TABLE 5.1. Example for standard addition in the decimal system. The subscripts in the second row are the carries.

5377	first summand
8125	second summand
3492	first interim result
10010	carries
13402	second interim result
000100	carries
013502	final result
0000000	

TABLE 5.2. Example for von Neumann's addition in the decimal system.

In contrast to standard addition, von Neumann's addition is a parallel algorithm with several iterations. The idea is to add the digits at each position in parallel (the interim result). If this result is not admissible in the given digit system, then a non-zero carry is produced and the interim result is corrected correspondingly at this position. However, this carry is not added immediately: The interim result and the carries are the input for the next iteration. When the carry sequence only contains zeros, then the algorithm terminates. An example for von Neumann's addition is shown in Table 5.2 for the addition of two decimal expansions.

The number of iterations of von Neumann's addition is of interest as it corresponds to the running time.

The outline of this chapter is as follows. In Section 5.2, we define (q, d)-expansions and symmetric signed digit expansions. We first analyze the standard addition in Sections 5.3–5.5. The algorithms and the corresponding transducers for the standard addition of (q, d)-expansions and symmetric signed digit expansions are presented in Section 5.3. Our probabilistic model is to choose both summands of length ℓ independently such that each expansion of length ℓ is equally likely. In the case of the symmetric signed digit expansions, the dependencies between the digits require approximating the equidistribution with an error that does not influence the final result. The corresponding probabilities for general regular languages are defined in Lemma 5.4.1 in Section 5.4, see also [93] and [83]. In Section 5.5, we combine this approximate equidistribution with the transducers from Section 5.3.2 to obtain an asymptotic analysis including the expectation, the variance and asymptotic normality in the main Theorems 5.1 and 5.2 for the (q, d)-system and the symmetric signed digit system, respectively.

Then, we analyze von Neumann's addition. We start in Section 5.6 with the algorithms and the automaton. Theorem 5.3 provides a general framework for the analysis of sequences occurring in this context. Then we again use the approximate equidistribution from Section 5.4 to asymptotically analyze the number of iterations of von Neumann's addition in Theorem 5.4 in Section 5.7. This analysis extends the results in [72] and [59] to the symmetric signed digit expansions and to include not only the expected value but also the variance and a convergence in distribution.

5.2. Digit Expansions

In this section, we define the digit expansions which will be used in later sections. We also recall their properties.

5.2.1. (q,d)-expansions.

Definition 5.2.1. Let $-q < d \leq 0$ be two integers with $q \geq 2$. The (q,d)-expansion of an integer x is the q-ary expansion $(x_\ell \dots x_0)_q$ with digits $x_i \in \{d, \dots, q+d-1\}$ such that $x = \sum_{i=0}^{\ell} x_i q^i$.

Example 5.2.2. The $(4,-1)$-expansion of 3 is $(1\bar{1})_4$, where we write $\bar{1}$ for the digit -1.

The (q,d)-expansion exists for all integers if $d \neq 0$ and $d \neq -q+1$. For $d=0$ (this is the standard q-ary expansion), only the non-negative integers have a (q,d)-expansion. Conversely, for $d=-q+1$, only the non-positive integers have a (q,d)-expansion. If the (q,d)-expansion of an integer exists, then it is unique up to leading zeros.

If q is odd and $d = \frac{-q+1}{2}$, then the (q,d)-expansion minimizes the sum of absolute values of the digits among all q-ary expansions with arbitrary digits (see [58]).

5.2.2. Symmetric Signed Digit Expansion. We recall the definition of the symmetric signed digit expansion (SSDE) as defined in [58] and further analyzed in [59].

Definition 5.2.3. Let $q \geq 2$ be an even integer. The symmetric signed digit expansion (SSDE) of an integer is the q-ary digit expansion $(x_\ell \dots x_0)_q$ with $x_i \in \{-\frac{q}{2}, \dots, \frac{q}{2}\}$ such that the syntactical rule

$$|x_j| = \frac{q}{2} \quad \Longrightarrow \quad 0 \leq \operatorname{sgn}(x_j) x_{j+1} \leq \frac{q}{2} - 1$$

is satisfied for $0 \leq j < \ell$.

In [58], it is shown that each integer n has a unique SSDE (up to leading zeros). It minimizes the sum of absolute values of the digits among all q-ary expansions of n with arbitrary digits (cf. [58]).

For $q=2$, we obtain the digit set $\{0, \pm 1\}$ and the syntactical rule that at least one of any two adjacent digits is zero. This digit expansion is also called non-adjacent form (cf. [89]).

5.3. Standard Addition

We write bold face letters for sequences which are padded with zeros on the left.

Let $\boldsymbol{x} = \dots x_1 x_0$ and $\boldsymbol{y} = \dots y_1 y_0$ be the two summands given as q-ary expansions with digit set D (possibly satisfying some syntactical rules). Then standard addition can be written in the form

$$\begin{array}{cccc} \dots & & x_1 & x_0 \\ \dots & c_2 & y_{1 c_1} & y_{0 c_0} \\ \hline \dots & & z_1 & z_0 \end{array}$$

where $x_i + y_i + c_i = z_i - q c_{i+1}$, $c_0 = 0$ with $z_i \in D$ and $\boldsymbol{z} = \dots z_1 z_0$ satisfying the syntactical rules of the digit system under consideration. We asymptotically analyze the sequence of carries $\boldsymbol{c} = \dots c_2 c_1$.

From a different point of view, the standard addition with digit set D is a conversion between different digit sets: We have a q-ary digit expansion with digits in $D+D$ and we

1	2	3	$\bar{1}$
1_1	2_0	$1_{\bar{1}}$	$\bar{1}_0$
3	$\bar{1}$	3	3

TABLE 5.3. Example for standard addition for $(5,-1)$-expansions. The subscripts in the second row are the carries.

1	$\bar{1}$	0	2
1_0	$\bar{1}_1$	1_1	2_0
2	$\bar{1}$	$\bar{2}$	0

TABLE 5.4. Example for standard addition for SSDEs for $q = 4$. The subscripts in the second row are the carries.

want to transform this digit expansion into a digit expansion with digit set D satisfying all syntactical rules. This can be written in the form

$\cdots$	c_2	$s_1 c_1$	$s_0 c_0$
$\cdots$		z_1	z_0

where $s_i = x_i + y_i \in D + D$. We call the sequence $\boldsymbol{s}$ the digitwise sum of $\boldsymbol{x}$ and $\boldsymbol{y}$ and write $\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{y}$.

We will mostly use this point of view. Most of the algorithms and transducers require the input of $\boldsymbol{s}$. If there are syntactical rules for $\boldsymbol{x}$ and $\boldsymbol{y}$, then the sequence $\boldsymbol{s}$ can not be arbitrary.

Remark 5.3.1. From this point of view, it is clear that interchanging two digits x_i and y_i of the two summands does not influence the result, but only both summands. The carries, the digitwise sum and the steps taken by the algorithms and the transducers stay the same as they depend only on the digitwise sum.

5.3.1. Algorithms.

5.3.1.1. *Standard Addition for (q,d)-expansions.* The digit set is $D = \{d, \ldots, q + d - 1\}$. Algorithm 5.1 transforms a q-ary expansion with digit set $D + D$ into a (q,d)-expansion. As there are no syntactical rules, all digits are independent. Thus, we do not have to look ahead when choosing the carry.

An example of standard addition for $(5,-1)$-expansions using this algorithm is given in Table 5.3.

5.3.1.2. *Standard Addition for SSDEs.* Let $q \geq 2$ be even. Algorithm 5.2 transforms a q-ary expansion with digit set $\{-q, \ldots, q\}$ into a SSDE. As the choice between the redundant digits $\frac{q}{2}$ and $-\frac{q}{2}$ depends on the next digit, we have to look ahead at the next digit in these cases. This algorithm is an extension of the one in [59] taking into account that we start with a larger digit set.

An example of standard addition for SSDEs with $q = 4$ using this algorithm is given in Table 5.4.

Algorithm 5.1 Standard addition for two (q, d)-expansions

Input: digit expansion $(s_\ell \dots s_0)_q$ with digits in $\{2d, \dots, 2q + 2d - 2\}$
Output: (q, d)-expansion z of $(s_\ell \dots s_0)_q$

```
z = ()
c = 0
for j = 0 to ℓ do
    a = s_j + c
    c = 0
    if a ≥ q + d then
        c = 1
    else if a ≤ d − 1 then
        c = −1
    end if
    a = a − cq
    z = (a) + z
end for
```

Algorithm 5.2 Standard addition for two SSDEs

Input: digit expansion $(s_\ell \dots s_0)_q$ with digits in $\{-q, \dots, q\}$
Output: SSDE z of $(s_\ell \dots s_0)_q$

```
s_{ℓ+1} = 0
z = ()
c = 0
for j = 0 to ℓ do
    a = s_j + c
    c = 0
    if a > q/2 then
        c = 1
    else if a < −q/2 then
        c = −1
    else if a = q/2 and (−q/2 ≤ s_{j+1} < 0 or q/2 ≤ s_{j+1} < q) then
        c = 1
    else if a = −q/2 and (−q < s_{j+1} ≤ −q/2 or 0 < s_{j+1} ≤ q/2) then
        c = −1
    end if
    a = a − cq
    z = (a) + z
end for
```

5.3.2. Transducers. In this section, we present the transducer for the algorithms presented in the last section.

We are not interested in the output of the addition, but only in the carries. Thus we only use the carries as the output of the transducer. But, if required, the output digits can easily be reconstructed.

In our setting, a transducer consists of a finite set of states S, a finite input alphabet $D+D$, an output alphabet, a set of transitions $E \subseteq S^2 \times (D+D)$ with input labels in $D+D$, output labels in the output alphabet for each transition, and an initial state. All states are final.

The input of the transducer is a digit expansion with digits in $D+D$. The output of the transducer is the sequence of labels of a path starting in the initial state with the given input as the input label. In our cases, there exists always such a path and it is unique (i.e., the transducer is complete and deterministic).

The labels of the states encode the current carry (except for the situations when we have to look ahead). The number of states is independent of q. The number of transitions between two states depends on the base q.

To plot the transducer, we group these transitions and their labels. We draw only one arc and write the label $M \mid c$ for a set $M \subset D+D$ to represent a group of transitions consisting of one transition with input label m and output label c for every $m \in M$. If M is the empty set, then there are no such transitions. This may happen for special values of d or q.

The output label of a transition is one carry c, a pair of carries c, or no carry c, i.e., $c \in \{0, 1, \bar{1}, -\} \cup \{0, 1, \bar{1}\}^2$, where "$-$" denotes the empty output. The input of the transducer is the sequence $\boldsymbol{s}$ of digitwise sums.

Let ℓ and u be the minimum and the maximum of the doubled digit set $D+D$. For the labels of the transitions, we define

$$M + \varepsilon = \big(\{m + \varepsilon \mid m \in M\} \cup (M \cap \{\ell, u\})\big) \cap (D+D)$$

for $\varepsilon = \pm 1$ and a set M. This definition is motivated by the following interpretation: Whenever a set $M = \{j, \dots, u\}$ occurs, it is actually meant to be the interval $[j, \infty)$ intersected with the doubled digit set. Subtracting 1 leads to $[j-1, \infty)$, again intersected with the doubled digit set. This corresponds to $M - 1$ as defined above.

5.3.2.1. *Standard Addition for (q,d)-expansions.* The transducer in Figure 5.1 computes the carries as in Algorithm 5.1. We use the sets $L = \{2d, \dots, d-1\}$, $D = \{d, \dots, q+d-1\}$ and $H = \{q+d, \dots, 2q+2d-2\}$.

The transitions are constructed by using Algorithm 5.1 for the current input and carry.

5.3.2.2. *Standard Addition for SSDEs.* The transducer in Figure 5.2 computes the carries as in Algorithm 5.2. We use the sets $L = \{0, \dots, \frac{q}{2} - 1\}$, $H = \{\frac{q}{2} + 1, \dots, q\}$ and $H_q = \{\frac{q}{2}, \dots, q-1\}$.

The transitions are constructed by using Algorithm 5.2 for the current input and carry.

The labels of the states -1, 0 and 1 encode the current carry. In the states with labels $\pm\frac{q}{2}$, we do not know yet whether the digit of the sum should be $\frac{q}{2}$ or $-\frac{q}{2}$ and thus, which carry is produced. To decide this, we have to look at the next digit. Thus, the transitions leading to a state $\pm\frac{q}{2}$ have no output ($-$) and the transitions starting at a state $\pm\frac{q}{2}$ have two output digits.

5.4. Approximate Equidistribution

As a probabilistic input model, we want to use an equidistribution on all digit expansions satisfying certain syntactical rules. This is easy in the case of (q,d)-expansions (see Section 5.4.1) because there are no syntactical rules. But in the case of a general regular language, like the SSDE, we can only approximate an equidistribution by Lemma 5.4.1. However, this approximation does not influence the main terms of the results.

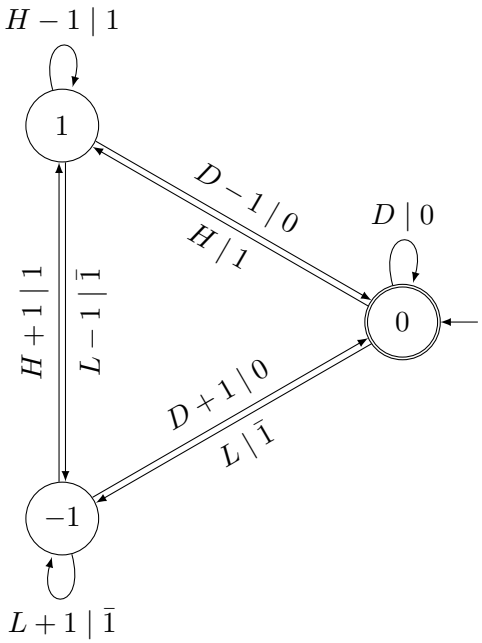


FIGURE 5.1. Standard addition for two (q, d)-expansions.

A regular language is recognized by an automaton. An automaton is defined to consist of states, transitions between these states with labels, an initial state and final states. So to say, it is a transducer without output. The automaton recognizes a word from a language, if there exists a path starting at the initial state, leading to a final state with this word as the label.

We call an automaton aperiodic if its underlying directed graph is aperiodic, i.e., the greatest common divisor of all lengths of directed cycles of the graph is 1. If the underlying directed graph is strongly connected, then the automaton is so, too. If an automaton is strongly connected and aperiodic, then the adjacency matrix of the underlying graph is primitive.

Given an automaton $\mathcal{A}$ for a regular language, we automatically construct transition probabilities between the states to obtain an approximate equidistribution on all words of given length ℓ. The weight of the word is the product of the transition probabilities multiplied with an *exit weight* (the factor in front of the product in (5.2) below). This corresponds to an approximate equidistribution on all paths of length ℓ of the underlying graph of the automaton starting in the initial state. Without the exit weights, these transition probabilities are the same as defined by Shannon in [93] and Parry in [83]. The computations of this lemma are submitted to be included as `Automaton.shannon_parry_markov_chain` in a future version of SageMath [96] (see Ticket #18089).

Lemma 5.4.1 ([46])**.** *Let $\mathcal{A}$ be a deterministic automaton with set of states $\{1, \ldots, n\}$, initial state 1, final states $\emptyset \neq F \subseteq \{1, \ldots, n\}$ recognizing a regular language $\mathcal{L}$. We assume that the adjacency matrix A of the underlying graph of $\mathcal{A}$ is primitive.*

The dominant eigenvalue of A is denoted by λ, all other eigenvalues of A are assumed to be of modulus less than or equal to $\xi\lambda$ for some $0 < \xi < 1$. If there are eigenvalues of modulus

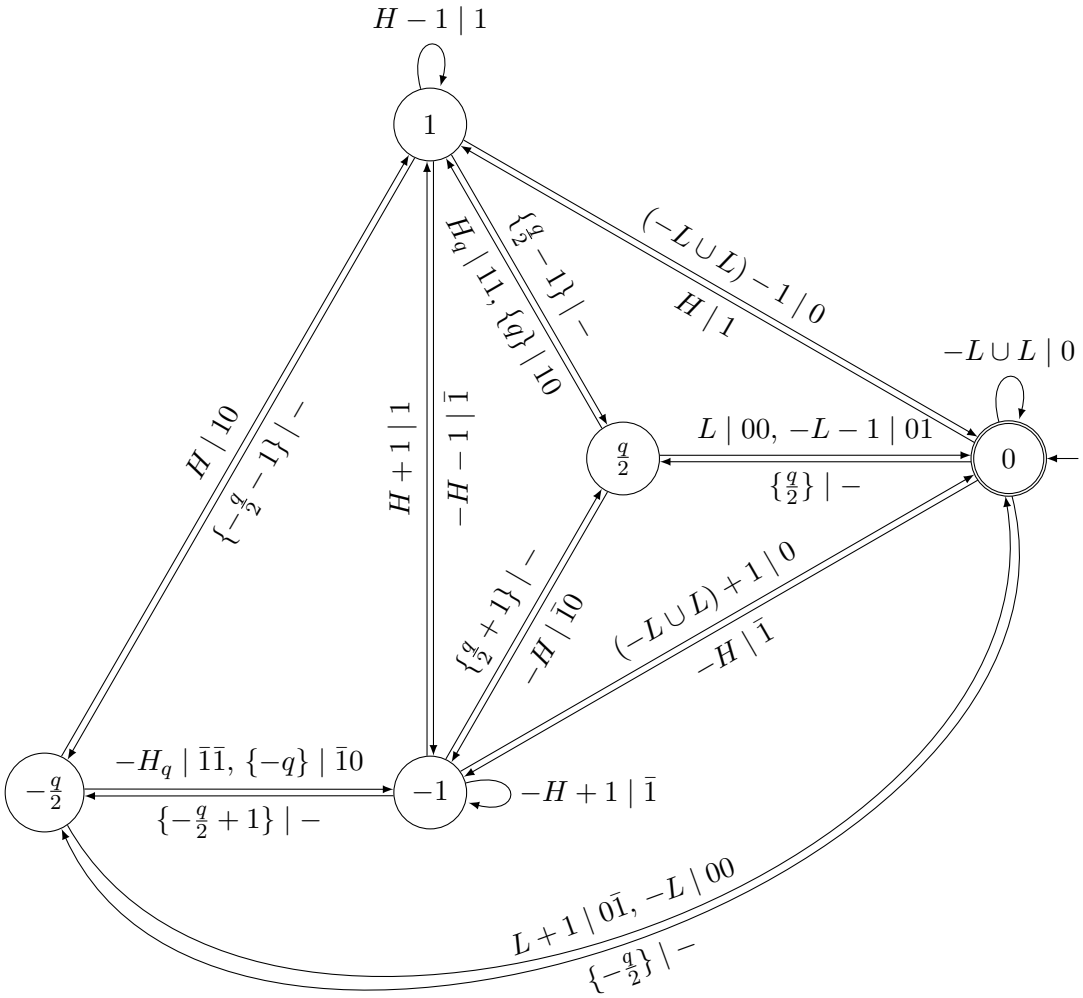


FIGURE 5.2. Standard addition for two SSDEs.

$\xi\lambda$, then each of them must be semisimple, i.e., its algebraic and geometric multiplicities coincide.

Let $w > 0$ and $u > 0$ be right and left eigenvectors of A to λ, respectively, such that $w_1 = 1$ and $\langle u, w\rangle = 1$.

For a transition t from some state i to some state j, we set

$$p_t = \frac{w_j}{w_i \lambda}. \tag{5.1}$$

For $\ell \geq 0$, the set of words of $\mathcal{L}$ of length ℓ is denoted by $\mathcal{L}_\ell$. For a word $x \in \mathcal{L}_\ell$, we denote the states and transitions used when $\mathcal{A}$ reads x by $1 = s_0, \ldots, s_\ell$ and $t_1, \ldots, t_\ell$, respectively. The weight $W_\ell(x)$ of x is then defined to be

$$W_\ell(x) = \frac{1}{w_{s_\ell} \langle u, e_F\rangle} \prod_{j=1}^{\ell} p_{t_j} \tag{5.2}$$

where e_F is the indicator vector of the set F of final states.

Then

$$\sum_{t \text{ leaves } i} p_t = 1 \tag{5.3}$$

holds for all states i and

$$W_\ell(x) = \frac{1}{|\mathcal{L}_\ell|}(1 + O(\xi^\ell)) \tag{5.4}$$

holds uniformly for $\ell \geq 0$ and $x \in \mathcal{L}_\ell$.

Furthermore, consider the time-homogeneous Markov chain $\mathcal{M}$ on the state space $\{1, \dots, n\}$ where the transition probability from state i to state j is $\sum_t p_t$ where the sum runs over all transitions in $\mathcal{A}$ from i to j. Then this Markov chain has the stationary distribution

$$(u_1 w_1, \dots, u_n w_n). \tag{5.5}$$

For large ℓ and a transition t from some state i to some state j, p_t can be thought as the probability of using t under the condition that the automaton is currently in state i. Note that the sum in (5.3) runs over all transitions leaving i such that multiple transitions between i and j are counted separately although their individual weights p_t only depend on i and j. It turns out that the exit weights do not influence the main term of our asymptotic expressions.

Proof of Lemma 5.4.1. We first note that the cardinality $|\mathcal{L}_\ell|$ is given by

$$|\mathcal{L}_\ell| = e_1^\top A^\ell e_F = \langle e_1, w\rangle\langle u, e_F\rangle\lambda^\ell(1 + O(\xi^\ell)) = \langle u, e_F\rangle\lambda^\ell(1 + O(\xi^\ell))$$

where $e_1 = (1, 0, \dots, 0)$.

For $x \in \mathcal{L}_\ell$ with associated sequence of states $(s_0, \dots, s_\ell)$, we have

$$W_\ell(x) = \frac{1}{w_{s_\ell}\langle u, e_F\rangle} \prod_{j=1}^{\ell} \frac{w_{s_j}}{w_{s_{j-1}}\lambda} = \frac{1}{w_{s_0}\langle u, e_F\rangle\lambda^\ell}.$$

As $w_{s_0} = w_1 = 1$, we get (5.4).

Next, we prove (5.3) by rewriting the sum as

$$\sum_{t \text{ leaves } i} p_t = \sum_{j=1}^{n} a_{ij}\frac{w_j}{w_i\lambda} = 1$$

by definition of w.

Finally, the transition matrix of the Markov chain $\mathcal{M}$ is

$$P = \Big(a_{ij}\frac{w_j}{w_i\lambda}\Big)_{1 \leq i,j \leq n}$$

by definition of the Markov chain and (5.1). Thus

$$P = \frac{1}{\lambda}\operatorname{diag}\Big(\frac{1}{w_1}, \dots, \frac{1}{w_n}\Big) A \operatorname{diag}(w_1, \dots, w_n).$$

As

$$\begin{aligned}(u_1 w_1, \dots, u_n w_n)P &= \frac{1}{\lambda}(u_1, \dots, u_n)A \operatorname{diag}(w_1, \dots, w_n) \\ &= (u_1, \dots, u_n)\operatorname{diag}(w_1, \dots, w_n) \\ &= (u_1 w_1, \dots, u_n w_n),\end{aligned}$$

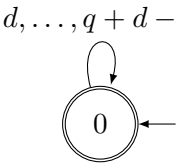


FIGURE 5.3. Automaton recognizing (q, d)-expansions.

$(u_1 w_1, \dots u_n w_n)$ is a left eigenvector of P to the eigenvalue 1. By definition of u and w, $\sum_{i=1}^{n} u_i w_i = 1$. As $\mathcal{M}$ is aperiodic and irreducible, $(u_1 w_1, \dots, u_n w_n)$ is the unique left eigenvector with this property and therefore the stationary distribution. □

The weight W_ℓ induces a probability distribution on the words of length ℓ up to an exponentially small error. Each word has approximately the same weight. If we see the transition probabilities as a part of the automaton, we obtain a probabilistic automaton:

Definition 5.4.2. A probabilistic automaton is an automaton together with a map $p\colon t \mapsto p_t$ from the set of transitions to the interval $[0, 1]$ such that

$$\sum_{t \text{ leaves } s} p_t = 1$$

holds for all states s. We call p_t the weight or the probability of the transition t.

5.4.1. Weights for (q, d)-expansions. We can use Lemma 5.4.1 in this case, too, but the digits of a (q, d)-expansion are independent of each other because there are no syntactical rules involving more than one digit. Therefore we can directly obtain equidistribution, not only approximating it. We first describe the direct way and later, in Remark 5.4.3, we consider using Lemma 5.4.1.

For any digit $x_0 \in D$, we use the weights $W_\ell(x_0) = \frac{1}{q}$. The exit weight is 1. By independence, we have the weight

$$W_\ell(x) = \frac{1}{q^\ell}$$

for a digit expansion x of length ℓ. With this weight, we have an equidistribution of all (q, d)-expansions of length ℓ.

Remark 5.4.3. The same weights can be obtained by Lemma 5.4.1. The transition probabilities are $p_{0\to 0} = q^{-1}$. As the automaton recognizing (q, d)-expansions has only one state (see Figure 5.3), there is no error term in (5.4).

5.4.2. Weights for SSDEs. The automaton in Figure 5.4 recognizes SSDEs. The adjacency matrix of this automaton is

$$A = \begin{pmatrix} 0 & \frac{q}{2} & 0 \\ 1 & q-1 & 1 \\ 0 & \frac{q}{2} & 0 \end{pmatrix}$$

where the states are ordered by their labels.

The matrix A has the eigenvalues q, -1 and 0. The vectors $(\frac{1}{q+1}, \frac{q}{q+1}, \frac{1}{q+1})$ and $(\frac{1}{2}, 1, \frac{1}{2})^\top$ are the left and right eigenvector corresponding to the eigenvalue q, respectively. The transition probabilities are

$$p_{-1\to 0} = p_{1\to 0} = \frac{2}{q}, \quad p_{0\to 1} = p_{0\to -1} = \frac{1}{2q}, \quad p_{0\to 0} = \frac{1}{q}. \tag{5.6}$$

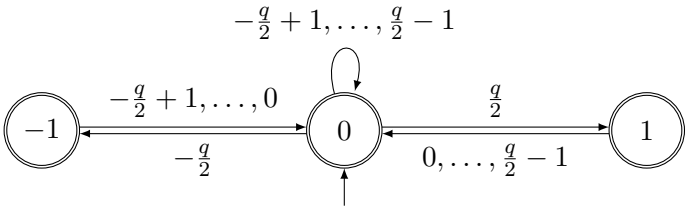


FIGURE 5.4. Automaton recognizing SSDEs.

The constant in the error term is $\xi = \frac{1}{q}$. The exit weights are $(2,1,2)\cdot\frac{q+1}{q+2}$.

With these transition probabilities, the asymptotic frequencies of the digits (cf. [58]) can be computed as

$$\begin{cases} \frac{1}{2(q+1)} & \text{if the digit is } \pm\frac{q}{2}, \\ \frac{q+2}{q(q+1)} & \text{if the digit is } 0, \\ \frac{1}{q} & \text{otherwise} \end{cases} \tag{5.7}$$

by a steady state analysis of the related Markov chain using (5.5).

5.5. Asymptotic Analysis of the Standard Addition

In this section, we use the probabilistic model defined in Section 5.4 for the input sequence of the transducers in Section 5.3.2. Then we will use Lemma 5.5.1 to obtain expectation, variance and asymptotic normality of the number of carries.

In Sections 5.5.1 and 5.5.2, we will construct probabilistic automata whose transition labels are the carries and where each transition has a weight corresponding to the weight constructed in Section 5.4.

Let m and n be two functions mapping the output of a transition into the real numbers; for brevity we write $m(t)$ and $n(t)$ without mentioning the output label of the transition t. In our setting m will count the number of carries 1, and n the number of carries -1 of the output of a transition. We consider the two random variables M_ℓ and N_ℓ which are the sum of the values of m and n, respectively, over a path of length ℓ with probability the product of the weights of this path multiplied with the exit weight.

The transition matrix $A(x,y)$ of a probabilistic automaton with K states and two functions m and n is a $K\times K$ matrix whose (i,j)-th entry is

$$\sum_{t:\ i\to j} p_t x^{m(t)} y^{n(t)}$$

where p_t is the weight of the transition t.

The next lemma is a slight modification of [56, Theorem 3.9] taking into account the non-uniform distribution of the input alphabet.

Lemma 5.5.1. *Let $\mathcal{A}$ be a strongly connected, aperiodic probabilistic automaton where all states are final. Let m and n be functions mapping the output of a transition into the real numbers and $A(x,y)$ be the associated transition matrix of the automaton, where x and y mark m and n, respectively. Let M_ℓ and N_ℓ be the associated random variables as defined above.*

Define the function $f(x,y,z) = \det(I - zA(x,y))$. *Then the expected value of* (M_ℓ, N_ℓ) *is* $(e_m, e_n)\ell + \mathcal{O}(1)$ *with*

$$e_m = \left.\frac{f_x}{f_z}\right|_{(1,1,1)},$$
$$e_n = \left.\frac{f_y}{f_z}\right|_{(1,1,1)}.$$

The variance-covariance matrix $\left(\begin{smallmatrix} v_m & c \\ c & v_n \end{smallmatrix}\right)\ell + \mathcal{O}(1)$ *has the entries*

$$v_m = \left.\frac{f_x^2(f_{zz}+f_z) + f_z^2(f_{xx}+f_x) - 2f_xf_zf_{xz}}{f_z^3}\right|_{(1,1,1)}, \tag{5.8}$$

$$v_n = \left.\frac{f_y^2(f_{zz}+f_z) + f_z^2(f_{yy}+f_y) - 2f_yf_zf_{yz}}{f_z^3}\right|_{(1,1,1)}, \tag{5.9}$$

$$c = \left.\frac{f_xf_y(f_{zz}+f_z) + f_z^2f_{xy} - f_yf_zf_{xz} - f_xf_zf_{yz}}{f_z^3}\right|_{(1,1,1)}. \tag{5.10}$$

Furthermore, if v_m *and* v_n *are non-zero, then* M_ℓ *and* N_ℓ *are asymptotically normally distributed, respectively. If the variance-covariance matrix is non-singular, then* M_ℓ *and* N_ℓ *are asymptotically jointly normally distributed.*

Proof. The moment generating function is

$$\mathbb{E}\exp(s_1M_\ell + s_2N_\ell) = [z^\ell]e_1^\top(I - zA(e^{s_1}, e^{s_2}))^{-1}w_F$$

where e_1 is a unit vector with a 1 at the position of the initial state and the entries of w_F are the exit weights of the states. Since the automaton is probabilistic and aperiodic, the unique dominant eigenvalue of $A(1,1)$ is 1. Thus the same arguments apply as in [56] (or Chapter 3) after replacing "complete" by "probabilistic". We obtain the same formulas for the constants of the expectation, the variance and the covariance. Also the central limit theorem follows. □

5.5.1. Standard Addition for (q,d)-expansions. To construct the probabilistic automaton, we start with the transducer in Figure 5.1, and use the weights from Section 5.4.1.

All steps in this section, including the computation of the constants in Theorem 5.1, can be done in the mathematical software system SageMath [96] by using the included finite state machine package described in [49] and Chapter 6. A notebook with the used code can be found at [54].

The construction in this section is more general than needed for the case of independent digits as in (q,d)-expansions. But discussing it here in full generality allows reusing the same ideas for the case of dependent digits as in SSDEs later on. We will use the same construction for SSDEs in Sections 5.5.2 and 5.7.

In this section, let $\mathcal{A}$ be the automaton in Figure 5.3, equipped with the weight $\frac{1}{q}$ for every transition and the exit weight 1 for every state (by Section 5.4.1). Construct $\mathcal{A}^2$ as the *additive Cartesian product*[1] of $\mathcal{A}$ with itself by the following rules:

[1]This can also be seen as the composition of a transducer performing digitwise addition (without considering any carries) and the Cartesian product of $\mathcal{A}$ with itself. This corresponds to the SageMath methods `transducers.add` and `Transducer.cartesian_product`, respectively. The composition can be computed by the SageMath method `Transducer.composition`.

- The states of $\mathcal{A}^2$ are pairs of states of $\mathcal{A}$.
- There is a transition from (a, b) to (c, d) with label $x+y$ in $\mathcal{A}^2$ if there are transitions from a to c with label x and b to d with label y in $\mathcal{A}$.
- The weight of a transition in $\mathcal{A}^2$ is the product of the weights of the two transitions in $\mathcal{A}$.
- The exit weight of a state in $\mathcal{A}^2$ is the product of the exit weights of the two states in $\mathcal{A}$.

The probabilistic automaton $\mathcal{A}^2$ recognizes all possible sequences $\boldsymbol{s}$ of digitwise sums with the correct weights for the equidistribution on the independent (q, d)-expansions $\boldsymbol{x}$ and $\boldsymbol{y}$.

In this section, let $\mathcal{B}$ be the transducer in Figure 5.1 performing the standard addition of two (q, d)-expansions. Next, we construct $\mathcal{S}_{(q,d)}$ as the composition $\mathcal{B} \circ \mathcal{A}^2$ by the following rules:

- The states of $\mathcal{S}_{(q,d)}$ are pairs of states of $\mathcal{B}$ and $\mathcal{A}^2$.
- For each pair of transitions from a to c with input label s and output label k in $\mathcal{B}$ and from b to d with weight w and label s in $\mathcal{A}^2$, there is a transition from (a, b) to (c, d) with weight w and label k in $\mathcal{S}_{(q,d)}$.
- The exit weight of a state in $\mathcal{S}_{(q,d)}$ is the exit weight of the corresponding state in $\mathcal{A}^2$.

The probabilistic automaton $\mathcal{S}_{(q,d)}$ recognizes the sequence of carries $\boldsymbol{c}$ with the correct weights for the equidistribution on the independent (q, d)-expansions $\boldsymbol{x}$ and $\boldsymbol{y}$. The probabilistic automaton $\mathcal{S}_{(q,d)}$ has three states.

To determine the transition matrix of $\mathcal{S}_{(q,d)}$, we use the following lemma to compute the number of transitions between two states. The lemma is proved by an inclusion-exclusion argument.

Lemma 5.5.2 ([46])**.** *Let*

$$N(x_{\min}, x_{\max}, y_{\min}, y_{\max}, s_{\min}, s_{\max}) = \\ |\{(x, y) \in \mathbb{Z}^2 \mid x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, s_{\min} \le x + y \le s_{\max}\}|.$$

Then we have

$$\begin{aligned} N(x_{\min}, x_{\max}, y_{\min}, y_{\max}, s_{\min}, s_{\max}) = {} & N(0, \infty, 0, \infty, 0, s_{\max} - x_{\min} - y_{\min}) \\ & - N(0, \infty, 0, \infty, 0, s_{\max} - x_{\min} - y_{\max} - 1) \\ & - N(0, \infty, 0, \infty, 0, s_{\max} - x_{\max} - y_{\min} - 1) \\ & + N(0, \infty, 0, \infty, 0, s_{\max} - x_{\max} - y_{\max} - 2) \\ & - N(0, \infty, 0, \infty, 0, s_{\min} - x_{\min} - y_{\min} - 1) \\ & + N(0, \infty, 0, \infty, 0, s_{\min} - x_{\min} - y_{\max} - 2) \\ & + N(0, \infty, 0, \infty, 0, s_{\min} - x_{\max} - y_{\min} - 2) \\ & - N(0, \infty, 0, \infty, 0, s_{\min} - x_{\max} - y_{\max} - 3) \end{aligned}$$

with $N(0, \infty, 0, \infty, 0, s_{\max}) = 0$ *if* $s_{\max}$ *is negative and*

$$N(0, \infty, 0, \infty, 0, s_{\max}) = \frac{1}{2}(s_{\max} + 2)(s_{\max} + 1)$$

otherwise.

This gives the transition matrix in Table A.1 in the appendix where x marks carries 1 and y marks carries -1. For example, the entry in the first row and column is

$$\frac{(d-1)(d-2)}{2q^2}y = \sum_{\substack{x,y\in D\\ x+y\in L+1}} p_{0\to 0}p_{0\to 0}y = \frac{1}{q^2}N(d, q+d-1, d, q+d-1, 2d, d)y$$

because this entry corresponds to the transitions from -1 to -1 with input label $L+1$ and output label $\bar{1}$ in $\mathcal{B}$ and from $(0,0)$ to $(0,0)$ in $\mathcal{A}^2$.

With the transition matrix, the next theorem follows directly from Lemma 5.5.1.

Theorem 5.1. *Let M_ℓ and N_ℓ be the number of carries 1 and -1, respectively, when adding two independent random (q,d)-expansions of length ℓ. The expected value of (M_ℓ, N_ℓ) is $(e_1, e_{-1})\ell + \mathcal{O}(1)$ with constants*

$$e_1 = \frac{(q+d-1)^2}{2(q-1)^2},$$
$$e_{-1} = \frac{d^2}{2(q-1)^2}.$$

The variance-covariance matrix of (M_ℓ, N_ℓ) is $\left(\begin{smallmatrix} v_1 & c \\ c & v_{-1}\end{smallmatrix}\right)\ell + \mathcal{O}(1)$ with constants

$$v_1 = \frac{(q+d-1)^2(q^4-2q^3d-q^2d^2-4qd^2-2q^2-d^2+2d+1)}{4(q-1)^5(q+1)},$$
$$v_{-1} = \frac{d^2(2q^4-q^2d^2-4q^3-6q^2d-4qd^2+4q^2+6qd-d^2-4q+2)}{4(q-1)^5(q+1)},$$
$$c = \frac{d(q+d-1)(q^3d+q^2d^2-q^3+3q^2d+4qd^2+2q^2-3qd+d^2-q-d)}{4(q-1)^5(q+1)}.$$

Furthermore, the number of carries 1 and -1 is asymptotically jointly normally distributed for $d \neq 0, -q+1$. For $d=0$, M_ℓ is asymptotically normally distributed and $N_\ell = 0$ because the carry -1 does not occur. For $d=-q+1$, the same holds with M_ℓ and N_ℓ exchanged.

Remark 5.5.3. The expected value for carries in the addition of (q,d)-expansions corresponds to the result in [79]. There, the authors find the stationary distribution

$$\frac{1}{2(q-1)^2}(d^2,\ q^2-2q+1-2qd+2d-2d^2,\ (q+d-1)^2)$$

for the states $(-1,0,1)$ of the carry process. For $d=\frac{-q+1}{2}$, this stationary distribution can also be found in [22].

5.5.2. Standard Addition for SSDEs. To cope with the dependencies between the digits, we have to combine the conditional probabilities of the automaton in Figure 5.4 with the carries computed by the automaton in Figure 5.2. This is done in the same way as in Section 5.5.1.

All steps in this section, including the computation of the constants in Theorem 5.2, can be done in the mathematical software system SageMath [96] by using its included finite state machine package described in [49] and Chapter 6. A notebook with the used code can be found at [54].

In this section, let $\mathcal{A}$ be the automaton in Figure 5.4 equipped with the weights in (5.6) and let $\mathcal{B}$ be the transducer in Figure 5.2 performing the standard addition of two SSDEs.

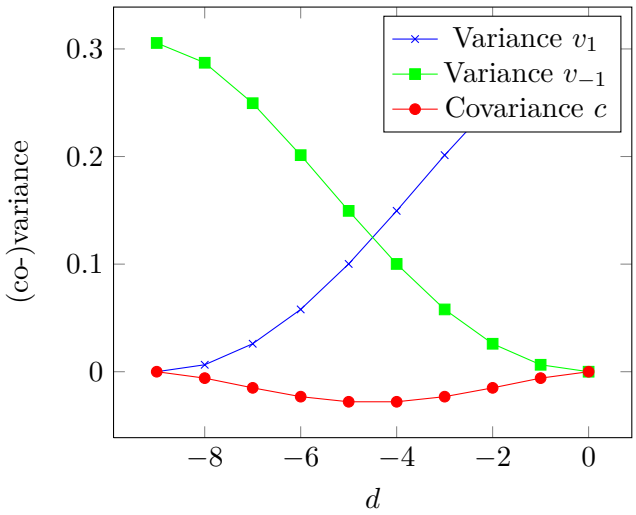


FIGURE 5.5. Variances and covariance for $(10, d)$-expansions of Theorem 5.1.

We first construct the additive Cartesian product $\mathcal{A}^2$, recognizing all possible sequences $\boldsymbol{s}$ of digitwise sums with the correct weights approximating the equidistribution on two independent SSDEs $\boldsymbol{x}$ and $\boldsymbol{y}$. This probabilistic automaton has 9 states.

Next, we construct $\mathcal{S}_{\mathrm{SSDE}}$ as the composition $\mathcal{B} \circ \mathcal{A}^2$. This probabilistic automaton recognizes the sequence of carries $\boldsymbol{c}$ with the correct weights approximating the equidistribution on two independent SSDEs $\boldsymbol{x}$ and $\boldsymbol{y}$. We want to asymptotically analyze the number of carries equal to 1. This gives a transducer with 45 states.

Because of symmetries (cf. Remark 5.3.1), we can simplify $\mathcal{S}_{\mathrm{SSDE}}$ such that it has only 14 states[2]:

Lemma 5.5.4. *A probabilistic automaton can be simplified by applying the following rules:*

- *If between two states, there are two transitions with the same label, then these two transitions can be combined. The weights are summed up in this process.*
- *Let $\{C_1, \ldots, C_k\}$ be a partition of the states of the transducer with the following property: If a, $b \in C_j$ are two states, then there is a bijection between the transitions leaving a and the ones leaving b which preserves the label, the weight of the transition and into which set of the partition the transitions lead. These bijections define an equivalence relation on the transitions leaving a set of the partition.*

  Then each set of the partition can be contracted to a new state. For each equivalence class of transitions, there is one transition in the simplified transducer.

Thus, we obtain a 14×14 transition matrix of $\mathcal{S}_{\mathrm{SSDE}}$ given in Table A.2 in the appendix (using Lemma 5.5.2).

Theorem 5.2. *The expected value of the number of carries equal to* 1 *when adding two SSDEs of length ℓ is*

$$\frac{q^2 + 2q + 4}{8(q+1)^2}\ell + \mathcal{O}(1)$$

[2]For the actual computation, it is more efficient to already simplify $\mathcal{A}^2$ by the SageMath method `FiniteStateMachine.markov_chain_simplification`, such that it only has 6 states.

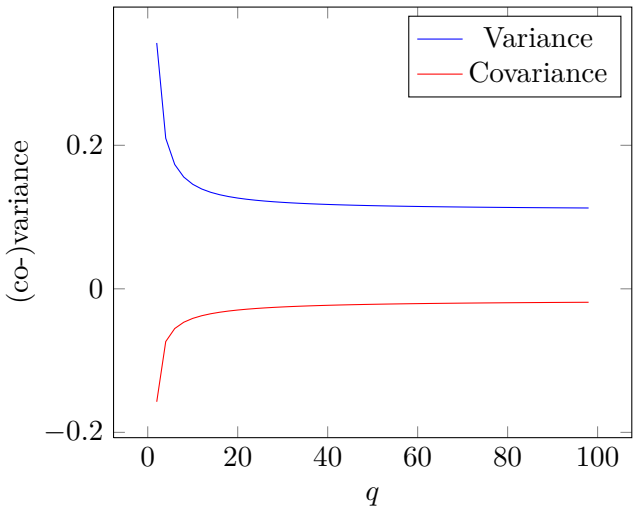


FIGURE 5.6. Variance and covariance for SSDEs for $q = 2, \dots, 100$ of Theorem 5.2.

and the variance is

$$\frac{7q^6 + 48q^5 + 159q^4 + 128q^3 - 48q^2 - 12q - 8}{64(q+1)^5(q-1)}\ell + \mathcal{O}(1).$$

The same result holds for carries equal to -1*. The covariance between carries* 1 *and* -1 *is*

$$-\frac{q^6 + 24q^5 + 33q^4 + 80q^3 + 120q^2 - 12q - 8}{64(q+1)^5(q-1)}\ell + \mathcal{O}(1).$$

The number of carries 1 *and* -1 *is asymptotically jointly normally distributed.*

PROOF. We can compute the determinant $f(x, y, z) = \det(I - zA(x, y))$ of the transition matrix $A(x, y)$ in the appendix of the simplified automaton $\mathcal{S}_{\text{SSDE}}$ with 14 states. Thus, Lemma 5.5.1 implies the expected value, the variance and the central limit theorem where the input sequence is the sum of two independent SSDEs of length ℓ with the approximate equidistribution W_ℓ.

As the (exact) equidistribution $\mathbb{P}_\ell$ satisfies $\mathbb{P}_\ell = (1 + \mathcal{O}(\xi^\ell))W_\ell$, these results also hold for the (exact) equidistribution. □

Remark 5.5.5. If we neglect the dependencies between two adjacent digits, we obtain a different result: Assume that the digits are independently distributed with probabilities given in (5.7). Then the expected value of the number of carries 1 is

$$\frac{q^{12} + 7q^{11} + 19q^{10} + 27q^9 + 24q^8 + 9q^7 - 15q^6 - 15q^5 + 47q^4 + 104q^3 + 64q^2 - 48q - 48}{8(q+1)^3q^2(q^7 + 4q^6 + 5q^5 - q^4 - 9q^3 - 8q^2 + 4)}$$

and the variance is

$$\begin{aligned}\frac{1}{64}(&7q^{38} + 152q^{37} + 1557q^{36} + 9958q^{35} + 44300q^{34} + 144166q^{33} + 349511q^{32}\\ &+ 622942q^{31} + 756995q^{30} + 432788q^{29} - 439628q^{28} - 1347486q^{27} - 1407649q^{26}\\ &- 466340q^{25} - 39181q^{24} - 2293904q^{23} - 6902413q^{22} - 9055044q^{21} - 2972395q^{20}\end{aligned}$$

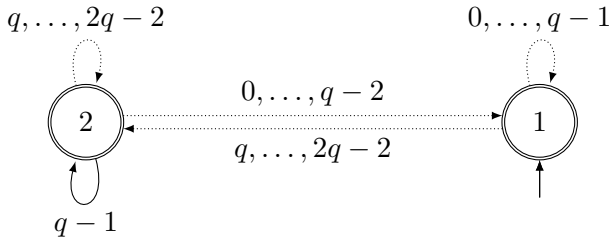


FIGURE 5.7. Automaton to find the longest carry generating sequence for von Neumann's addition of two standard q-ary expansions.

$$
\begin{aligned}
&+ 10157788q^{19} + 19040707q^{18} + 12034998q^{17} - 7655356q^{16} - 21471482q^{15} \\
&- 15688011q^{14} + 1495584q^{13} + 10611092q^{12} + 5762536q^{11} - 1482784q^{10} \\
&- 1794016q^9 + 1000784q^8 + 744768q^7 - 1199872q^6 - 1204224q^5 + 120832q^4 \\
&+ 574464q^3 + 172032q^2 - 73728q - 36864) \\
&\times q^{-4}(q+1)^{-6}(q^7 + 4q^6 + 5q^5 - q^4 - 9q^3 - 8q^2 + 4)^{-3} \\
&\times (q^7 + 2q^6 + q^5 + q^4 + q^3 - 2q^2 + 4)^{-1}.
\end{aligned}
$$

As expected, the limit for q to infinity is the same.

5.6. Von Neumann's Addition

In this section, we analyze von Neumann's addition algorithm for SSDEs, a parallel algorithm using several iterations. This algorithm was analyzed by Knuth in [72] for standard q-ary expansions. In [59], this analysis was extended to (q, d)-expansions and SSDEs. However, for $q \geq 4$, the hardware and software available at that time made the use of the probabilistic model of Section 5.4.2 computationally infeasible. The approximate model described in Remark 5.5.5 was used instead. As Remark 5.5.5 demonstrates, this approximation may lead to other main terms in the expectation and the variance.

As before, we choose an approximate equidistribution for all independent pairs of SSDEs of length ℓ as our probabilistic input model. In contrast to the result in [59], we obtain more natural constants occurring in the main term of the expectation and the variance.

For von Neumann's addition of two standard q-ary digit expansions, the number of iterations depends on the longest subsequence $(q-1)\dots(q-1)j$ with $j \geq q$ of the digitwise sum $\boldsymbol{s}$, see [72]. Such sequences can be found by an automaton with two classes of transitions (see Figure 5.7 and [59, Figure 1]). One class corresponds to the digit $(q-1)$ of a carry generating sequence and is depicted by solid lines. The other class corresponds to all other digits (including the digit j of a carry generating sequence) and is depicted by dotted lines. The longest consecutive run of solid edges in the automaton in Figure 5.7 corresponds to the number of iterations of von Neumann's addition minus 2. The asymptotic analysis of these longest runs can be performed using the probabilistic version of the automaton in Figure 5.7. We will extend this approach to SSDEs with arbitrary even base using a larger probabilistic automaton in Section 5.7.

5.6.1. Algorithm. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two SSDEs. The idea of the algorithm is to construct the sequence of digitwise sums $\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{y}$ and correct each position if the number at this

$$\begin{array}{rrr}
(110\bar{1}\bar{2})_4 = \boldsymbol{x} = \boldsymbol{z}^{(0)} = & 314 \\
(101\bar{1}\bar{2})_4 = \boldsymbol{y} = \boldsymbol{c}^{(0)} = & 266 \\
\hline
(21120)_4 = \quad \boldsymbol{z}^{(1)} = & 600 \\
(000\bar{1}\bar{1}0)_4 = \quad \boldsymbol{c}^{(1)} = & -20 \\
\hline
(021010)_4 = \quad \boldsymbol{z}^{(2)} = & 580 \\
(0000000)_4 = \quad \boldsymbol{c}^{(2)} = & 0
\end{array}$$

TABLE 5.5. Example for von Neumann's addition for SSDEs with $q = 4$. We have $t(110\bar{1}\bar{2}, 101\bar{1}\bar{2}) = 2$.

position is not in the digit set or at the border of the digit set where we have to take into account the syntactical rule.

As in [59], we define $(\boldsymbol{z}, \boldsymbol{c}) = \mathrm{add}(\boldsymbol{s})$ with $\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{y}$ by

$$\begin{aligned}
c_0 &= 0, \\
c_{j+1} &= \begin{cases} \operatorname{sgn}(s_j) & \text{if } |s_j| > \frac{q}{2}, \\ & \quad \text{or } |s_j| = \frac{q}{2} \text{ and} \\ & \quad (\operatorname{sgn}(s_j) s_{j+1}) \bmod q \geq \frac{q}{2} \\ 0 & \text{otherwise,} \end{cases} \\
z_j &= s_j - c_{j+1} q.
\end{aligned}$$

Here, the choice of the carry c_{j+1} corresponds to the one in Algorithm 5.2. By iterating this step we obtain $(\boldsymbol{z}^{(k+1)}, \boldsymbol{c}^{(k+1)}) = \mathrm{add}(\boldsymbol{z}^{(k)} + \boldsymbol{c}^{(k)})$ with $\boldsymbol{z}^{(0)} = \boldsymbol{x}$ and $\boldsymbol{c}^{(0)} = \boldsymbol{y}$. If $\boldsymbol{c}^{(k)} = 0$, then $\boldsymbol{z}^{(k)}$ is the SSDE of the sum $\boldsymbol{x} + \boldsymbol{y}$ and the algorithm stops. Note that during this process, $\boldsymbol{z}^{(k)}$ and $\boldsymbol{c}^{(k)}$ are not necessarily SSDEs.

In [59], the correctness and the termination of this algorithm were proved. We denote the number of iterations of von Neumann's addition algorithm by $t(\boldsymbol{x}, \boldsymbol{y}) = \min\{k \geq 0 : \boldsymbol{c}^{(k)} = 0\}$.

5.6.2. Automaton. A description of all SSDEs $\boldsymbol{x}$ and $\boldsymbol{y}$ with $t(\boldsymbol{x}, \boldsymbol{y}) = k$ is given in [59]. This description is in terms of an automaton and leads to the automaton in [59, Figure 5] reproduced here as Figure 5.8. We use the sets $L = \{0, \ldots, q/2 - 1\}$, $L_0 = L \setminus \{0\}$, $H = \{q/2 + 1, \ldots, q\}$ and $H_q = H \setminus \{q\}$.

From [59, Theorem 3.4], we know that $t(\boldsymbol{x}, \boldsymbol{y}) \leq k + 2$ if and only if this automaton traverses at most k consecutive solid transitions when reading $(s_j)_{j \geq 0}$.

Remark 5.6.1. Strictly speaking, the automaton reads the sequence $(s_j)_{j \geq 0}$ where $s_j = x_j + y_j$ for $j \leq J$ and $s_j = 0$ for $j > J$, for some J. However, most of the solid edges are visited while $j \leq J$. All transitions with label 0 lead to state 1. Those from states 2 and 7 are solid edges, all others are dotted. If the transition is in state 2 (or 7) after reading s_J, an additional solid edge will be traversed. Thus, we have to specially treat the states 2 and 7.

5.7. Asymptotic Analysis of von Neumann's Addition

For the asymptotic analysis, we combine the automaton in Figure 5.8 with the probabilistic model for SSDEs from Section 5.4.2 in the same way as in Section 5.5.2.

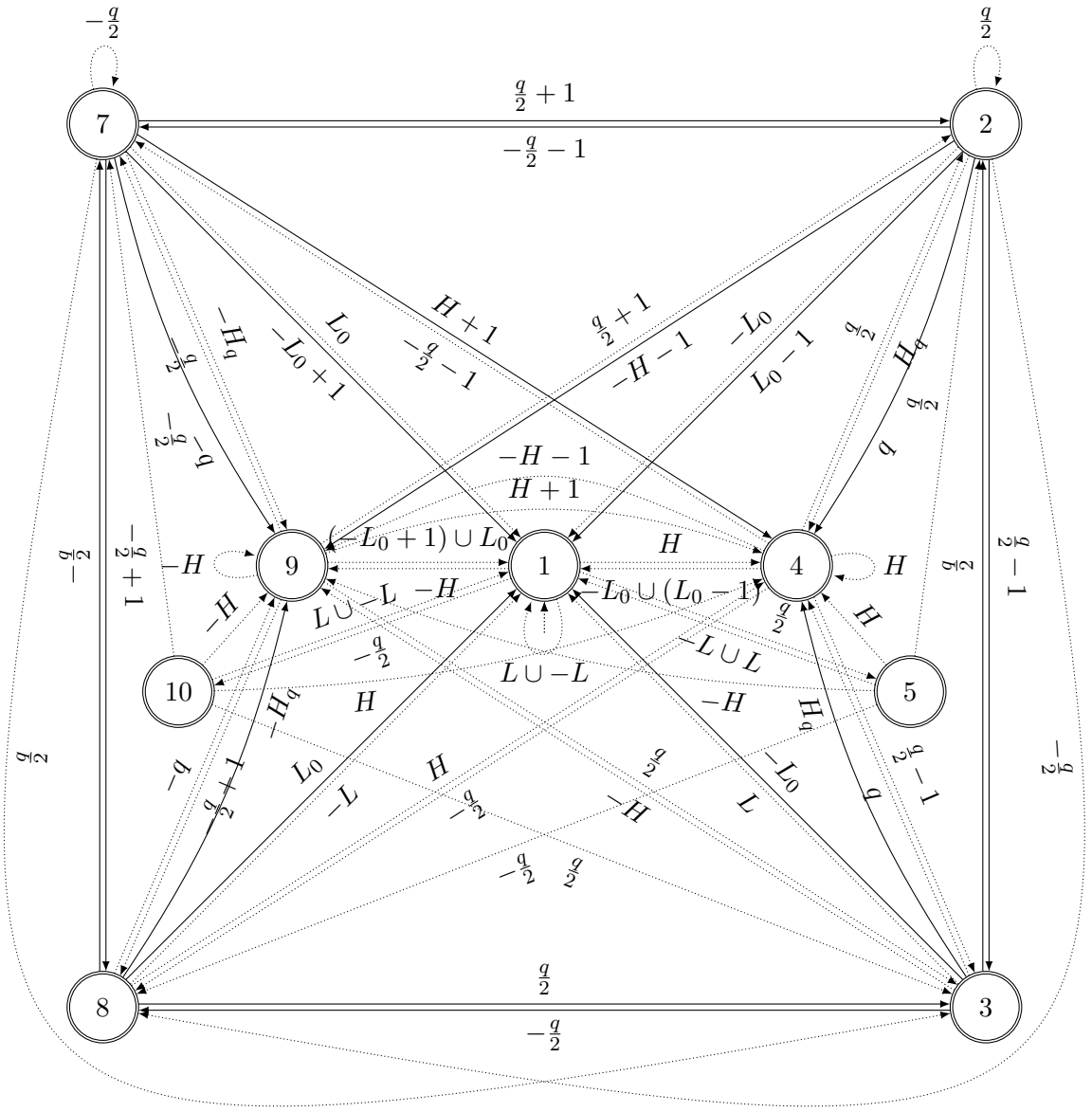


FIGURE 5.8. Automaton in [59, Figure 5]: $t(\boldsymbol{x},\boldsymbol{y}) \le k+2$ if and only if the automaton traverses at most k solid edges when reading $(s_j)_{j\ge 0}$.

All steps in this section, including the computation of the constants in Theorem 5.4, can be done in the mathematical software system SageMath [96] by using the included finite state machine package described in [49] and Chapter 6. A notebook with the used code can be found at [54].

We again use the automata $\mathcal{A}$ and $\mathcal{A}^2$ described in Section 5.5.2, recognizing SSDEs and the digitwise sum of two SSDEs, respectively. As before, the next step is to construct the Cartesian product $\mathcal{N}_{\mathrm{SSDE}}$ of the automaton $\mathcal{B}$ in Figure 5.8 and $\mathcal{A}^2$.

After simplifying this construction as described in Lemma 5.5.4, the probabilistic automaton $\mathcal{N}_{\mathrm{SSDE}}$ has 12 states:

$$
\begin{aligned}
&\{(1,(-1,1)),(1,(1,-1))\},\\
&\{(4,(0,0)),(9,(0,0))\},\\
&\{(5,(0,1)),(5,(1,0)),(10,(-1,0)),(10,(0,-1))\},\\
&\{(2,(0,1)),(2,(1,0)),(7,(-1,0)),(7,(0,-1))\},\\
&\{(5,(0,0)),(10,(0,0))\},\\
&\{(2,(0,0)),(7,(0,0))\},\\
&\{(1,(-1,0)),(1,(0,-1)),(1,(0,1)),(1,(1,0))\},\\
&\{(3,(0,1)),(3,(1,0)),(8,(-1,0)),(8,(0,-1))\},\\
&\{(1,(0,0))\},\\
&\{(3,(0,0)),(8,(0,0))\},\\
&\{(4,(1,1)),(9,(-1,-1))\},\\
&\{(4,(0,1)),(4,(1,0)),(9,(-1,0)),(9,(0,-1))\}.
\end{aligned}
\tag{5.11}
$$

In this case, the simplification is done in the same way as in Lemma 5.5.4, but also taking into account the class (dotted or solid) of a transition. The partition of the set of states was constructed by the symmetries between the two sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ described in Remark 5.3.1, for example $\{(1,(-1,1)),(1,(1,-1))\}$, and the additional vertical symmetry of the automaton in Figure 5.8, for example $\{(4,1,1),(9,-1,-1)\}$.

The state $(1,(0,0))$ is initial and all states are final.

The next theorem is an extension of Lemma 2.5 in [59] additionally including the variance and convergence in distribution.

Theorem 5.3. *Let $w_{\ell k}$, ℓ, $k \geq 0$, be non-negative numbers with generating function*

$$G_k(z) = \frac{R_k(z)}{S_k(z)} = \sum_{\ell \geq 0} w_{\ell k} z^\ell$$

such that $w_{\ell,k}$ is non-decreasing in k.

Assume that

$$
\begin{aligned}
R_k(z) &= r_0(z) + r_1\Big(z, \Big(\frac{z}{a_1}\Big)^k, \dots, \Big(\frac{z}{a_m}\Big)^k\Big),\\
S_k(z) &= (1-z)s_0(z) + \Big(\frac{z}{a_1}\Big)^k s_1(z) + s_2\Big(z, \Big(\frac{z}{a_1}\Big)^k, \dots, \Big(\frac{z}{a_m}\Big)^k\Big),
\end{aligned}
$$

where r_0, s_0, and s_1 are real polynomials in z (not depending on k). Furthermore, r_1 and s_2 are real polynomials in z, $(z/a_1)^k$, …, $(z/a_m)^k$ for some $m \geq 2$ and some real numbers $1 < a := a_1 < |a_2| \leq |a_3| \leq \cdots \leq |a_m|$ such that each of the summands in r_1 is divisible by one of the terms $(z/a_1)^k$, …, $(z/a_m)^k$ and each of the summands in s_2 is divisible by one of the terms $(z/a_1)^{2k}$, $(z/a_2)^k$, …, $(z/a_m)^k$. Assume furthermore that $r_0(1) \neq 0$.

Then $G(z) := \frac{r_0(z)}{(1-z)s_0(z)} = \lim_{k\to\infty} G_k(z)$ and

$$G(z) = \sum_{\ell \geq 0} w_\ell z^\ell$$

with $w_\ell = w_{\ell k}$ for $k \geq \ell$. Additionally, $w_\ell \neq 0$ for $\ell \geq \ell_0$ for a suitable ℓ_0.

Let $(X_\ell)_{\ell\ge\ell_0}$ be the sequence of random variables with support $\mathbb{N}_0$ defined by

$$\mathbb{P}(X_\ell \le k) = \frac{w_{\ell k}}{w_\ell}.$$

Define

$$\delta := s_1(1)/s_0(1), \qquad \rho := \min\left(\log|a_2|/\log a_1, 2\right) - 1.$$

If s_0 does not have any zero in $|z| \le 1$ and $\delta > 0$, then the asymptotic formula

$$\frac{w_{\ell k}}{w_\ell} = \exp(-\delta\ell/a^k)(1+o(1)) \tag{5.12}$$

holds as $\ell \to \infty$ for $k = \log_a \ell + \mathcal{O}(1)$. Hence the shifted random variable $X_\ell - \log_a \ell$ converges weakly to a limiting distribution if ℓ runs through a subset of the positive integers such that the fractional part $\{\log_a \ell\}$ of $\log_a \ell$ converges.

The expected value of X_ℓ is

$$\mathbb{E}X_\ell = \log_a \ell + \log_a \delta + \frac{\gamma}{\log a} + \frac{1}{2} + \Psi_0(\log_a \ell + \log_a \delta) + O\left(\frac{\log^{\rho+3}\ell}{\ell^\rho}\right), \tag{5.13}$$

the variance is

$$\begin{aligned}\mathbb{V}X_\ell = \frac{\pi^2}{6\log^2 a} + \frac{1}{12} + \Psi_1(\log_a \ell + \log_a \delta) - \frac{2\gamma}{\log a}\Psi_0(\log_a \ell + \log_a \delta) \\ - \Psi_0^2(\log_a \ell + \log_a \delta) + \mathcal{O}\Big(\frac{\log^{\rho+4}\ell}{\ell^\rho}\Big),\end{aligned} \tag{5.14}$$

where γ is the Euler–Mascheroni constant, and $\Psi_0(x)$ and $\Psi_1(x)$ are periodic functions (with period 1 and mean value 0), given by the Fourier expansions

$$\Psi_0(x) = -\frac{1}{\log a}\sum_{n\ne 0}\Gamma\Big(-\frac{2n\pi i}{\log a}\Big)e^{2n\pi i x}, \tag{5.15}$$

$$\Psi_1(x) = \frac{2}{\log^2 a}\sum_{n\ne 0}\Gamma'\Big(-\frac{2n\pi i}{\log a}\Big)e^{2n\pi i x}. \tag{5.16}$$

Proof. Without loss of generality, we can assume $r_0(1)/s_0(1) = 1$, as otherwise $w_{\ell k}$ and w_ℓ are multiplied by a constant. Thus all assumptions of [59, Lemma 2.5] are satisfied except for the form of the denominator and numerator of G_k. However, all additional terms and their derivatives are of order $\mathcal{O}(c^{-k}a^{-k})$ and $\mathcal{O}(ka^{-k}c^{-k})$, respectively, for $|z| \le 1 + \frac{1}{k}$. Thus they do not influence the proof and the result in [59, Lemma 2.5].

Let $0 \le k_2 \le k_3$ denote the constants from [59]. The denominator S_k of G_k has exactly one simple singularity $\zeta_k = 1 + \delta a^{-k} + o(a^{-k})$ in the disk $\{z : |z| \le 1 + C\}$ for some $C > 0$ and $k \ge k_2$ (see the proof of [59, Lemma 2.5]). Since G_k and G are rational functions, G_k and G can be continued analytically beyond their dominant singularities ζ_k and 1, respectively. We have $\lim_{k\to\infty} \operatorname{Res}_{z=\zeta_k} G_k(z) = \operatorname{Res}_{z=1} G(z) = 1$. Thus [86, Theorem 1] implies (5.12) and the limiting distribution.

The coefficients $w_{\ell k}$ and w_ℓ of z^ℓ in G_k and G, respectively, coincide for $k \ge \ell$. Thus the support of X_ℓ is finite. Furthermore, the condition on s_0 implies that $w_\ell = 1 + \mathcal{O}(\kappa^\ell)$ for a constant $0 \le \kappa < 1$ by singularity analysis. Thus, the expectation is

$$\mathbb{E}X_\ell = \sum_{k\ge 0} k\mathbb{P}(X_\ell = k) = \sum_{k=0}^{\ell}\Big(1 - \frac{w_{\ell k}}{w_\ell}\Big) = \sum_{k=0}^{\ell}(1 - w_{\ell k}) + \mathcal{O}(\ell\kappa^\ell). \tag{5.17}$$

Then, (5.13) follows from [59, Lemma 2.5]. The second moment is

$$\mathbb{E}X_\ell^2 = \sum_{k\geq 0} k^2 \mathbb{P}(X_\ell = k) = \sum_{k=0}^{\ell} (2k+1)\Big(1 - \frac{w_{\ell k}}{w_\ell}\Big) \tag{5.18}$$

$$= \sum_{k=0}^{\ell} (2k+1)(1-w_{\ell k}) + \mathcal{O}(\ell^2 \kappa^\ell). \tag{5.19}$$

As $\sum_{k=0}^{\ell}(1-w_{\ell k})$ has already been computed for the expectation, we are left with $\sum_{k=0}^{\ell} k(1-w_{\ell k})$. We use the estimate

$$w_{\ell k} = \exp(-\ell\delta/a^k)(1 + \mathcal{O}(k/a^k) + \mathcal{O}(\ell k/(a^k c^k))) + \mathcal{O}((1+C/2)^{-\ell})$$

for $k_3 \leq k \leq n$ and $c = a^\rho$ from [59]. Replacing $w_{\ell k}$ with $\exp(-\ell\delta/a^k)$ yields the error terms (see [59])

$$|w_{\ell k} - \exp(-\ell\delta/a^k)| = \begin{cases} \mathcal{O}(\ell^{-2}) & \text{for } 0 \leq k \leq \log_a(\ell\delta/(4\log\ell)), \\ \mathcal{O}(\log_a^{\rho+2}\ell/\ell^\rho) & \text{for } \log_a(\ell\delta/(4\log\ell)) \leq k \leq 5\log_a\ell, \\ \mathcal{O}(\ell^{-3}) & \text{for } 5\log_a\ell \leq k \leq \ell. \end{cases} \tag{5.20}$$

As $1-\exp(-\ell\delta/a^k)$ is exponentially small for $k > \ell$, we obtain

$$\sum_{k=0}^{\ell} k(1-w_{\ell k}) = \sum_{k\geq 0} k(1-\exp(-\ell\delta/a^k)) + \mathcal{O}\Big(\frac{\log^{\rho+4}\ell}{\ell^\rho}\Big). \tag{5.21}$$

The Mellin transform (see [28]) of the harmonic sum $F(x) = \sum_{k\geq 0} k(1-\exp(-x/a^k))$ is

$$F^*(s) = \frac{-a^s}{(1-a^s)^2}\Gamma(s)$$

for $-1 < \Re s < 0$. The singular expansion of this Mellin transform at $\Re s = 0$ is

$$\begin{aligned} F^*(s) \asymp & -\frac{1}{\log^2 a} s^{-3} + \frac{\gamma}{\log^2 a} s^{-2} + \Big(\frac{1}{12} - \frac{1}{2\log^2 a}\Big(\gamma^2 + \frac{\pi^2}{6}\Big)\Big) s^{-1} \\ & - \sum_{n\neq 0} \frac{\Gamma(-\chi_n)}{\log^2 a}(s+\chi_n)^{-2} - \sum_{n\neq 0} \frac{\Gamma'(-\chi_n)}{\log^2 a}(s+\chi_n)^{-1} \end{aligned}$$

for $\chi_n = \frac{2\pi i n}{\log a}$. Thus,

$$\begin{aligned} F(x) = & \frac{1}{2}\log_a^2 x + \frac{\gamma}{\log a}\log_a x - \frac{1}{12} + \frac{1}{2\log^2 a}\Big(\gamma^2 + \frac{\pi^2}{6}\Big) \\ & - \frac{\log_a x}{\log a} \sum_{n\neq 0} \Gamma(-\chi_n)\exp(2\pi i n \log_a x) \\ & + \frac{1}{\log^2 a} \sum_{n\neq 0} \Gamma'(-\chi_n)\exp(2\pi i n \log_a x) + \mathcal{O}(x^{-1}). \end{aligned}$$

Thus, $\mathbb{V}X_\ell = \mathbb{E}X_\ell^2 - (\mathbb{E}X_\ell)^2$, (5.18), (5.17), (5.13) and (5.21) give the variance as stated in (5.14). □

Theorem 5.4. *Let $q \geq 2$ be even. Then the expected number of iterations when adding two SSDE of length ℓ with von Neumann's algorithm is*

$$\log_q \ell + \log_q \delta + \frac{\gamma}{\log q} + \frac{5}{2} + \Psi_0(\log_q \ell + \log_q \delta) + \mathcal{O}(\ell^{-1} \log^4 \ell) \tag{5.22}$$

where

$$\delta = \frac{(q-1)(4q^{10} + 10q^9 + 18q^8 - 4q^7 - 10q^6 + 7q^5 + 44q^4 - 29q^3 - 8q^2 - 20q + 16)}{4q^3(q+1)^2(4q^7 - q^5 - 6q^4 + 8q^3 + 2q - 4)},$$

$\Psi_0(x)$ is a 1-periodic function with mean 0 given by the Fourier expansion

$$\Psi_0(x) = -\frac{1}{\log q} \sum_{k \neq 0} \Gamma\Big(-\frac{2k\pi i}{\log q}\Big) e^{2k\pi i x}. \tag{5.23}$$

The variance of the number of iterations is

$$\frac{\pi^2}{6 \log^2 q} + \frac{1}{12} + \Psi_1(\log_q \ell + \log_q \delta) - \frac{2\gamma}{\log q} \Psi_0(\log_q \ell + \log_q \delta) - \Psi_0^2(\log_q \ell + \log_q \delta) + \mathcal{O}(\ell^{-1} \log^5 \ell) \tag{5.24}$$

where Ψ_1 is a 1-periodic function with mean 0 given by the Fourier expansion

$$\Psi_1(x) = \frac{2}{\log^2 q} \sum_{k \neq 0} \Gamma'\Big(-\frac{2k\pi i}{\log q}\Big) e^{2k\pi i x}. \tag{5.25}$$

The asymptotic formula

$$\mathbb{P}_\ell(t(\boldsymbol{x}, \boldsymbol{y}) \leq k) = \exp(-\delta \ell / q^k)(1 + o(1))$$

holds as $\ell \to \infty$ for $k = \log_q \ell + \mathcal{O}(1)$. The random variable $t(\boldsymbol{X}, \boldsymbol{Y}) - \log_q \ell$ converges weakly to a double-exponential random variable if ℓ runs through a subset of the positive integers such that the fractional part $\{\log_q \ell\}$ converges.

Remark 5.7.1. A similar result for $q \geq 4$ was obtained in [59] using the same probabilistic model as in Remark 5.5.5. This changes the main term of the expected value. In [59], the logarithm of the main term was taken to the base α^{-1} with $\alpha = q^{-1} - q^{-4} + \mathcal{O}(q^{-5})$. In contrast, we here obtain the logarithm of the main term in (5.22) to the base q, which is a more natural constant appearing in this context.

For $q = 2$, this result is contained in [59].

Proof. Let $\mathbb{P}_\ell$ be the (exact) equidistribution of all SSDE of length ℓ. For $k > \ell + 2$, we know that $\mathbb{P}_\ell(t(\boldsymbol{X}, \boldsymbol{Y}) \leq k) = 1$ because an input sequence of length ℓ traverses at most ℓ solid edges in the automaton in Figure 5.8.

If we use the approximate equidistribution $W_\ell = (1 + \mathcal{O}(\xi^\ell))\mathbb{P}_\ell$ of all SSDE of length ℓ, an exponentially small error term is introduced. Because of the finite support, this error term does not change the main term of the expectation, the variance and the distribution function. Thus, also the limiting distribution remains the same.

We will use Theorem 5.3 with the generating function

$$G_k(z) = \sum_{\ell \geq 0} w_{\ell k} z^\ell$$

for $w_{\ell k} = W_\ell(t(\boldsymbol{x}, \boldsymbol{y}) - 2 \leq k)$, $k \geq 0$. To construct this generating function, we use the same techniques as in [59].

The generating function $G_k(z)$ counts the weighted number of paths in the automaton $\mathcal{N}_{\mathrm{SSDE}}$ of the pattern $\ldots \mathcal{B}^+\mathcal{R}^{\{1,k\}}\mathcal{B}^+\mathcal{R}^{\{1,k\}}\ldots$ where $\mathcal{B}^+$ is an arbitrary non-empty sequence of dotted transitions and $\mathcal{R}^{\{1,k\}}$ is a non-empty sequence of solid transitions of length at most k. The first transition can be a dotted or a solid transition. We stop with either arbitrarily many dotted transitions or at most k solid transitions, where we have to take into account the special situation in states 2 or 7 in the automaton in Figure 5.8 (see also Remark 5.6.1): Because of the solid transition starting in 2 and 7 with label 0, we are not allowed to stop with k solid transitions in state 2 or 7 but only with at most $k-1$ ones.

To find the generating functions for $\mathcal{B}^+$ and $\mathcal{R}^{\{1,k\}}$, we use the transition matrices for the dotted and the solid parts of the automaton $\mathcal{N}_{\mathrm{SSDE}}$.

Let $q \geq 6$. The transition matrix R for the solid transitions of automaton $\mathcal{N}_{\mathrm{SSDE}}$ is a 12×12 matrix given in Table A.4 in the appendix (using Lemma 5.5.2). The transition matrix B for the dotted transitions of automaton $\mathcal{N}_{\mathrm{SSDE}}$ is given in Table A.5 (using Lemma 5.5.2). The order of the states is given in (5.11) and also in Table A.3 in the appendix.

The (matrix) generating function for arbitrary non-empty dotted paths $\mathcal{B}^+$ is

$$B^+(z) = (I - zB)^{-1} - I.$$

The entry (i, j) of this matrix is the generating function of non-empty dotted paths of arbitrary length starting in state i and leading to state j. For arbitrary non-empty solid paths, the (matrix) generating function is

$$R^+(z) = (I - zR)^{-1} - I.$$

To obtain the (matrix) generating function $R^{\{1,k\}}$ for non-empty solid paths $\mathcal{R}^{\{1,k\}}$ of length at most k, we have to restrict each entry of R^+ corresponding to an infinite geometric series to a finite geometric series.[3] We will illustrate this procedure on

$$\frac{-q^4z + 10q^3z - 3q^2z^2 - 24q^2z + 10qz^2 + 8z^2}{-8q^4 + 8q^3z - 8q^2z + 8qz^2}, \tag{5.26}$$

the entry at position $(5, 1)$ of R^+. The partial fraction decomposition of (5.26) with respect to z is

$$-\frac{(3q+2)(q-4)}{8q} + \frac{(q-4)(q+4)}{4(q+1)} \cdot \frac{1}{1+z/q^2} + \frac{(q-1)(q-2)(q-4)}{8q(q+1)} \cdot \frac{1}{1-z/q}.$$

By truncating the infinite geometric sum $(1-z)^{-1}$ after $k+1$ summands, i.e., by replacing it with $(1-z^{k+1})(1-z)^{-1}$, we obtain

$$-\frac{(3q+2)(q-4)}{8q} + \frac{(q-4)(q+4)}{4(q+1)} \cdot \frac{1-(-z/q^2)^{k+1}}{1+z/q^2} + \frac{(q-1)(q-2)(q-4)}{8q(q+1)} \cdot \frac{1-(z/q)^{k+1}}{1-z/q}.$$

Let

$$M_k(z) = \begin{pmatrix} 0 & B^+(z) \\ R^{\{1,k\}}(z) & 0 \end{pmatrix}$$

[3] It is also possible to use $R^{\{1,k\}}(z) = zR + \cdots + z^kR^k = (I - z^{k+1}R^{k+1})(I - zR)^{-1} - I$. However, this involves a power of the *symbolic* matrix R with the *symbolic* exponent k. This would require a full symbolic eigenvalue decomposition of R. The approach chosen here avoids this by introducing the length restriction on each entry individually.

be the block matrix of total size 24×24. Then, the (matrix) generating function of non-empty paths $\dots\mathcal{B}^+\mathcal{R}^{\{1,k\}}\mathcal{B}^+\mathcal{R}^{\{1,k\}}\dots$ is

$$(I-M_k(z))^{-1}-I.$$

To take into account the initial states and the exit weights in the automaton $\mathcal{N}_{\mathrm{SSDE}}$, we define the initial vector

$$u=(0,0,0,0,0,0,0,0,1,0,0,0;0,0,0,0,0,0,0,0,0,1,0,0,0).$$

Using the exit weights in Table A.3 in the appendix, we further define the exit vector

$$\begin{aligned}v^\top=\Big(\frac{q+1}{q+2}\Big)^2&(4,1,2,0,1,0,2,2,1,1,4,2;4,1,2,2,1,1,2,2,1,1,4,2)^\top\\&+\Big(\frac{q+1}{q+2}\Big)^2M_{k-1}(z)(0,0,0,2,0,1,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0)^\top\end{aligned}$$

taking into account the special situation with states 2 and 7 in the automaton in Figure 5.8.

Then, the generating function is

$$G_k(z)=u((I-M_k(z))^{-1}-I)v+\Big(\frac{q+1}{q+2}\Big)^2$$

where we add the exit weight of state $(1,(0,0))$ because the empty word was not counted until now. The result is

$$G_k(z)=\frac{r_0(z)+\big(\frac{z}{q}\big)^k r_1\big(z,\big(\frac{z}{q}\big)^k,\big(-\frac{z}{q^2}\big)^k\big)}{(1-z)s_0(z)+\big(\frac{z}{q}\big)^k s_1(z)+\big(-\frac{z}{q^2}\big)^k s_2\big(z,\big(\frac{z}{q}\big)^k,\big(-\frac{z}{q^2}\big)^k\big)}\tag{5.27}$$

with

$$\begin{aligned}r_0(z)&=4q^7(q+1)^3(4z^2-3q^2z-q^3)(2qz^4-4z^4+8q^3z^2-q^5z^2-6q^4z^2+4q^7),\\s_0(z)&=4q^7(q+1)(q+2)^2(z+q)(z-q^2)(2qz^4-4z^4+8q^3z^2-q^5z^2-6q^4z^2+4q^7),\\s_1(z)&=-(q+z)z^2(q+2)^2q^4(4q^{12}+6q^{11}z+2q^{10}z^2-4q^{11}-24q^{10}z\\&\quad-8q^9z^2+24q^{10}+26q^9z+4q^8z^2+5q^7z^3-7q^6z^4-48q^9\\&\quad-20q^8z+18q^7z^2-9q^6z^3+34q^5z^4+5q^4z^5+32q^8+36q^6z^2\\&\quad-32q^5z^3-59q^4z^4-25q^3z^5-112q^5z^2+84q^4z^3+40q^3z^4\\&\quad+44q^2z^5+64q^4z^2-48q^3z^3+4q^2z^4-36qz^5-16qz^4+16z^5)\end{aligned}$$

and some polynomials r_1 and s_2 in z, $(z/q)^k$ and $(-z/q^2)^k$ with coefficients in $\mathbb{Q}[q]$. The polynomial s_0 does not have any zeros in the closed unit disc. We have $r_0(1)\neq 0$ and $\delta>0$.

For $q\leq 4$, the construction of the generating function $G_k(z)$ is the same, only the matrices R and B and the vectors u and v are slightly different. Nevertheless, (5.27) including the definitions of all the occurring polynomials is still valid.

By Theorem 5.3, we obtain the expectation, the variance, the distribution function and the limiting distribution of the non-negative truncation of $t(\boldsymbol{x},\boldsymbol{y})-2$. From (5.20) and the monotonicity of $w_{\ell k}$, we know that $w_{\ell-2}=w_{\ell-1}=\mathcal{O}(\ell^{-2})$. Therefore, the results transfer to the random variable $t(\boldsymbol{x},\boldsymbol{y})$ as stated in the theorem. □

CHAPTER 6

Automata and Transducers in the SageMath Mathematical Software System

This chapter is a tutorial for the finite state machine package within the mathematical software system SageMath. The package is included in SageMath since version 5.13 [94], see [48] and was developed in the framework of this thesis. Its idea is to conveniently work with automata and transducers. Thus, all results of this thesis are implemented as methods of this finite state machine package and the examples are computed by using them accordingly.

This chapter corresponds to the tutorial [49], which is submitted for publication. The code of the finite state machine package can be found at `http://git.sagemath.org/sage.git/tree/src/sage/combinat/finite_state_machine.py?id=6.7` and `http://git.sagemath.org/sage.git/tree/src/sage/combinat/finite_state_machine_generators.py?id=6.7`. This chapter and the package itself are joint work with Clemens Heuberger and Daniel Krenn.

6.1. Introduction

6.1.1. Automata and Transducers. Automata and transducers are studied as an object and used as a tool in discrete mathematics and theoretical computer science. To clarify those terms we start with an informal description. An *automaton* reads the letters of an input string using a finite memory. When finished, it either accepts its input or not. A *transducer* additionally writes an output string. A common way to model those machines are a *set of states* together with *transitions* from one state to another. This can also be seen as a directed graph.

At the beginning, the current state of the automaton or transducer is one of its *initial states*. Whenever a letter is read, the current state changes according to its outgoing transitions (or more precisely, to the input labels of its outgoing transitions, which were assigned at creation time). If we are working with a transducer, then, whenever some transition is used, a word can be written as output, too. If, after reading the whole input, the current state is one of the machine's *final states*, then the input gets *accepted* (otherwise *rejected*).

For our purposes, the set of states will always be finite, and therefore, we use *finite state machine* as an umbrella term for automaton and transducer. For a general reference on automata and transducers, see Hopcroft, Motwani and Ullman [65] or Sakarovitch [90].

6.1.2. Finite State Machines in SageMath. For the last couple of years, more and more researchers are using the free and open source mathematical software system SageMath [97]. It offers an immense amount of mathematical objects combined with algorithms for working with them. Contributions to its code-base are subjected to a transparent peer-review process.

This tutorial introduces the finite state machine module implemented in and contributed to SageMath by the authors. One of our main motivations was to allow the use of the mathematical objects available in SageMath in the construction and manipulation of automata and

transducers. On the other hand, mathematical objects associated with finite state machines such as the underlying digraph or its (weighted) adjacency matrix can then further be processed in SageMath to e.g. compute asymptotic expressions. This process is illustrated in Section 6.3.

It should also be mentioned that one of the authors wrote a similar unpublished package for Mathematica [104], which is used in [6–8,37,43,45,47,57,60–64]. But this implementation was of limited scope and the rise of SageMath made it clear that it is ripe for a redesigned version. Having the finite state machine module readily available in a publicly available and continuously maintained system also leads to more transparency in the computational parts of publications.

There are quite a lot of implementations for finite state machines available [73]. One of the fastest libraries is OpenFST [82], for which a Python interface [87] exists, too. But since it is written in C/C++, it does not work well with the mathematical objects defined in SageMath. Other non-Python modules are, for example, [3,13,92]. There also exist a couple of Python packages, e.g. [4,32,88], which can also be found on the Python-Wiki[1]. Some of those are specialized (and thus not flexible enough) and implement only partial support for both automata and transducers. It seems that some of them are even out-dated and not developed any further.

6.1.3. What Can You Find in This Tutorial? The aim of this piece of work is to demonstrate some of the functionality of the developed finite state machines package in the form of a tutorial. Detailed documentation of the available methods and their parameters can, as usual, be found in the SageMath documentation[2]. There, further examples are presented as well.

Along the way, a new result on a digit system related to the non-adjacent form [89] is proved. The new digit expansion we consider has base 2 and uses the digit set $\{-2,-1,0,1,2\}$. We compute the expected value of the Hamming weight, i.e., the number of non-zero digits, of this digit expansion of integers less than 2^k. Although the digit set is larger than the one for the non-adjacent form, it turns out that the expected value of this new digit expansion is worse than that of the standard binary expansion and therefore worse than that of the non-adjacent form.

Let us give a brief overview on how to show the result mentioned above and why it is an enormous advantage to use SageMath for constructing and simulating automata and transducers. We start at the beginning, namely with the generation, which can be done in several ways. First, we can simply list all transitions of a finite state machine (cf. Section 6.2.1). Second, we can use transition functions written in SageMath to construct a transducer (cf. Section 6.2.3, but we will use it in several other places as well). This is of course possible for any type of finite state machine. Another way is to construct machines by suitably combining smaller building blocks: We manipulate and combine several finite state machines properly (Sections 6.2.4 and 6.3.2). A couple of variants and more advanced constructions will also be shown (see, for example, Sections 6.3.3 and 6.3.5). When the desired automaton or transducer is finally constructed, we obtain for example the adjacency matrix (Section 6.3.7) or the asymptotic behavior of the output (cf. Section 6.3.8) by just one function call.

[1]The Python-Wiki can be found at `https://wiki.python.org/moin/FiniteStateMachine`.

[2]The SageMath documentation of the finite state machine module can be found at `http://www.sagemath.org/doc/reference/combinat/sage/combinat/finite_state_machine.html`.

Note that all of these steps can be computed within SageMath. Thus, we can use everything offered by the mathematical software system and construct powerful finite state machines to analyze, without shuffling data from one system to another.

6.1.4. How Do I Get This Awesome New Finite State Machines Package? To keep it short: it is already included in SageMath[3]. If you are using the finite state machines package of SageMath in your own work, please let us know.

6.2. Three Kinds of Calculating the Non-Adjacent Form as a Warm-Up

Before we start with our tutorial, let us have a look at the terms used in this section. We start by explaining the classical *non-adjacent form*, abbreviated as *NAF*, cf. Reitwiesner [89], of an integer. It is a representation with base 2 and digits -1, 0 and $+1$, such that two neighboring digits are not both non-zero. This means that we forbid 11, $\bar{1}1$, $1\bar{1}$ and $\bar{1}\bar{1}$ in the digit expansion where we abbreviated $\bar{1} = -1$. For example, we have

$$12 = 1 \cdot 16 + 0 \cdot 8 - 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = (10\bar{1}00)_2. \tag{6.1}$$

It can be shown that this leads to a unique representation of each integer, cf. Reitwiesner [89].

Note that in the following (more precisely, from Section 6.2.4 on), we will add a summand $0 \cdot \frac{1}{2}$ at expansions like (6.1). This means, we write

$$12 = (10\bar{1}00.0)_2.$$

It will turn out that this is convenient, since we are working with halves at lot in our main example in Section 6.3.

Usually, digit expansions are written from the most significant digit (on the left-hand side) to the least significant digit (on the right-hand side). In the context of transducers, this is often reversed. There, a digit expansion starts on left-hand side with the least significant digit and the most significant one is on the right-hand side. Therefore, we have two different notations. For digit expansions, like in $(10\bar{1}00)_2$, we write the least significant digit on the right. For inputs and outputs of finite state machines, like in `[0, 0, -1, 0, 1]`, we write the least significant digit on the left. Consequently, we also speak of trailing zeros if we append zeros after the most significant digit.

6.2.1. Creating a Transducer from Scratch. In [60, Figure 2], a transducer for converting the binary expansion of an integer n into its non-adjacent form is given. We reproduce it here as Figure 6.1 and directly translate it into SageMath. We write[4]

```
NAF1 = Transducer([('I', 0, 0, None), ('I', 1, 1, None),
                   (0, 0, 0, 0), (0, 1, 1, 0),
                   (1, 0, 0, 1), (1, 2, 1, -1),
                   (2, 1, 0, 0), (2, 2, 1, 0)],
                  initial_states=['I'], final_states=[0],
                  input_alphabet=[0, 1])
```

[3]The basic version was included into SageMath 5.13 [94] (for details see the relevant ticket [48] on the SageMath trac server). A huge bunch of features is included in SageMath 6.2. Some of the latest improvements are either already merged in the current development branch or still in some branch on the trac server. To work with all features of this tutorial, use SageMath 6.7 or later.

[4]This document was created using SageTEX, which comes along with SageMath. It allows the following: We type SageMath source code directly in the TEX-document, this code is then executed by SageMath and the corresponding outputs (results) are typeset here.

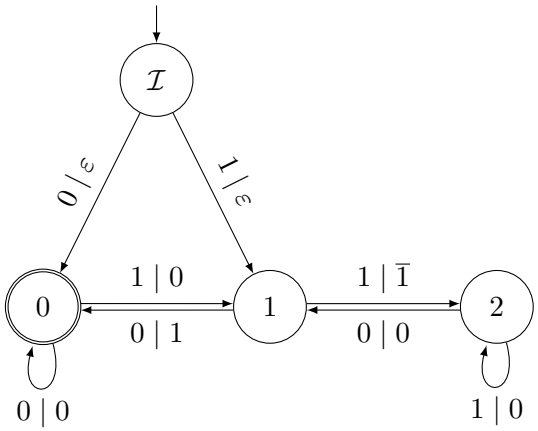


FIGURE 6.1. Transducer to compute the non-adjacent form.

to construct this transducer with states `'I'` (the string `I`), `0`, `1` and `2` (the integers 0, 1 and 2, respectively). The list of 4-tuples defines the transitions of the transducer. For example, `(1, 2, 1, -1)` is a transition from state 1 to state 2 with input 1 and output -1. Input here means reading a (more precisely, the next) digit of the binary expansion of n. The output is a digit of the non-adjacent form, written step-by-step. Note that we read and write the expansions from the least significant digit to the most significant one and we start at the digit corresponding to $2^0 = 1$.

This approach required us to manually model the digit conversion as a transducer (or, in this particular instance, by a reference to available literature). Shouldn't there be an easier method using the full power of SageMath? For sure, and we will do so in later examples to demonstrate various approaches for constructing transducers.

6.2.2. The Non-Adjacent Form of Twelve. For convenience, we set `NAF = NAF1`. We can use this transducer to compute the non-adjacent form of, for example, our lucky number twelve[5], which was used already at the beginning of this warm-up. We decide to use

```
sage.combinat.finite_state_machine.FSMOldProcessOutput = False
```

which activates the "new behavior" of the finite state machine package[6] in SageMath.

Then, as a first try, we type

```
NAF_of_12 = NAF(12.digits(base=2))
```

and get

```
ValueError: Invalid input sequence.
```

[5]You may ask why 12 is our lucky number. In fact, it is not! But it is not that bad. This number is, beside the actual 13 our second lucky number. The reason of preferring 12 over 13 is simple and of educational character: The digit expansion (used in this tutorial) of 13 is too symmetric. This may lead to confusion whether those expansions are read from left to right or the other way round.

[6]There was change in the output behavior of a couple of commands. In order not to break backwards compatibility, the old output is deprecated for at least a year, according to the *SageMath Developer's Guide*. After that, the new behavior will be the default. In order to get the new one already, we use the flag `FSMOldProcessOutput`. Similarly, there is `FSMOldCodeTransducerCartesianProduct`, which we are going to set to `False` as well. This is no longer needed in 2016.

An error message? Huh? So did we make a mistake in our construction? Fortunately not; the transducer has just not finished yet: With the input `[0, 0, 1, 1]`, which is the binary expansion of 12 from the least significant to the most significant digit, we would stop in the non-final state with label 2, as we can see by typing

```
NAF.process(12.digits(base=2)),
```

which results in `(False, 2, [0, 0, -1])`. Here, the first component indicates whether the input is accepted or not, the second component is the label of the state where we stopped, and the third component of this triple is the output of the transducer if this state would be final.

By adding enough trailing zeros to the expansion of 12, we reach a final state and we ensure that all carries are processed. We type

```
NAF_of_12 = NAF(12.digits(base=2) + [0, 0, 0])
```

and get the output `[0, 0, -1, 0, 1, 0]`. This list corresponds to the digits of the non-adjacent form of 12, starting with the digit corresponding to 1 at the left, and then continuing with the digits corresponding to 2, 4, 8, 16, and 32.

But do we really want to think about trailing zeros? This should be done by SageMath. And that is possible by

```
NAF = NAF.with_final_word_out(0)
```

This function constructs a final output for every state. The final output of a state is appended to the "normal" output if we stop in this state reading some input. Transducers with final output are called *subsequential*, cf. [91]. The method `with_final_word_out` computes the final output by reading as many zeros as necessary to reach a final state (if possible). The corresponding output is then the final output.

Now, we compute the non-adjacent form of 12 again by

```
NAF_of_12 = NAF(12.digits(base=2))
```

without thinking about how many trailing zeros we have to add. And the result `[0, 0, -1, 0, 1]` is still the same as before (except for one trailing zero).

6.2.3. Calculating the Non-Adjacent Form with Less Thinking. A different approach to construct a transducer calculating the non-adjacent form is via a transition function.

To get this function, we think about the following algorithm rewriting the binary expansion to the NAF: We start by determining the least significant digit n_0 of the non-adjacent form of the integer n. This can be decided by looking at the two least significant digits of the binary expansion: If n is even, then the digit of the non-adjacent form is zero, if n is odd, it is 1 or $\bar{1}$, depending on n modulo 4. As the next step of this algorithm, we have to compute the non-adjacent form of $\frac{1}{2}(n - n_0)$.

Reformulating this as a transition function leads to the following code:

```
def NAF_transition(state_from, read):
    if state_from == 'I':
        write = None
        state_to = read
        return (state_to, write)
    current = 2*read + state_from
    if current % 2 == 0:
        write = 0
```

```
    elif current % 4 == 1:
        write = 1
    else:
        write = -1
    state_to = (current - write) / 2
    return (state_to, write)
```

Here, `%` is the remainder of the integer division in SageMath.

The transducer defined by this transition function can be built by

```
NAF2 = Transducer(NAF_transition,
                  initial_states=['I'],
                  final_states=[0],
                  input_alphabet=[0, 1]).with_final_word_out(0)
```

We can check whether the two transducers are the same by

```
NAF == NAF2
```

which, luckily, yields `True`.

6.2.4. A Third Construction of the Same Transducer. The non-adjacent form can also be constructed in the following way. We start with the binary expansions of $\frac{3n}{2}$ and of $\frac{n}{2}$. We subtract each digit of $\frac{n}{2}$ from the corresponding digit of $\frac{3n}{2}$. This leads to a digit expansion of n with digits $\{-1, 0, 1\}$ in base 2. One can prove that this digit expansion is the non-adjacent form of n (cf. [17], see also [100, Theorem 10.2.4]).

For this construction we need a few simple transducers (as, for example one for multiplying by 3 and one for performing subtraction), which we combine later appropriately. We will also reuse these machines in a later example for the $\frac{3}{2}-\frac{1}{2}$-non-adjacent form in Section 6.3.

So let us start with the times-3-transducer, i.e., one that takes a binary number n as input and outputs $3n$ (in binary). We do this, as above, by a transition function. We define

```
def f(state_from, read):
    current = 3*read + state_from
    write = current % 2
    state_to = (current - write) / 2
    return (state_to, write)
```

to compute the next output digit (`write`) and the new carry (encoded in `state_to`) from the input digit (`read`) and the previous carry (`state_from`) in the multiplication-by-3-algorithm. From this transition function we get the following transducer:

```
Triple = Transducer(f, input_alphabet=[0, 1],
                    initial_states=[0],
                    final_states=[0]).with_final_word_out(0)
```

Eager as we are, we test this construction by

```
three_times_four = Triple(4.digits(base=2))
```

and get `[0, 0, 1, 1]`, which equals 12. Hooray!

Back to business; our goal is to calculate binary-$3n$ minus binary-n. To do so, we need a transducer which acts as identity (for the binary-n-part), i.e., writes out everything that is read in. Here,

```
Id = Transducer([(0, 0, 0, 0), (0, 0, 1, 1)],
                initial_states=[0], final_states=[0],
```

```
                    input_alphabet=[0, 1])
```

does the trick. Maybe this is a good point to mention that a couple of of commonly used transducers are already prebuilt in SageMath. We get the above also by

```
prebuiltId = transducers.Identity([0, 1])
```

where we just have to specify the alphabet `[0, 1]`. Note that there are various different transducers in the generator `transducers`; take a look into its SageMath documentation.

As a next step (before we dart for subtraction), we want a transducer which produces pairs of the digits of $3n$ and of n simultaneously. This can be achieved with

```
sage.combinat.finite_state_machine.\
    FSMOldCodeTransducerCartesianProduct = False
Combined_3n_n = Triple.cartesian_product(Id).relabeled()
```

As in Section 6.2.2, we have to deactivate backwards compatible code; in this instance we have to use `FSMOldCodeTransducerCartesianProduct`.

The function `relabeled()` just renames the states with integers starting at 0 (more precisely, returns a copy with relabeled states). Let us test this machine by

```
twelve_and_four = Combined_3n_n(4.digits(base=2))
```

It returns `[(0, 0), (0, 0), (1, 1), (1, None)]`, which seems to be correct.

We further construct a transducer computing the component-wise difference: Its input is a pair like the output of `Combined_3n_n` and the output is the difference of the two entries. We use the operator

```
def g(read0, read1):
    return ZZ(read0) - ZZ(read1)
```

and generate the transducer by

```
Minus = transducers.operator(g, input_alphabet=[None, -1, 0, 1])
```

Here we use that `ZZ(None)` is 0.

Of course, there is not only a prebuilt identity transducer, but also a prebuilt transducer for component-wise difference, available as

```
prebuiltMinus = transducers.sub([-1, 0, 1])
```

But unfortunately, it can only work with numbers, and we also want to subtract `None`. The final outputs are the reason: Sometimes, one component is `None`. For example, the final output of state 1 is

```
final_word_out = Combined_3n_n.state(1).final_word_out
```

which yields `[(1, None)]`.

Finally, by

```
NAF3 = Minus(Combined_3n_n).relabeled()
```

we obtain a transducer computing the non-adjacent form of $3n - n = 2n$. This means, `NAF3` is built as the composition of `Minus` and `Combined_3n_n`, which could also have been called by using the method `.composition`.

Let us test this construction. For example,

```
NAF_of_12 = NAF3(12.digits(base=2))
```

returns `[0, 0, 0, -1, 0, 1]`. This is, once again, the non-adjacent form expansion of 12, see (6.1), but now starting with the digit corresponding to $\frac{1}{2}$ (which is obviously 0) at the left, and then continuing with the digits corresponding to 1, 2, 4, 8, 16 and 32.

Now we have finished our warm-up and are ready for the main example, which will be dealing with $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form.

6.3. An Example: Three-Half–One-Half-Non-Adjacent Forms

We start this example by answering the question posed by the following title.

6.3.1. What is the Three-Half–One-Half-Non-Adjacent Form? We have (or, at least, we can calculate) the non-adjacent forms of $\frac{3n}{2}$ and of $\frac{n}{2}$. Inspired by the construction of the NAF presented in Section 6.2.4 (as a remainder: we subtracted two binary expansions), we do this also with these two NAFs. Thus, we define the $\frac{3}{2}$–$\frac{1}{2}$-*non-adjacent form* of an integer n as the digit expansion obtained by subtracting each digit of the NAF of $\frac{n}{2}$ from the corresponding digit of the NAF of $\frac{3n}{2}$. This leads to a digit expansion of n with digits $\{-2,-1,0,1,2\}$ in base 2. For example, the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form of—guess which number comes now—12 is

$$12 = (10010.0)_2 - (10\bar{1}0.0)_2 = (1\bar{1}020.0)_2.$$

In a first step, we want to calculate this new expansion; see the following sections. On the one hand, we are lazy and want to reuse as much as possible from the already constructed finite state machines. On the other hand, we are motivated to use our new knowledge working with those automata and transducers. So the idea will be to combine several already known transducers appropriately.

6.3.2. Combining Small Transducers to a Larger One. We first combine the transducers `Triple` and `NAF` to obtain a transducer to compute the non-adjacent form of $3n$. For convenience, we choose `NAF = NAF3`, because there we do not have to consider an empty output of a transition.

We combine by

```
NAF3n = NAF(Triple)
```

which builds the composition of the two transducers involved and therefore gives us a gadget to get the non-adjacent form of $3n$.

Next, we construct a transducer which builds the non-adjacent forms of $3n$ and n simultaneously by

```
Combined_NAF_3n_n = NAF3n.cartesian_product(NAF).relabeled()
```

The function `cartesian_product` sounds familiar, since we used it in Section 6.2.4 already. It constructs a transducer which writes pairs of digits.

Finally, by reusing `Minus`, we construct

```
T = Minus(Combined_NAF_3n_n).relabeled()
```

This transducer finally computes the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form. To get some information like the number of states of the finite state machine, we type `T` in SageMath and see

```
Transducer with 9 states
```

Let us continue with the example from the beginning. To compute this new digit expansion of 12, we type

```
expansion_of_12 = T(12.digits(base=2))
```

The output is

```
[0, 0, 0, 2, 0, -1, 1]
```

which is the $\frac{3}{n}$–$\frac{1}{2}$-non-adjacent form of 12 starting with the digit corresponding to $\frac{1}{4}$. This starting digit has the following reasons: The output of the transducer `NAF` starts with the digit corresponding to $\frac{1}{2}$ when reading n. We use the non-adjacent form of $\frac{n}{2}$, which thus starts at the digit corresponding to $\frac{1}{4}$.

6.3.3. An Alternative Construction. We had three different ways to get a transducer calculating the non-adjacent form, so you might guess that there are also several ways to construct a transducer computing the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form of n. Indeed there are. In this part of the tutorial, we describe another way, which uses a more general function of our SageMath's finite state machines module.

We want a transducer processing `NAF3n` and `NAF` (we already have constructed those two) at the same time and subtracting the output. Therefore we define the functions

```
def minus(trans1, trans2):
    if trans1.word_in == trans2.word_in:
        return (trans1.word_in,
                trans1.word_out[0] - trans2.word_out[0])
    else:
        raise LookupError
```

and

```
from itertools import izip_longest
def final_minus(state1, state2):
    return [x - y for x, y in
        izip_longest(state1.final_word_out,
                     state2.final_word_out,
                     fillvalue=0)]
```

to combine the output of two transitions or two final outputs. As we have only one input sequence and we want to process both transducers simultaneously, we can only combine transitions with the same input digit. Otherwise, an exception is raised. In contrast to our first construction, we pad the final outputs with zeros instead of the default padding with `None` as it is used by `cartesian_product`.

Now we construct the new transducer as the product of `NAF3n` and `NAF` by

```
Talternative = NAF3n.product_FiniteStateMachine(
                            NAF, minus,
                            final_function=final_minus).relabeled()
```

This transducer computes the new digit expansion of n from the least significant digit to the most significant one, starting with the digit corresponding to $\frac{1}{4}$.

The function `product_FiniteStateMachine` combines any two transitions of `NAF3n` and `NAF` and constructs a new transition in `T` with input and output defined by `minus`. Of course, in this example, `product_FiniteStateMachine` is much to complicated. But, one can combine finite state machines in a much more general way with `product_FiniteStateMachine` than with `cartesian_product` (we used the latter already twice in some constructions above).

With this construction, we obtain the same transducer as `T`. This can also be checked by

```
Talternative == T
```

which yields `True`.

6.3.4. Getting a Picture. Up to now, we constructed a couple of finite state machines, but we never really saw one. So, the time has come to do this. With `T.plot()` we get a first graphical representation of the transducer. This was easy, but we are not fully satisfied. For example, the labels of the transitions are missing. And, maybe we want to rearrange the states a little bit to obtain less crossings of the transitions. This can be achieved by the following: `view(T)` gives a second graphical representation of the transducer. Maybe the arrangement of the states is not nicer than before, but we will improve this a lot.

We first choose the coordinates of the states by

```
T.set_coordinates({
    0: (-2, 0.75),
    1: (0, -1),
    2: (-6, -1),
    3: (6, -1),
    4: (-4, 2.5),
    5: (-6, 5),
    6: (6, 5),
    7: (4, 2.5),
    8: (2, 0.75)})
```

Furthermore, in transition labels, we prefer "$\overline{1}$" over "-1", so we choose the appropriate formatting function. Additionally, we choose in which directions the arrows with the final outputs should point.

```
T.latex_options(format_letter=T.format_letter_negative,
                accepting_where={
                  0: 'right',
                  1: 'below',
                  2: 'below',
                  3: 'below',
                  4: 60,
                  5: 'above',
                  6: 'above',
                  7: 120,
                  8: 'left'},
                accepting_show_empty=True)
```

Now, the output of `view(T)` in SageMath looks like Figure 6.2. The \$-symbol signals the end of the input sequence. Further customization of the underlying Ti*k*Z-code is possible, see the documentation of `latex_options`.

On the other hand, by typing `latex(T)` we get this Ti*k*Z-code for the transducer, which can be used to include a figure of the finite state machine in a LaTeX-document, like it was done for this tutorial. To succeed, we need to use the package `tikz` and to include the line `\usetikzlibrary{automata}` in the preamble of the LaTeX-document.

6.3.5. Recognizing Everything. We did a good job computing the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form of an integer in the sections above. But what if someone gives you an expansion and asks: "Is this one of the leading actors of this play?", what should we tell him? Reformulated this means that we want to recognize whether a given digit expansion is a $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent

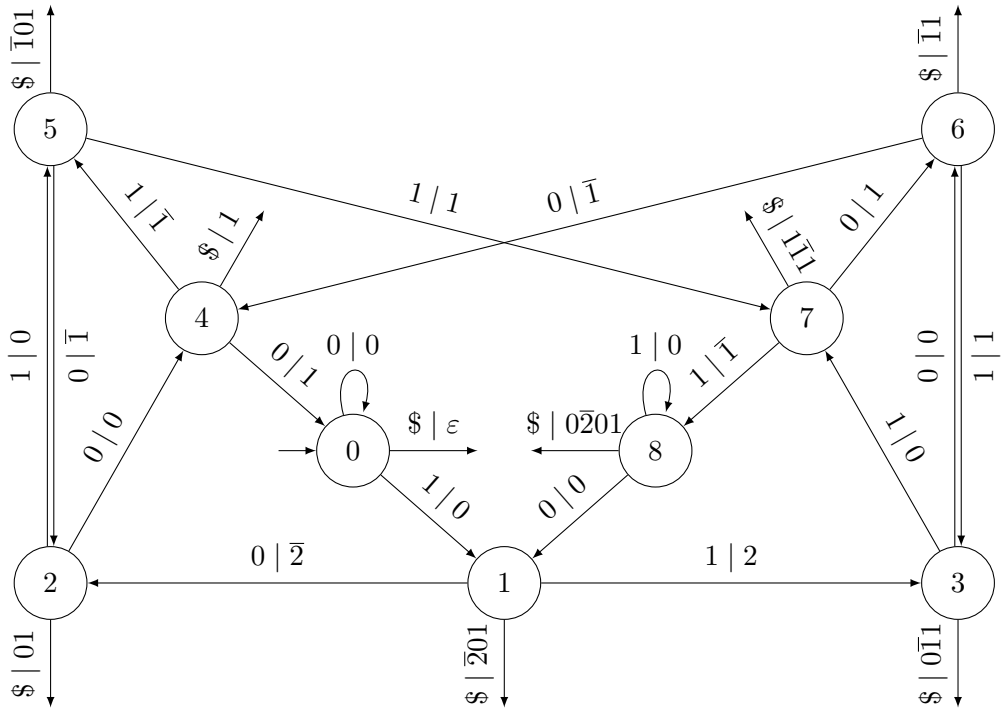


FIGURE 6.2. Transducer `T` to compute the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form of n.

form or not. Since we are used to finite state machines now, we look for a method involving those. This will now bring automata into play. An automaton reads the expansion and says either yes (a correct expansion) or no.

From the previous constructions we have the transducer `T` which writes $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent forms. Now, we simply "forget" the input of every transition and only consider the output labels, which we can do by

```
R = T.output_projection()
```

The automaton `R` recognizes exactly all $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent forms.

By typing `R`, we see that this automaton has 10 states. As some of the transitions have a longer input word (type `R.transitions()` to see all transitions), we call

```
R = R.split_transitions()
```

to split up these transitions into paths. Then we have more states than before: There are 23 states.

Calling the function `R.is_deterministic()` returns `False`, which clearly means our automaton is non-deterministic. As deterministic automata are much nicer, we ask for an equivalent deterministic automaton by

```
Rdet = R.determinisation()
```

Be aware that the determinisation of an automaton can increase the number of states exponentially. But in our case, `Rdet` has only 22 states, that are less than before. Why is that? The reason is that some states are equivalent in some sense. Thus, they could be "merged".

But first, let us test this automaton with the previously calculated expansion of 12. We can check whether it accepts it by `Rdet(expansion_of_12)`, like it should be. We get `True`, which we expected.

As mentioned above, the determinisation can lead to very large automata. Thus, we ask ourselves the following question: Does there exist an equivalent automaton with less states? To find out, let us try

```
Rdet1 = Rdet.minimization()
```

to get a minimal equivalent deterministic automaton. Fortunately, it has only 17 states. For this minimization Moore's algorithm [76] was used.

One might toss in that Moore's minimization algorithm only works well for deterministic automata, but `Rdet` originally comes from the non-deterministic automaton `R`. Is there a possibility to directly minimize `R`? Yes, of course! We should try another algorithm for non-deterministic automata: Brzozowski's algorithm [14]. Let us apply it directly on `R`, without determinizing before, by

```
Rdet2 = R.minimization(algorithm='Brzozowski')
```

which leads to the same minimal deterministic automaton with 17 states we already had.

So, now we are also familiar with automata in SageMath. But, a lot more is possible with these machines than mentioned here.

6.3.6. (Heavy) Weights. Our original motivation to study the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form comes from analyzing its (Hamming) weight, i.e., the number of non-zero digits. We want to compare the Hamming weights of the different digit expansions: standard binary expansion, non-adjacent form and $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form.

Using the ideas of the constructions above, this is also not difficult. We construct a transducer computing the weight of the input by

```
def weight(state_from, read):
    write = ZZ(read != 0)
    return (0, write)
Weight = Transducer(weight, input_alphabet=srange(-2, 2+1),
                    initial_states=[0], final_states=[0])
```

Here, we use the convention that the integer values of `True` and `False` are 1 and 0, respectively. The transducer `Weight` writes a 1 for every non-zero input, which means that the weight is encoded in unary in the output string.

There exists also a prebuilt transducer which we could use instead of our own construction. It is available via

```
prebuiltWeight = transducers.weight(srange(-2, 2+1))
```

Composing the weight-transducer with the one calculating the $\frac{3}{2}$–$\frac{1}{2}$-NAF by

```
W = Weight(T)
```

we end up with a transducer with 9 states computing the Hamming weight of this new digit expansion of n (given in binary). For instance,

```
W(12.digits(base=2))
```

yields `[0, 0, 0, 1, 0, 1, 1]`, which means the weight is 3.

The transducer `W` can be further simplified by preponing the output in each state by the command

```
W.prepone_output()
```

The function `prepone_output` tries to shift the output letters from one transition to another one such that each letter is written as early as possible. If, for example, all transitions leaving a (non-final) state write the same output letter 0, then this letter 0 can already be written by all transitions leading to this state.

If you wonder, why there is the word "heavy" in the title of this part, read on until the end of the example.

6.3.7. Also Possible: Adjacency Matrices. We want to asymptotically analyze the expected value of the Hamming weight of our new digit expansion for all positive integers less than 2^k, where k is a fixed large number.

One way to do the asymptotic analysis is by means of the adjacency matrix of the transducer. By

```
var('y')
def am_entry(trans):
    return y^add(trans.word_out) / 2
A = W.adjacency_matrix(entry=am_entry)
```

we obtain a matrix, where the entry at (k, l) is y^h if there is a transition with output h from state k to l and 0 otherwise. The generated adjacency matrix is

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{2}y^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2}y & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}y & 0 & 0 & 0 & 0 & \frac{1}{2}y & 0 \\ 0 & 0 & 0 & \frac{1}{2}y & \frac{1}{2}y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}y & 0 & \frac{1}{2} \\ 0 & \frac{1}{2}y^2 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}.$$

For $y = 1$, this is simply the transition probability matrix. Its normalized left eigenvector to the eigenvalue 1 gives the stationary distribution. We write

```
(pi_not_normalized,) = (A.subs(y=1) - A.parent().identity_matrix())\
                           .left_kernel().basis()
pi = pi_not_normalized / pi_not_normalized.norm(p=1)
```

and obtain `(1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9)`.

To obtain the average Hamming weight of the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form, we compute the expected output vector in each state as

```
expected_output = derivative(A, y).subs(y=1)\
                          * vector(len(W.states())*[1])
```

and obtain $\left(1, 0, 0, 0, \frac{1}{2}, 1, 1, \frac{1}{2}, 1\right)$. Note that the derivative here simply computes the expected output for every transition. We could also have called `adjacency_matrix` with a suitably modified entry function.

The expected density is therefore

```
pi * expected_output
```

which yields $\frac{5}{9}$. This means the main term of the average number of non-zero digits in $\frac{3}{2}$–$\frac{1}{2}$-NAFs of length k is $\frac{5}{9}k$.

6.3.8. More on the Hamming Weight by Letting SageMath Do the Work. We have the impression that the analysis of the previous section could be done (or, rephrased, we want that this should be done) more automatically. Indeed, we can let SageMath do the work for us, and it does it very well: It not only outputs the mean of the Hamming weight, but also its variance and more.

By

```
var('k')
moments = W.asymptotic_moments(k)
```

we obtain a dictionary whose entries are the expectation and the variance of the sum of the output of the transducer, and the covariance of the sum of the output and the input of the transducer (cf. [56] or Chapter 3). The probability model is the equidistribution on all input sequences of a fixed length k.

The expected value of the Hamming weight of the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form is

$$\frac{5}{9}\,k + \mathcal{O}\,(1).$$

as k tends to infinity.

This function can also give us the variance of the Hamming weight of the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form, which is

$$\frac{44}{243}\,k + \mathcal{O}\,(1).$$

Of course we could do a lot more beautiful stuff. We could construct a bivariate generating function. From this, we could obtain more terms and better error terms of the asymptotic expansion of the expected value, the variance and higher moments. We could also prove a central limit theorem. And everything by using the full power of SageMath. But, again, we do not want to go into details here. We refer to the book of Flajolet and Sedgewick [30] for details on the asymptotic analysis of digit expansions and other sequences.

6.3.9. What Does This Mean for This Brand New Digit Expansion? In the past several sections, we were able to calculate the average Hamming weight of the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form asymptotically by the help of the finite state machines package in SageMath. But what does this result tell us?

So let us compare this digit expansion with the standard binary expansion and the classical non-adjacent form. The expected value of the Hamming weight of the standard binary expansion can be calculated by

```
expectation_binary = Id.asymptotic_moments(k)['expectation']
```

which gives

$$\frac{1}{2}\,k + \mathcal{O}\,(1).$$

Of course, it was not necessary to use the transducer `Weight` here (since we only have digits 0 and 1). The expected value of the weight of the NAF can be obtained with the code

```
expectation_NAF = Weight(NAF).asymptotic_moments(k)['expectation']
```

which produces the weight

$$\frac{1}{3}\,k + \mathcal{O}\,(1),$$

cf. also [77]. Note that in this particular construction (only digits -1, 0 and 1), we could have used the prebuilt transducer

```
Abs = transducers.abs([-1, 0, 1])
```

instead of the weight-transducer.

Both values are (asymptotically) less than the

$$\frac{5}{9}\,k + \mathcal{O}\,(1)$$

of the $\frac{3}{2}$–$\frac{1}{2}$-non-adjacent form, which means that those expansions have much more non-zero digits and therefore are much "heavier" on average. So, from a point of view of minimizing the Hamming weight, this new expansion is disappointing: it uses more digits but realizes a larger weight.

6.4. Selected Technical Details of the Finite State Machines Package

This final section contains a few selected technical details of the implementation of the finite state machines module in SageMath [73].

6.4.1. Class Structure. The class structure of the finite state machine bundle is quite easy. We have one main class, namely `FiniteStateMachine` (inheriting from `SageObject`), which contains most of the code and algorithms, in particular the ones valid generally. Derived from it, there is a class `Automaton`, which contains, among others, routines to produce deterministic automata and to minimize automata. Similarly, a class `Transducer` is also derived from `FiniteStateMachine`. As an example, transducers, in contrast to general finite state machines and automata, provide a method to be simplified.

Moreover, states and transitions are encapsulated in classes `FSMState` and `FSMTransition`, respectively.

6.4.2. Storage of States and Transitions. Each instance of the class

`FiniteStateMachine`

stores a list of its states. Additionally (to speed up searching) a dictionary mapping labels of states to references of states is created.

Transitions are stored differently. Each state in the finite state machine holds a list of its outgoing transitions. Since a transition also knows the state to which it is going, we can see this as a variant of an asymmetric doubly-linked list[7].

All input and output words of transitions and states are always stored as lists, even if the output word is only one digit.

[7]Note that we do not have a "fully" doubly-linked list, since we do not store a list of incoming transitions to a state. In the rare occasions where we need a "fully" doubly-linked list, e.g. `prepone_output`, we build it on the fly.

APPENDIX A

Transition Matrices

This appendix contains the transition matrices used in Chapter 5.

$(-1,(0,0))$	$(0,(0,0))$	$(1,(0,0))$
$(d-1)(d-2)y$	$-2d^2-2dq+q^2+6d+3q-4$	$(d+q-1)(d+q-2)x$
$(d-1)dy$	$-2d^2-2dq+q^2+2d+q$	$(d+q)(d+q-1)x$
$(d+1)dy$	$-2d^2-2dq+q^2-2d-q$	$(d+q+1)(d+q)x$

TABLE A.1. Transition matrix of $\mathcal{S}_{(q,d)}$ in Section 5.5.1 multiplied with $2q^2$. The order of the states is given in the first line.

$6q^2-12q+8$	4	$4(q-2)y$	8	$4q-8$	$4q-8$	8	$4(q-2)x$	$(q-2)(q-4)y$	$(q-2)(q-4)x$	$2y$	$2x$	$4q-8$	$4q-8$
$8q^2$	0	0	0	0	0	0	0	0	0	0	0	0	0
$6(q-2)q$	0	$4qy$	0	0	$4q$	0	0	$2(q-2)qy$	0	0	0	$8q$	0
$2(qy+2q-2y)q$	0	$4qy^2$	0	0	$4qy$	0	0	$2(q-2)qy^2$	0	0	0	0	0
$2(3q-2)q$	0	$4(q-2)y$	8	0	$4q-8$	8	0	$2(q-2)(q-4)y$	0	0	0	$8q-16$	0
$2(3q-2)q$	0	0	8	$4q-8$	0	8	$4(q-2)x$	0	$2(q-2)(q-4)x$	0	0	0	$8q-16$
$2(qx+2q-2x)q$	0	0	0	$4qx$	0	0	$4qx^2$	0	$2(q-2)qx^2$	0	0	0	0
$6(q-2)q$	0	0	0	$4q$	0	0	$4qx$	0	$2(q-2)qx$	0	0	0	$8q$
$6(q-2)q$	4	$4qy$	8	$4q-16$	$4q$	8	$4(q-4)x$	$(q-2)qy$	$(q-4)(q-6)x$	$2y$	$2x$	$4q$	$4q-16$
$6(q-2)q$	4	$4(q-4)y$	8	$4q$	$4q-16$	8	$4qx$	$(q-4)(q-6)y$	$(q-2)qx$	$2y$	$2x$	$4q-16$	$4q$
$4(q-2)q$	0	0	0	0	0	0	0	$4(q-2)qy$	0	0	0	$16q$	0
$4(q-2)q$	0	0	0	0	0	0	0	0	$4(q-2)qx$	0	0	0	$16q$
$(3qy+3q-6y-2)q$	4	$4qy^2$	0	$4q-8$	$4qy$	0	$4(q-2)x$	$(q-2)qy^2$	$(q-2)(q-4)x$	$2y$	$2x$	0	0
$(3qx+3q-6x-2)q$	4	$4(q-2)y$	0	$4qx$	$4q-8$	0	$4qx^2$	$(q-2)(q-4)y$	$(q-2)qx^2$	$2y$	$2x$	0	0

TABLE A.2. Transition matrix of $\mathcal{S}_{\text{SSDE}}$ for $q \geq 8$ in Section 5.5.2 multiplied with $8q^2$. The order of the states is $\{(0,(0,0))\}$, $\{(0,(-1,1)),(0,(1,-1))\}$, $\{(-1,(-1,0)),(-1,(0,-1))\}$, $\{(-q/2,(-1,0)),(-q/2,(0,-1))\}$, $\{(0,(-1,0)),(0,(0,-1))\}$, $\{(0,(0,1)),(0,(1,0))\}$, $\{(q/2,(0,1)),(q/2,(1,0))\}$, $\{(1,(0,1)),(1,(1,0))\}$, $\{(-1,(0,0))\}$, $\{(1,(0,0))\}$, $\{(-1,(-1,-1))\}$, $\{(1,(1,1))\}$, $\{(-q/2,(0,0))\}$, $\{(q/2,(0,0))\}$.

$(1,(-1,1))$	$(4,(0,0))$	$(5,(0,1))$	$(2,(0,1))$	$(5,(0,0))$	$(2,(0,0))$	$(1,(-1,0))$	$(3,(0,1))$	$(1,(0,0))\}$	$(3,(0,0))$	$(4,(1,1))$	$(4,(0,1))$
4	1	2	2	1	1	2	2	1	1	4	2

TABLE A.3. Exit weights of $\mathcal{N}_{\text{SSDE}}$ in Section 5.7 multiplied with $\left(\frac{q+2}{q+1}\right)^2$. As discussed in (5.11), the states of $\mathcal{N}_{\text{SSDE}}$ are equivalence classes of states. For brevity, we list one representative for each state of $\mathcal{N}_{\text{SSDE}}$ to give the order of the states.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	$4q(q-2)$	$8q$	0	0
0	0	0	0	0	0	0	0	0	0	0	0
4	$(q-4)(q-6)$	0	8	0	$4(q-4)$	$4(q-4)$	8	$3q(q-2)$	$4q$	4	$4(q-4)$
0	0	0	0	0	0	0	0	0	0	0	0
0	$2(q-2)(q-4)$	0	8	0	$8(q-2)$	$4(q-2)$	8	$2q(q-2)$	0	0	$4(q-2)$
0	0	0	0	0	0	0	0	0	0	0	0
0	$(q-2)(q-4)$	0	8	0	$4(q-2)$	$4(q-2)$	8	$(3q-4)(q-2)$	$4(q-2)$	0	$4(q-2)$
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

TABLE A.4. Transition matrix R for the solid transitions in $\mathcal{N}_{\text{SSDE}}$ for $q \geq 6$ in Section 5.7 multiplied with $8q^2$. The order of the states is the same as in Table A.3.

0	0	0	0	0	0	0	0	$8q^2$	0	0	0
4	$2(q-4)^2$	0	16	0	$8(q-3)$	$8(q-3)$	16	$2(3q-2)(q-2)$	$8(q-1)$	4	$8(q-3)$
0	$2(q-2)(q-4)$	0	8	0	$8(q-2)$	$4(q-2)$	8	$2q(3q-2)$	0	0	$4(q-2)$
0	$2(q-2)(q-4)$	0	8	0	$8(q-2)$	$4(q-2)$	8	$2q(q-2)$	0	0	$4(q-2)$
4	$2(q-2)(q-4)$	0	8	0	$4(q-2)$	$8(q-2)$	8	$6q^2-12q+8$	$4(q-2)$	4	$8(q-2)$
0	$(q-2)(q-4)$	0	8	0	$4(q-2)$	$4(q-2)$	8	$(3q-4)(q-2)$	$4(q-2)$	0	$4(q-2)$
0	$2(q-2)(q-4)$	16	0	$8(q-2)$	0	$4(q-2)$	0	$2q(3q-2)$	0	0	$4(q-2)$
0	0	0	0	0	0	0	0	$4q^2$	0	0	0
4	$2(q-2)(q-4)$	16	0	$8(q-2)$	0	$8(q-2)$	0	$6q^2-12q+8$	0	4	$8(q-2)$
4	$(q-2)(q-4)$	0	0	0	0	$4(q-2)$	0	$q(3q-2)$	0	4	$4(q-2)$
0	$4(q-2)(q-4)$	0	0	0	$16(q-2)$	0	0	$4q(q-2)$	$16q$	0	0
0	$2(q-2)(q-4)$	0	8	0	$8(q-2)$	$4(q-2)$	$8z$	$6(q-2)q$	$8q$	0	$4(q-2)$

TABLE A.5. Transition matrix B for the dotted transitions in $\mathcal{N}_{\text{SSDE}}$ for $q \geq 6$ in Section 5.7 multiplied with $8q^2$. The order of the states is the same as in Table A.3.

Bibliography

[1] Jean-Paul Allouche and Jeffrey Shallit, *Automatic sequences: Theory, applications, generalizations*, Cambridge University Press, Cambridge, 2003.

[2] Tom Apostol, *Modular functions and Dirichlet series in number theory*, Graduate Texts in Mathematics, vol. 41, Springer, New York, 1976.

[3] *The automata standard template library*, `http://astl.sourceforge.net`, 2013.

[4] *automata 0.1.4*, `https://pypi.python.org/pypi/automata`, 2013.

[5] Roberto Avanzi, *A note on the signed sliding window integer recoding and a left-to-right analogue*, Selected Areas in Cryptography: 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers (H. Handschuh and A. Hasan, eds.), Lecture Notes in Comput. Sci., vol. 3357, Springer-Verlag, Berlin, 2005, pp. 130–143.

[6] Roberto Avanzi, Clemens Heuberger, and Helmut Prodinger, *Scalar multiplication on Koblitz curves. Using the Frobenius endomorphism and its combination with point halving: Extensions and mathematical analysis*, Algorithmica **46** (2006), 249–270.

[7] ———, *Arithmetic of supersingular Koblitz curves in characteristic three*, Cryptology ePrint Archive, Report 2010/436, 2010.

[8] ———, *Redundant τ-adic expansions I: Non-adjacent digit sets and their applications to scalar multiplication*, Des. Codes Cryptogr. **58** (2011), 173–202.

[9] Guy Barat and Peter J. Grabner, *Distribution of binomial coefficients and digital functions*, J. London Math. Soc. (2) **64** (2001), no. 3, 523–547.

[10] Nader L. Bassily and Imre Kátai, *Distribution of the values of q-additive functions on polynomial sequences*, Acta Math. Hungar. **68** (1995), no. 4, 353–361.

[11] Edward A. Bender and Fred Kochman, *The distribution of subword counts is usually normal*, European J. Combin. **14** (1993), no. 4, 265–275.

[12] Valérie Berthé and Michel Rigo (eds.), *Combinatorics, automata and number theory*, Encyclopedia Math. Appl., vol. 135, Cambridge University Press, Cambridge, 2010.

[13] *dk.brics.automaton 1.11-8*, `http://www.brics.dk/automaton/`, 2011.

[14] Janusz A. Brzozowski, *Canonical regular expressions and minimal state graphs for definite events*, Proc. Sympos. Math. Theory of Automata (New York, 1962), Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963, pp. 529–561.

[15] Emmanuel Cateland, *Suites digitales et suites k-régulières*, Ph.D. thesis, Université Bordeaux, 1992.

[16] Seth Chaiken, *A combinatorial proof of the all minors matrix tree theorem*, SIAM J. Alg. Disc. Meth. **3** (1982), no. 3, 319–329.

[17] S. H. Chang and N. Tsao-Wu, *Distance and structure of cyclic arithmetic codes*, Proc. Hawaii International Conference on System Sciences, vol. 1, 1968, pp. 463–466.

[18] Louis H. Y. Chen, Hsien-Kuei Hwang, and Vytas Zacharovas, *Distribution of the sum-of-digits function of random integers: a survey*, **11** (2014), 177–236.

[19] Jean Coquet, *Power sums of digital sums*, J. Number Theory **22** (1986), no. 2, 161–176.

[20] Hubert Delange, *Sur la fonction sommatoire de la fonction "somme des chiffres"*, Enseignement Math. (2) **21** (1975), 31–47.

[21] Persi Diaconis, *The distribution of leading digits and uniform distribution mod* 1, Ann. Probability **5** (1977), no. 1, 72–81.

[22] Persi Diaconis and Jason Fulman, *Combinatorics of balanced carries*, Adv. in Appl. Math. **59** (2014), 8–25.

[23] *NIST Digital library of mathematical functions*, `http://dlmf.nist.gov/`, Release 1.0.9 of 2014-08-29, 2010, Online companion to [81].

[24] Michael Drmota, *Random trees*, SpringerWienNewYork, 2009.
[25] Michael Drmota and Peter J. Grabner, *Analysis of digital functions and applications*, Combinatorics, automata and number theory (Valérie Berthé and Michel Rigo, eds.), Encyclopedia Math. Appl., vol. 135, Cambridge University Press, Cambridge, 2010, pp. 452–504.
[26] Philippe Dumas, *Joint spectral radius, dilation equations, and asymptotic behavior of radix-rational sequences*, Linear Algebra Appl. **438** (2013), no. 5, 2107–2126.
[27] Philippe Dumas, *Asymptotic expansions for linear homogeneous divide-and-conquer recurrences: Algebraic and analytic approaches collated*, Theoret. Comput. Sci. **548** (2014), 25–53.
[28] Philippe Flajolet, Xavier Gourdon, and Philippe Dumas, *Mellin transforms and asymptotics: Harmonic sums*, Theoret. Comput. Sci. **144** (1995), 3–58.
[29] Philippe Flajolet, Peter Grabner, Peter Kirschenhofer, Helmut Prodinger, and Robert F. Tichy, *Mellin transforms and asymptotics: digital sums*, Theoret. Comput. Sci. **123** (1994), 291–314.
[30] Philippe Flajolet and Robert Sedgewick, *Analytic combinatorics*, Cambridge University Press, Cambridge, 2009.
[31] Philippe Flajolet, Wojciech Szpankowski, and Brigitte Vallée, *Hidden word statistics*, J. ACM **53** (2006), no. 1, 147–183.
[32] *FSA – Finite State Automaton processing in Python*, `http://www.osteele.com/software/python/fsa/`, 2004.
[33] Chris D. Godsil and Gordon Royle, *Algebraic graph theory*, Graduate texts in mathematics, vol. 207, Springer Verlag (New York), 2001.
[34] Massimiliano Goldwurm and Roberto Radicioni, *Average value and variance of pattern statistics in rational models*, Implementation and Application of Automata (Jan Holub and Jan Žďárek, eds.), Lecture Notes in Comput. Sci., vol. 4783, Springer Berlin Heidelberg, 2007, pp. 62–72.
[35] Peter J. Grabner, Clemens Heuberger, and Helmut Prodinger, *Subblock occurrences in signed digit representations*, Glasg. Math. J. **45** (2003), 427–440.
[36] ———, *Distribution results for low-weight binary representations for pairs of integers*, Theoret. Comput. Sci. **319** (2004), 307–331.
[37] ———, *Counting optimal joint digit expansions*, Integers **5** (2005), no. 3, A9.
[38] Peter J. Grabner, Clemens Heuberger, Helmut Prodinger, and Jörg Thuswaldner, *Analysis of linear combination algorithms in cryptography*, ACM Trans. Algorithms **1** (2005), 123–142.
[39] Peter J. Grabner and Hsien-Kuei Hwang, *Digital sums and divide-and-conquer recurrences: Fourier expansions and absolute convergence*, Constr. Approx. **21** (2005), 149–179.
[40] Peter J. Grabner and Jörg M. Thuswaldner, *On the sum of digits function for number systems with negative bases*, Ramanujan J. **4** (2000), no. 2, 201–220.
[41] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete mathematics. A foundation for computer science*, second ed., Addison-Wesley, 1994.
[42] Florian Heigl and Clemens Heuberger, *Analysis of digital expansions of minimal weight*, 23rd Intern. Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA'12), DMTCS Proceedings, 2012, pp. 399–411.
[43] Clemens Heuberger, *Minimal expansions in redundant number systems: Fibonacci bases and greedy algorithms*, Period. Math. Hungar. **49** (2004), 65–89.
[44] ———, *Hwang's quasi-power-theorem in dimension two*, Quaest. Math. **30** (2007), 507–512.
[45] ———, *Redundant τ-adic expansions II: Non-optimality and chaotic behaviour*, Math. Comput. Sci. **3** (2010), 141–157.
[46] Clemens Heuberger, Personal Communication, 2013–2015.
[47] Clemens Heuberger, Rajendra Katti, Helmut Prodinger, and Xiaoyu Ruan, *The alternating greedy expansion and applications to left-to-right algorithms in cryptography*, Theoret. Comput. Sci. **341** (2005), 55–72.
[48] Clemens Heuberger, Daniel Krenn, and Sara Kropf, *Finite state machines, automata, transducers*, `http://trac.sagemath.org/ticket/15078`, 2013, module in Sage 5.13.
[49] Clemens Heuberger, Daniel Krenn, and Sara Kropf, *Automata and transducers in the computer algebra system Sage*, 2014, arXiv:1404.7458 [cs.FL].
[50] Clemens Heuberger and Sara Kropf, *Analysis of the binary asymmetric joint sparse form*, Combin. Probab. Comput. **23** (2014), 1087–1113.

[51] Clemens Heuberger and Sara Kropf, *FiniteStateMachine.asymptotic_moments: New method*, `http://trac.sagemath.org/ticket/16145`, 2014, merged in Sage 6.3.beta2.
[52] Clemens Heuberger, Sara Kropf, and Helmut Prodinger, *Asymptotic analysis of the sum of the output of transducers*, 25th International Conference on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA'14), DMTCS-HAL Proceedings Series, vol. BA, 2014, pp. 145–156.
[53] ———, *Analysis of carries in signed digit expansions*, 2015, arXiv:1503.08816 [math.CO].
[54] Clemens Heuberger, Sara Kropf, and Helmut Prodinger, *Analysis of carries in signed digit expansions—online resources*, `http://arxiv.org/src/1503.08816`, 2015.
[55] Clemens Heuberger, Sara Kropf, and Helmut Prodinger, *Output sum of transducers: Limiting distribution and periodic fluctuation*, Electron. J. Combin. **22** (2015), no. 2, 1–53.
[56] Clemens Heuberger, Sara Kropf, and Stephan Wagner, *Variances and covariances in the central limit theorem for the output of a transducer*, European J. Combin. **49** (2015), 167–187.
[57] Clemens Heuberger and James A. Muir, *Minimal weight and colexicographically minimal integer representations*, J. Math. Cryptol. **1** (2007), 297–328.
[58] Clemens Heuberger and Helmut Prodinger, *On minimal expansions in redundant number systems: Algorithms and quantitative analysis*, Computing **66** (2001), 377–393.
[59] ———, *Carry propagation in signed digit representations*, European J. Combin. **24** (2003), 293–320.
[60] ———, *Analysis of alternative digit sets for nonadjacent representations*, Monatsh. Math. **147** (2006), 219–248.
[61] ———, *The Hamming weight of the non-adjacent-form under various input statistics*, Period. Math. Hungar. **55** (2007), 81–96.
[62] ———, *On α-greedy expansions of numbers*, Adv. in Appl. Math. **38** (2007), 505–525.
[63] ———, *Analysis of complements in multi-exponentiation algorithms using signed digit representations*, Internat. J. Found. Comput. Sci. **20** (2009), 443–453.
[64] Clemens Heuberger, Helmut Prodinger, and Stephan G. Wagner, *Positional number systems with digits forming an arithmetic progression*, Monatsh. Math. **155** (2008), 349–375.
[65] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley series in computer science, Addison-Wesley, 2001.
[66] Hsien-Kuei Hwang, *On convergence rates in the central limit theorems for combinatorial structures*, European J. Combin. **19** (1998), 329–343.
[67] Imre Kátai and József Mogyoródi, *On the distribution of digits*, Publ. Math. Debrecen **15** (1968), 57–68.
[68] Tosio Kato, *Perturbation theory for linear operators*, Springer, 1976.
[69] Peter Kirschenhofer, *Subblock occurrences in the q-ary representation of n*, SIAM J. Algebraic Discrete Methods **4** (1983), no. 2, 231–236.
[70] ———, *On the variance of the sum of digits function*, Number-Theoretic Analysis (Edmund Hlawka and Robert F. Tichy, eds.), Lecture Notes in Mathematics, vol. 1452, Springer Berlin Heidelberg, 1990, pp. 112–116.
[71] Peter Kirschenhofer and Helmut Prodinger, *Subblock occurrences in positional number systems and Gray code representation*, J. Inform. Optim. Sci. **5** (1984), no. 1, 29–42.
[72] Donald E. Knuth, *The average time for carry propagation*, Nederl. Akad. Wetensch. Indag. Math. **40** (1978), 238–242.
[73] Daniel Krenn, Personal Communication, 2014.
[74] Blake Madill and Narad Rampersad, *The abelian complexity of the paperfolding word*, Discrete Math. **313** (2013), no. 7, 831–838.
[75] John W. Moon, *Some determinant expansions and the matrix-tree theorem*, Discrete Math. **124** (1994), 163–171.
[76] Edward F. Moore, *Gedanken experiments on sequential machines*, Automata Studies (Claude E. Shannon and John McCarthy, eds.), Annals of Mathematics Studies, no. 34, Princeton Universty Press, 1956, pp. 129–153.
[77] François Morain and Jorge Olivos, *Speeding up the computations on an elliptic curve using addition-subtraction chains*, RAIRO Inform. Théor. Appl. **24** (1990), 531–543.
[78] James A. Muir and Douglas R. Stinson, *Minimality and other properties of the width-w nonadjacent form*, Math. Comp. **75** (2006), 369–384.
[79] Fumihiko Nakano and Taizo Sadahiro, *A generalization of carries processes and Eulerian numbers*, Adv. in Appl. Math. **53** (2014), 28–43.

[80] Pierre Nicodème, Bruno Salvy, and Philippe Flajolet, *Motif statistics*, Theoret. Comput. Sci. **287** (2002), no. 2, 593–617.
[81] Frank W. J. Olver, Daniel W. Lozier, Ronald F. Boisvert, and Charles W. Clark (eds.), *NIST Handbook of mathematical functions*, Cambridge University Press, New York, 2010.
[82] *OpenFst Library 1.3.4*, `http://openfst.org`, 2013.
[83] William Parry, *Intrinsic Markov chains*, Trans. Amer. Math. Soc. **112** (1964), 55–66.
[84] Manfred Peter, *The asymptotic distribution of elements in automatic sequences*, Theoret. Comput. Sci. **301** (2003), 285–312.
[85] Helmut Prodinger, Personal Communication, 2013–2014.
[86] Helmut Prodinger and Stephan Wagner, *Bootstrapping and double-exponential limit laws*, Discrete Math. Theor. Comput. Sci. **17** (2015), no. 1, 123–144.
[87] *pyopenfst*, `http://code.google.com/p/pyopenfst/`, 2013.
[88] *python-automata 1.0*, `https://code.google.com/p/python-automata/`, 2007.
[89] George W. Reitwiesner, *Binary arithmetic*, Advances in Computers, Vol. 1, Academic Press, New York, 1960, pp. 231–308.
[90] Jacques Sakarovitch, *Elements of automata theory*, Cambridge University Press, Cambridge, 2009, Translated from the 2003 French original by Reuben Thomas.
[91] Marcel-Paul Schützenberger, *Sur une variante des fonctions sequentielles*, Theoret. Comput. Sci. **4** (1977), no. 1, 47–57.
[92] *SFST 1.4.6h*, `http://www.cis.uni-muenchen.de/~schmid/tools/SFST/`, 2013.
[93] Claude E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27** (1948), 379–423.
[94] William A. Stein et al., *Sage Mathematics Software (Version 5.13)*, The Sage Development Team, 2013, `http://www.sagemath.org`.
[95] ———, *Sage Mathematics Software (Version 6.3)*, The Sage Development Team, 2014, `http://www.sagemath.org`.
[96] ———, *Sage Mathematics Software (Version 6.5)*, The Sage Development Team, 2015, `http://www.sagemath.org`.
[97] ———, *Sage Mathematics Software (Version 6.7)*, The Sage Development Team, 2015, `http://www.sagemath.org`.
[98] Gérard Tenenbaum, *Sur la non-dérivabilité de fonctions périodiques associées à certaines formules sommatoires*, The mathematics of Paul Erdős, I (Ronald L. Graham and Jaroslav Nešetřil, eds.), Algorithms Combin., vol. 13, Springer, Berlin, 1997, pp. 117–128.
[99] Jörg M. Thuswaldner, *Summatory functions of digital sums occurring in cryptography*, Period. Math. Hungar. **38** (1999), no. 1-2, 111–130.
[100] Jacobus Hendricus van Lint, *Introduction to coding theory*, Graduate Texts in Mathematics, vol. 86, Springer, 1992.
[101] John von Neumann, *Collected works. Vol. V: Design of computers, theory of automata and numerical analysis*, The Macmillan Co., New York, 1963.
[102] Stephan Wagner, Personal Communication, 2013.
[103] Edmund T. Whittaker and George N. Watson, *A course of modern analysis*, Cambridge University Press, Cambridge, 1963, Reprint of the fourth (1927) edition.
[104] Wolfram Research, Inc., *Mathematica (Version 5.2)*, 2005.
[105] Antoni Zygmund, *Trigonometric series*, vol. I & II combined, Cambridge University Press, Cambridge, 2002.