On congruence schemes for constant terms and their applications

Armin Straub*
Department of Mathematics and Statistics
University of South Alabama

May 19, 2022

Abstract

Rowland and Zeilberger devised an approach to algorithmically determine the modulo p^r reductions of values of combinatorial sequences representable as constant terms (building on work of Rowland and Yassawi). The resulting p-schemes are systems of recurrences and, depending on their shape, are classified as automatic or linear. We revisit this approach, provide some additional details such as bounding the number of states, and suggest a third natural type of scheme that combines benefits of automatic and linear ones. We illustrate the utility of these "scaling" schemes by confirming and extending a conjecture of Rowland and Yassawi on Motzkin numbers.

1 Preliminaries

1.1 Introduction

Throughout, let p be a prime and denote with $\mathbb{Z}_p$ the p-adic integers. If $A(n)$ is a sequence of p-adic integers with the property that its ordinary generating function $\sum_{n\geq 0} A(n)x^n$ is algebraic over $\mathbb{Z}_p(x)$, then, for any integer $r \geq 1$, the reductions $A(n)$ modulo p^r are p-automatic (that is, there exists a finite state automaton which computes the values $A(n)$ modulo p^r from the base p digits of n; see Section 1.2 for a simple example, and [2] for an introduction to automatic sequences in general). A multivariate generalization of this result was proved by Christol, Kamae, Mendes France and Rauzy [8] in the case $r = 1$, while the extension to $r \geq 1$ is due to Denef and Lipshitz [9]. Based on the proof in [9], Rowland and Yassawi [19] provided a constructive proof of the following result.

Theorem 1.1 ([19, Theorem 2.1]). *Suppose that $A(n)$ is a sequence of p-adic integers that can be represented as the diagonal of a multivariate rational function in $\mathbb{Z}_p(x_1, x_2, \ldots, x_d)$. Then, for any integer $r \geq 1$, the reductions $A(n)$ modulo p^r are p-automatic.*

* *Email:* `straub@southalabama.edu`

Here, the diagonal of a rational function in $\mathbb{Z}_p(x_1, x_2, \ldots, x_d)$ with power series

$$\sum_{n_1, n_2, \ldots, n_d \geq 0} c(n_1, n_2, \ldots, n_d) x_1^{n_1} \cdots x_d^{n_d}$$

is the (univariate) sequence $c(n, n, \ldots, n)$. Bostan, Lairez and Salvy [5] recently showed that the diagonals of rational functions in $\mathbb{Z}(x_1, x_2, \ldots, x_d)$ are precisely those sequences expressible as multiple binomial sums. A conjecture of Christol [7] suggests that every integer sequence, which grows at most exponentially and which satisfies a linear recursion with polynomial coefficients, is of this form. This illustrates that Theorem 1.1 applies to a large class of the sequences naturally arising in combinatorics.

Under the assumptions of Theorem 1.1, Rowland and Yassawi [19] described practical algorithms to compute a finite state automaton that encodes the values $A(n)$ modulo p^r and applied these to a wide variety of combinatorial sequences, obtaining a host of fascinating and inspiring conjectures as well as elegantly reproving known results. Subsequently, Rowland and Zeilberger [20] provided a similar algorithm, as well as a clever and useful new variation, for the case of sequences $A(n)$ expressible as constant terms, meaning that

$$A(n) = \operatorname{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})], \tag{1}$$

where $P, Q \in \mathbb{Z}[\boldsymbol{x}^{\pm 1}]$ are Laurent polynomials in $\boldsymbol{x} = (x_1, \ldots, x_d)$. We revisit this approach in Section 2 and provide some additional details such as bounding the number of states in Theorem 2.4, resulting in bounds that are similar to those obtained by Rowland and Yassawi [19] for the case of diagonals of rational functions.

The two algorithms of Rowland and Zeilberger result in systems of recurrences, called (congruence) p-schemes, which depending on their shape are classified as *linear* or *automatic* (where an automatic p-scheme is a linear p-scheme that is equivalent to a finite state automaton). We add a third special type of p-scheme which we call *scaling* and which naturally lies between the two. For certain purposes, these scaling schemes combine benefits of automatic schemes and linear schemes: scaling schemes are (nearly) as easily analyzable as automatic ones, while their number of states and computational cost are often drastically reduced, much like for linear schemes.

If a sequence of constant terms $A(n)$ has p-adic valuation bounded by r, then the sequence of p-adic valuations of $A(n)$ is p-automatic as well, and a p-scheme for the valuations can be easily extracted from an automatic p-scheme for the values of $A(n)$ modulo p^r. We discuss this observation in Section 3 and reprove in Theorem 3.1 a result classifying the 2-adic valuation of Motzkin numbers that was conjectured by Amdeberhan, Deutsch and Sagan [10, Conjecture 5.5] and proven by Eu, Liu and Yeh [11]. We further observe that scaling p-schemes are particularly well suited for the purpose of studying p-adic valuations. As an application, we consider an open question of Rowland and Yassawi [19] that asks whether there exist infinitely many primes p such that p^2 never divides any Motzkin number $M(n)$. By computing congruence automata, Rowland and Yassawi showed that $p = 5$ and $p = 13$ are two such primes, and they conjectured that $31, 37, 61$ are such primes as well. We prove in Theorem 3.3 that their conjecture is true and extend it to all primes below 200, resulting in three additional primes with that property.

In order to perform the computations required for Theorem 3.3, we implemented the algorithm described in Section 2 in the open-source computer algebra system Sage [21]. This implementation is introduced in Section 4, followed by several examples and applications which reproduce and extend interesting computations and conjectures from [19] and [20].

Finally, in Section 5, we conclude with further motivation for seeking means to efficiently compute congruence schemes. In particular, we indicate open problems which show that, even in the case of the very well-studied Catalan numbers, intriguing new questions reveal themselves by studying congruence schemes.

1.2 Introductory examples

The Catalan numbers

$$C(n) = \frac{1}{n+1}\binom{2n}{n} = \binom{2n}{n} - \binom{2n}{n-1} \tag{2}$$

play a fundamental role [22] in combinatorics and have numerous combinatorial interpretations. It follows immediately from the latter representation in (2) that the Catalan numbers have the constant term expression

$$C(n) = \operatorname{ct}[(x^{-1} + 2 + x)^n (1 - x)]. \tag{3}$$

Based on this constant term expression (or an equivalent representation as the diagonal of a rational function), the algorithms of Rowland and Yassawi [19] and of Rowland and Zeilberger [20] can be used to construct finite state automata that describe the Catalan numbers modulo any fixed prime power.

Example 1.2. Figure 1, which is taken from [14], shows such a finite state automaton for the Catalan numbers $C(n)$ modulo 3. For instance, since 35 has the representation 1022 in base 3, to compute $C(35)$ modulo 3, we begin at the marked initial node and follow the arrows labeled 2, 2, 0 and 1 corresponding to the digits of 35 in base 3. After these four transitions, we are at the top-right node whose label 1 tells us that $C(35) \equiv 1$ modulo 3 (without computing that $C(35) = 3{,}116{,}285{,}494{,}907{,}301{,}262$). We note that a more transparent characterization can be obtained for Catalan numbers modulo any prime p through generalized Lucas congruences [15].

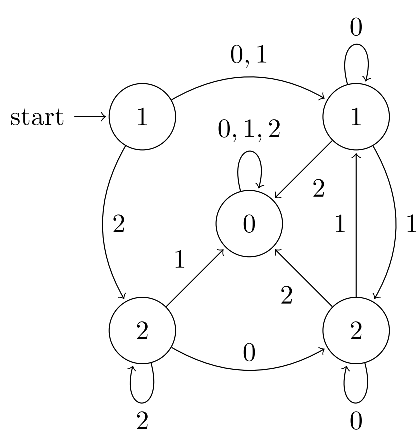

Figure 1: Congruence automaton for Catalan numbers modulo 3

Similar to the representation (3) for the Catalan numbers, the well-known sequence of Motzkin numbers $M(n)$ has the constant term representation

$$M(n) = \operatorname{ct}[(x^{-1} + 1 + x)^n (1 - x^2)], \tag{4}$$

which we will employ in the sequel as well.

As indicated in [20], any binomial coefficient sum of a certain kind can be transformed into a constant term representation. Famous instances of such sequences include the two Apéry sequences

$$\begin{aligned} B(n) &= \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k} = \operatorname{ct}\left[\frac{(x+1)(x+y)(x+y+1)}{xy}\right]^n, \\ A(n) &= \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 = \operatorname{ct}\left[\frac{(x+y)(z+1)(x+y+z)(y+z+1)}{xyz}\right]^n, \end{aligned} \tag{5}$$

which are the fundamental ingredients in Apéry's proofs [3], [17] of the irrationality of $\zeta(2)$ and $\zeta(3)$, respectively.

As a final example, we note that Gorodetsky [13] recently obtained particularly nice constant term representations for all Apéry-like sporadic sequences, allowing him to uniformly derive certain congruential properties.

1.3 Notation

In the sequel, we will use the vector notation $\boldsymbol{x} = (x_1, \ldots, x_d)$ and write, for instance, $\mathbb{Q}[\boldsymbol{x}^{\pm 1}] = \mathbb{Q}[x_1^{\pm 1}, \ldots, x_d^{\pm 1}]$ for the ring of Laurent polynomials in d variables with rational coefficients. We denote monomials as $\boldsymbol{x}^{\boldsymbol{k}} = x_1^{k_1} \cdots x_d^{k_d}$, where $\boldsymbol{k} = (k_1, \ldots, k_d)$ is the exponent vector.

We denote with Λ_p the Cartier operator

$$\Lambda_p\left[\sum_{\boldsymbol{k} \in \mathbb{Z}^d} a_{\boldsymbol{k}} \boldsymbol{x}^{\boldsymbol{k}}\right] = \sum_{\boldsymbol{k} \in \mathbb{Z}^d} a_{p\boldsymbol{k}} \boldsymbol{x}^{\boldsymbol{k}}.$$

Observe that, if $A(n) = \operatorname{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$, where $P, Q \in \mathbb{Z}[\boldsymbol{x}^{\pm 1}]$ are Laurent polynomials in $\boldsymbol{x} = (x_1, \ldots, x_d)$, then

$$\begin{aligned} A(pn+k) &= \operatorname{ct}[P(\boldsymbol{x})^{pn+k} Q(\boldsymbol{x})] \\ &\equiv \operatorname{ct}[P(\boldsymbol{x}^p)^n P(\boldsymbol{x})^k Q(\boldsymbol{x})] \pmod{p} \\ &= \operatorname{ct}[P(\boldsymbol{x})^n \Lambda_p[P(\boldsymbol{x})^k Q(\boldsymbol{x})]], \end{aligned} \tag{6}$$

where we used that $P(\boldsymbol{x})^{pn} \equiv P(\boldsymbol{x}^p)^n$ modulo p (see congruence (9) for a generalization modulo p^r). For the final equality note that $\operatorname{ct}[f(\boldsymbol{x}^p) g(\boldsymbol{x})] = \operatorname{ct}[f(\boldsymbol{x}) \Lambda_p[g(\boldsymbol{x})]]$ for any $f, g \in \mathbb{Z}[\boldsymbol{x}^{\pm 1}]$ because a term $a_{\boldsymbol{k}} \boldsymbol{x}^{\boldsymbol{k}}$ of $g(\boldsymbol{x})$ can contribute to the constant term only if (each component of) the exponent $\boldsymbol{k} = (k_1, \ldots, k_d)$ is divisible by p (since the latter is true for each term of $f(\boldsymbol{x}^p)$).

2 Congruence schemes

2.1 Linear and automatic congruence schemes

Let $A : \mathbb{Z}_{\geq 0} \to R$ be a sequence with values in a ring R. Following [20], we say that a *linear p-scheme* for $A(n)$ consists of sequences $A_0, A_1, \ldots, A_m : \mathbb{Z}_{\geq 0} \to R$ with $A_0 = A$ such that, for all

$i \in \{0, 1, \ldots, m\}$, $k \in \{0, 1, \ldots, p-1\}$ and $n \geq 0$,

$$A_i(pn + k) = \sum_{j=0}^{m} \alpha_{i,j}^{(k)} A_j(n) \tag{7}$$

for some $\alpha_{i,j}^{(k)} \in R$. Note that the linear p-scheme, including the values of all involved sequences, is determined by the transition coefficients $\alpha_{i,j}^{(k)}$ together with the initial conditions $c_i = A_i(0)$. In the sequel, we refer to the A_i as the *states* of the p-scheme. In particular, $m+1$ is the number of states of the p-scheme.

We note that $A(n)$ can be described by a linear p-scheme if and only if $A(n)$ is p-regular [1]. In the case where R is finite (in this paper, we only consider the case where $R = \mathbb{Z}/p^r\mathbb{Z}$ for some $r \geq 1$), these sequences are precisely the p-automatic ones.

Example 2.1. There exists a linear 3-scheme for the Catalan numbers $C(n)$ modulo 3 with two states $A_0, A_1 : \mathbb{N} \to \mathbb{Z}/3\mathbb{Z}$ and the following transitions:

$$\begin{aligned} A_0(3n) &= A_0(n) + A_1(n) & A_1(3n) &= 0 \\ A_0(3n+1) &= A_0(n) + A_1(n) & A_1(3n+1) &= A_0(n) + A_1(n) \\ A_0(3n+2) &= 2A_0(n) + A_1(n) & A_1(3n+2) &= A_0(n) + 2A_1(n) \end{aligned}$$

Together with the initial conditions

$$A_0(0) = 1, \quad A_1(0) = 0,$$

the above transitions uniquely describe all the values taken by the sequences A_0, A_1 and, therefore, the Catalan numbers $C(n)$ modulo 3. For instance, to determine $C(35)$ modulo 3, as in Example 1.2, we compute

$$A_0(35) = 2A_0(11) + A_1(11) = 2A_0(3) + A_1(3) = 2A_0(1) + 2A_1(1) = A_0(0) + A_1(0) = 1$$

which confirms that $C(35) \equiv 1$ modulo 3. We note that the above scheme is equivalent to the one given in [15, Example 1.1] though for the latter $A_0(n) + A_1(n)$ is chosen as the second state.

A linear p-scheme is called an *automatic p-scheme* if, for all i, the right-hand side of (7) is either 0 or of the form $A_{\sigma(k,i)}(n)$ for some $\sigma(k,i)$ (that is, $\alpha_{i,j}^{(k)} = 0$ if $j \neq \sigma(k,i)$ and $\alpha_{i,j}^{(k)} = 1$ if $j = \sigma(k,i)$). As indicated in Example 2.2 below, an automatic p-scheme is equivalent to a finite state automaton describing the sequence $A(n)$. Note that we find it convenient in practice to allow 0 as a right-hand side of (7) though one could certainly disallow this possibility at the potential cost of introducing an additional state representing the zero sequence.

Example 2.2. The 3-scheme in Example 2.1 is not automatic (if it were then, for instance, the right-hand side of $A_0(3n) = A_0(n) + A_1(n)$ would have to equal one of $A_0(n)$, $A_1(n)$, or 0; neither of these is the case as we can easily see directly or by computing the first few terms). However, at the cost of increasing the number of states from two to four, an equivalent automatic 3-scheme for the Catalan numbers $C(n)$ modulo 3 can be obtained as:

$$\begin{aligned} A_0(3n) &= A_1(n) & A_2(3n) &= A_3(n) \\ A_0(3n+1) &= A_1(n) & A_2(3n+1) &= 0 \\ A_0(3n+2) &= A_2(n) & A_2(3n+2) &= A_2(n) \\ A_1(3n) &= A_1(n) & A_3(3n) &= A_3(n) \\ A_1(3n+1) &= A_3(n) & A_3(3n+1) &= A_1(n) \\ A_1(3n+2) &= 0 & A_3(3n+2) &= 0 \end{aligned}$$

with initial conditions

$$A_0(0) = 1, \quad A_1(0) = 1, \quad A_2(0) = 2, \quad A_3(0) = 2.$$

The corresponding finite state automaton matches Figure 1 from Example 1.2, where A_0 is the initial node, A_1 is the top-right node, A_2 is the bottom-left node, and A_3 the bottom-right node. We note that the finite state automaton features a fifth node that explicitly represents the zero state (slight caution is needed when referring to the number of states as these might differ by one: the p-scheme has four states while the corresponding automaton has five states).

2.2 Scaling schemes

As somewhat illustrated by Examples 2.1 and 2.2, linear schemes typically require substantially fewer states than corresponding automatic schemes, which can make them considerably less costly to compute. On the other hand, automatic schemes have the advantage of typically being much easier to analyze. For instance, an automatic scheme makes it trivial to determine which values are obtained by the underlying sequence: namely, these are precisely the initial conditions (assuming that each node in the corresponding finite state automaton is reachable from the initial node, which is always the case when following the construction in [20] which is summarized below). On the other hand, it can be computationally expensive to extract this information from a linear scheme.

Aiming to combine the benefits of automatic and linear schemes, we consider schemes with the property that, for all i, the right-hand side of (7) consists of at most one term (that is, for each k and i, there is at most one j such that $\alpha_{i,j}^{(k)} \neq 0$). We refer to these as *scaling schemes*.

Example 2.3. Continuing Examples 2.1 and 2.2, the following defines a scaling 3-scheme for the Catalan numbers $C(n)$ modulo 3:

$$\begin{aligned} A_0(3n) &= A_1(n) & A_1(3n) &= A_1(n) & A_2(3n) &= A_1(n) \\ A_0(3n+1) &= A_1(n) & A_1(3n+1) &= 2A_1(n) & A_2(3n+1) &= 0 \\ A_0(3n+2) &= 2A_2(n) & A_1(3n+2) &= 0 & A_2(3n+2) &= A_2(n) \end{aligned}$$

with initial conditions

$$A_0(0) = 1, \quad A_1(0) = 1, \quad A_2(0) = 1.$$

We observe that it is straightforward to convert from a scaling scheme to an automatic one, and vice versa. To wit, let B_0, B_1, B_2, B_3 denote the four states of the automatic scheme from Example 2.2. Then $B_0 = A_0$, $B_1 = A_1$, $B_2 = 2A_2$, $B_3 = 2A_1$.

2.3 An algorithm to compute congruence schemes

Rowland and Zeilberger [20] offer the following algorithm to produce linear p-schemes for a sequence $A(n)$ represented as the constant terms $A(n) = \mathrm{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$ defined over $R = \mathbb{Z}/p^r$, the integers modulo p^r. Since we are working over the ring R, all corresponding equalities below are to be understood as congruences modulo p^r.

Starting with the state $A_0(n) = \mathrm{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$, we iteratively build a collection $A_0, A_1, \ldots$ of states $A_i(n) = \mathrm{ct}[P_i(\boldsymbol{x})^n Q_i(\boldsymbol{x})]$ as follows. For each state A_i and for each $k \in \{0, 1, \ldots, p-1\}$, we either express $A_i(pn+k)$ in terms of existing states or we add a new state. More precisely, to begin with, we write

$$A_i(pn+k) = \mathrm{ct}[\hat{P}(\boldsymbol{x})^n \hat{Q}(\boldsymbol{x})], \tag{8}$$

with $\hat{P}, \hat{Q}$ obtained as follows: let $\tilde{P}(\boldsymbol{x}) = P_i(\boldsymbol{x})^p$ and $\tilde{Q}(\boldsymbol{x}) = P_i(\boldsymbol{x})^k Q_i(\boldsymbol{x})$. If $\tilde{P}(\boldsymbol{x}) = \hat{P}(\boldsymbol{x}^p)$ for some $\hat{P}$, then, using (6),

$$\begin{aligned} A_i(pn+k) &= \mathrm{ct}[P_i(\boldsymbol{x})^{pn} P_i(\boldsymbol{x})^k Q_i(\boldsymbol{x})] = \mathrm{ct}[\tilde{P}(\boldsymbol{x})^n \tilde{Q}(\boldsymbol{x})] \\ &= \mathrm{ct}[\hat{P}(\boldsymbol{x}^p)^n \tilde{Q}(\boldsymbol{x})] = \mathrm{ct}[\hat{P}(\boldsymbol{x})^n \Lambda_p[\tilde{Q}(\boldsymbol{x})]] = \mathrm{ct}[\hat{P}(\boldsymbol{x})^n \hat{Q}(\boldsymbol{x})] \end{aligned}$$

for $\hat{Q}(\boldsymbol{x}) = \Lambda_p[\tilde{Q}(\boldsymbol{x})]$. Otherwise, we let $\hat{P} = \tilde{P}$ and $\hat{Q} = \tilde{Q}$. If the right-hand side of (8), that is $\mathrm{ct}[\hat{P}(\boldsymbol{x})^n \hat{Q}(\boldsymbol{x})]$, can be written as a linear combination of existing states $A_j(n)$ (of the form $A_j(n) = \mathrm{ct}[\hat{P}(\boldsymbol{x})^n Q_j(\boldsymbol{x})]$), then we move on to the next value of k (or to the next state A_{i+1}). On the other hand, if $\mathrm{ct}[\hat{P}(\boldsymbol{x})^n \hat{Q}(\boldsymbol{x})]$ cannot be written as a linear combination of existing states $A_j(n)$, then we add $\mathrm{ct}[\hat{P}(\boldsymbol{x})^n \hat{Q}(\boldsymbol{x})]$ as a new state to our collection of states. In either case, $A_i(pn+k)$ can now be expressed as in (7) as a linear combination of states. If this algorithm terminates, it therefore results in a linear p-scheme.

To see that the algorithm always terminates, first note that there are at most r different polynomials P_i (which, by construction, are essentially of the form $P(\boldsymbol{x})^{p^s}$ for some s) involved in the states $A_i(n) = \mathrm{ct}[P_i(\boldsymbol{x})^n Q_i(\boldsymbol{x})]$ because, for any Laurent polynomial $F \in \mathbb{Z}[\boldsymbol{x}^{\pm 1}]$,

$$F(\boldsymbol{x})^{p^r} \equiv F(\boldsymbol{x}^p)^{p^{r-1}} \pmod{p^r}. \tag{9}$$

Congruence (9) is well-known (see, for instance, [19, Proposition 1.9]). Second, as Rowland and Zeilberger [20] indicate, the degree (and low-degree) of the polynomials Q_i can be bounded, so that there are only finitely many possible states. We work out explicit bounds on the Q_i in Theorem 2.4 below.

Before doing so, we observe that essentially the same algorithm works to compute automatic as well as scaling schemes. Indeed, to obtain an automatic scheme, instead of checking whether (8) can be written as a linear combination of existing states, we only check whether (8) is equal to an existing state (and if it isn't, we add (8) as a new state). Likewise, to obtain a scaling scheme, we check whether (8) is equal to a multiple of an existing state. In either case, the algorithm is guaranteed to terminate for the same reason: namely, that there are only finitely many possible states.

2.4 Bounding the number of states

Let $\mathrm{dg} : R[\boldsymbol{x}^{\pm 1}] \to \mathbb{Z}_{\geq 0}$ denote any integer-valued degree-like function on Laurent polynomials, by which we mean that, for any $P, Q \in R[\boldsymbol{x}^{\pm 1}]$,

$$\mathrm{dg}(PQ) \leq \mathrm{dg}(P) + \mathrm{dg}(Q), \quad \mathrm{dg}(\Lambda_p[Q]) \leq \frac{\mathrm{dg}(Q)}{p}.$$

For instance, $\mathrm{dg}(Q)$ could be the total degree of Q, or $\mathrm{dg}(Q)$ could be the degree (or low-degree) with respect to any particular variable. We note that bounds similar to those in the next result are derived by Rowland and Yassawi [19] in the case of constructing automatic p-schemes for diagonals of rational functions.

Theorem 2.4. *The above construction of a p-scheme (whether automatic, scaling or linear) for $A_0(n) = \mathrm{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$ modulo p^r results in the states $A_i(n) = \mathrm{ct}[P_i(\boldsymbol{x})^n Q_i(\boldsymbol{x})]$ with at most r choices for $P_i(\boldsymbol{x})$. Moreover, we have*

$$\mathrm{dg}(Q_i) \leq p^{r-1} a - 1 + \max(0, b - a + 1),$$

where $a = \mathrm{dg}(P)$ *and* $b = \mathrm{dg}(Q)$.

Proof. As mentioned earlier, it is a consequence of congruence (9) that there are at most r different polynomials P_i. By construction, each state $A_i(n) = \mathrm{ct}[P_i(\boldsymbol{x})^n Q_i(\boldsymbol{x})]$ with $i \geq 1$ is obtained as

$$A_i(n) = A_0(p^s n + k)$$

for some $s \geq 1$ and some $k \in \{0, 1, \dots, p^s - 1\}$. If $s < r$, then

$$A_0(p^s n + k) = \mathrm{ct}[P(\boldsymbol{x})^{p^s n} P(\boldsymbol{x})^k Q(\boldsymbol{x})],$$

in which case

$$\mathrm{dg}(Q_i) \leq \mathrm{dg}(P(\boldsymbol{x})^k Q(\boldsymbol{x})) \leq ka + b \leq (p^{r-1} - 1)a + b = p^{r-1}a + (b - a). \tag{10}$$

On the other hand, if $s \geq r$, then it follows from (9) that

$$P(\boldsymbol{x})^{p^s} \equiv P(\boldsymbol{x}^{p^{s-r+1}})^{p^{r-1}} \pmod{p^r}$$

and, hence,

$$\begin{aligned} A_0(p^s n + k) &= \mathrm{ct}[P(\boldsymbol{x})^{p^s n} P(\boldsymbol{x})^k Q(\boldsymbol{x})] \\ &\equiv \mathrm{ct}[P(\boldsymbol{x}^{p^{s-r+1}})^{p^{r-1} n} P(\boldsymbol{x})^k Q(\boldsymbol{x})] \pmod{p^r} \\ &= \mathrm{ct}[P(\boldsymbol{x})^{p^{r-1} n} \Lambda_p^{s-r+1}[P(\boldsymbol{x})^k Q(\boldsymbol{x})]]. \end{aligned}$$

In particular, in this case,

$$\begin{aligned} \mathrm{dg}(Q_i) &\leq \mathrm{dg}(\Lambda_p^{s-r+1}[P(\boldsymbol{x})^k Q(\boldsymbol{x})]) \\ &\leq \frac{ka + b}{p^{s-r+1}} \leq \frac{(p^s - 1)a + b}{p^{s-r+1}} = p^{r-1}a + \frac{b - a}{p^{s-r+1}} \leq \begin{cases} p^{r-1}a + \frac{b-a}{p}, & \text{if } b \geq a, \\ p^{r-1}a - 1, & \text{if } b < a. \end{cases} \end{aligned} \tag{11}$$

Combining (10) and (11), we obtain the claimed bound. $\square$

We note that in the special case $r = 1$, where we are working modulo a prime p, the degree bounds are independent of p.

Corollary 2.5. *The above construction of a p-scheme for* $\mathrm{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$ *modulo p results in states* $A_i(n) = \mathrm{ct}[P(\boldsymbol{x})^n Q_i(\boldsymbol{x})]$ *with*

$$\mathrm{dg}(Q_i) \leq \max(\mathrm{dg}(P) - 1, \mathrm{dg}(Q)).$$

Proof. This is the special case $r = 1$ of Theorem 2.4. (Note that in this case $P_i(\boldsymbol{x}) = P(\boldsymbol{x})$ because $P(\boldsymbol{x})^{pn} \equiv P(\boldsymbol{x}^p)^n$ modulo p.) $\square$

Example 2.6. Suppose that we want to compute an automatic 2-scheme for the Motzkin numbers modulo 2. By (4), $M(n) = \mathrm{ct}[P(x)^n Q(x)]$ for $P(x) = x^{-1} + 1 + x$ and $Q(x) = 1 - x^2$. Hence, choosing dg to be the usual degree in Theorem 2.4, we have $a = 1$ and $b = 2$. On the other hand, choosing dg to be the low-degree, we have $a = 1$ and $b = 0$. We thus obtain the bounds

$$\deg(Q_i) \leq 2, \quad \text{low-deg}(Q_i) \leq 0.$$

Therefore, all states are of the form $A_i(n) = \mathrm{ct}[P(x)^n Q_i(x)]$ with $Q_i = \alpha_i + \beta_i x + \gamma_i x^2$ for some $\alpha_i, \beta_i, \gamma_i \in \{0, 1\}$. In particular, we know *a priori* that the desired 2-scheme can have at most $2^3 = 8$ states. In fact, as made explicit in [20], there exists such a scheme with 4 states. (In [20], the computation is performed using 30 as an upper bound for the maximum number of acceptable states. The general bounds discussed here show that we can confidently proceed without imposing an upper bound during the construction of the congruence scheme.)

Example 2.7. Likewise, for computing an automatic p-scheme for the Motzkin numbers modulo p^r, we find that

$$\deg(Q_i) \leq p^{r-1} + 1, \quad \text{low-deg}(Q_i) \leq p^{r-1} - 1. \tag{12}$$

In fact, as described in more detail in Example 2.8 below, the symmetry between x and x^{-1} in $P(x) = x^{-1} + 1 + x$ makes it possible to replace x^{-1} by x in Q and to thus choose the Q_i such that $\text{low-deg}(Q_i) = 0$. Hence, all states can be expressed as $A_i(n) = \mathrm{ct}[P_i(x)^n Q_i(x)]$ with at most r possibilities for P_i as well as Q_i with degree at most $p^{r-1} + 1$ and low-degree 0. This implies that there is an automatic p-scheme with at most $r \cdot (p^r)^{p^{r-1}+2} = r \cdot p^{r(p^{r-1}+2)}$ many states.

We can slightly improve this bound by observing that "most" states involve the P_i of highest degree. Indeed, it follows from (9) that, in the absence of further simplification, each polynomial P_i is one of $P(x)^{p^j}$ for $j \in \{0, 1, \ldots, r-1\}$. In that case, the states involving $P(x)^{p^j}$ for $j < r-1$ correspond to $A_0(p^j n + k)$ for some $k \in \{0, 1, \ldots, p^j - 1\}$, while all other states involve $P(x)^{p^{r-1}}$. In particular, there are at most p^j many states involving $P(x)^{p^j}$ for $j < r-1$. Therefore, there is an automatic p-scheme for the Motzkin numbers modulo p^r with at most

$$1 + p + p^2 + \cdots + p^{r-2} + p^{r(p^{r-1}+2)} = \frac{p^{r-1} - 1}{p - 1} + p^{r(p^{r-1}+2)} \tag{13}$$

many states, improving the earlier bound of $r \cdot p^{r(p^{r-1}+2)}$.

For instance, for the Motzkin numbers modulo 4 this means there is an automatic scheme with at most $1 + 2^8 = 257$ states. However, there exists such a scheme with only 14 states. In general, while the bounds for $\deg(Q_i)$ appear to be sharp, the resulting doubly-exponential bounds on the total number of states are far from effective. For the minimal numbers of states for small r, we refer to Example 4.4.

In the case of linear schemes, the above considerations imply that there is a linear p-scheme for the Motzkin numbers modulo p^r with at most

$$(1 + p + p^2 + \cdots + p^{r-2}) + (p^{r-1} + 2) = \frac{p^r - 1}{p - 1} + 2 \tag{14}$$

many states. In particular, there is a linear 2-scheme for the Motzkin numbers modulo 2^r with at most $2^r + 1$ many states. We note that the bounds (14) confirm weaker bounds conjectured by Henningsen [14] for $p = 2$ and $p = 3$.

In the same spirit, for scaling schemes, one can derive bounds for the maximum number of required states which are lower than (13) (the improved bounds are a bit worse than (13) divided by p^r) but significantly higher than (14). As in the case of automatic schemes, these bounds appear to not be effective. It would be of considerable interest to obtain sharper bounds for automatic and scaling schemes, even if restricted to certain families of constant terms.

Example 2.8. Recall from (4) that the Motzkin numbers have the constant term representation $M(n) = \mathrm{ct}[(x^{-1}+1+x)^n(1-x^2)]$. When computing a 2-scheme for $M(n)$ modulo 4, we obtain, for instance,

$$\begin{aligned} M(2n+1) &= \mathrm{ct}[(x^{-1}+1+x)^{2n}(x^{-1}+1+x)(1-x^2)] \\ &= \mathrm{ct}[(x^{-1}+1+x)^{2n}(x^{-1}+1-x^2-x^3)]. \end{aligned}$$

Note that $x^{-1}+1-x^2-x^3$ has degree 3 and low-degree 1 consistent with (12) for $p=2$, $r=2$. On the other hand, the symmetry between x and x^{-1} in $P(x) = x^{-1}+1+x$ implies that $\mathrm{ct}[(x^{-1}+1+x)^{2n}x^{-1}] = \mathrm{ct}[(x^{-1}+1+x)^{2n}x]$ so that

$$M(2n+1) = \mathrm{ct}[(x^{-1}+1+x)^{2n}(1+x-x^2-x^3)].$$

This observation allows us to write all states in the form $\mathrm{ct}[P_i(x)^nQ_i(x)]$ with low-deg$(Q_i)=0$. Note that this observation similarly applies to the computation of p-schemes for any constant term $A(n) = \mathrm{ct}[P(\boldsymbol{x})^nQ(\boldsymbol{x})]$ modulo any p^r provided that there are symmetries in $P(\boldsymbol{x})$ among the variables $\boldsymbol{x}^{\pm 1}$. Basic such symmetries are automatically taken into account by the implementation discussed in Section 4.

Example 2.9. Recall from (3) that the Catalan numbers have the constant term representation $C(n) = \mathrm{ct}[P(x)^nQ(x)]$ with $P(x) = x^{-1}+2+x$ and $Q(x) = 1-x$. Proceeding as in Example 2.7, we find the corresponding bounds $\deg(Q_i) \le p^{r-1}$ and low-deg$(Q_i)=0$. Consequently, there is a linear p-scheme for the Catalan numbers modulo p^r with at most

$$(1+p+p^2+\cdots+p^{r-2}) + (p^{r-1}+1) = \frac{p^r-1}{p-1}+1 \tag{15}$$

many states. In particular, in the case $r=1$, we conclude that there is a linear p-scheme for the Catalan numbers modulo p with 2 states. These schemes are made explicit in [15] where they are interpreted as generalized Lucas congruences.

For $p=2$ and $r>1$, the bound (15) can be improved by the observation $P(x) = (x^{-1/2}+x^{1/2})^2$ which implies that

$$P(x)^{2^{r-1}} \equiv P(x^2)^{2^{r-2}} \pmod{2^r},$$

while, by (9), this congruence only holds modulo 2^{r-1} for general $P(x)$. Using this congruence in place of (9) in the proof of Theorem 2.4, we find that it is possible to express every state $\mathrm{ct}[P_i(x)^nQ_i(x)]$ so that each P_i is one of $P(x)^{2^j}$ for $j \in \{0,1,\ldots,r-2\}$ and so that the stronger bound $\deg(Q_i) \le 2^{r-2}$ holds. Accordingly, if $r>1$, there is a linear 2-scheme for the Catalan numbers modulo 2^r with at most

$$(1+2+2^2+\cdots+2^{r-3}) + (2^{r-2}+1) = 2^{r-1} \tag{16}$$

many states. The bounds (15) and (16) confirm weaker bounds conjectured by Henningsen [14].

It would be of interest to determine whether the bounds (14) as well as (15) and (16) for the number of states of linear p-schemes can be further improved.

3 Schemes for p-adic valuations and applications

3.1 Computing schemes for p-adic valuations

As usual, the p-adic valuation of a nonzero integer c, denoted by $\nu_p(c)$, is the largest r such that p^r divides c. Suppose that a sequence $A(n)$ is such that its values modulo p^r are p-automatic (which includes any sequence that can be represented using constant terms). Rowland and Yassawi [19] observe that, if $A(n)$ is not divisible by arbitrarily large powers of p, then the sequence of p-adic valuations of $A(n)$ is p-automatic as well. Indeed, if $\nu_p(A(n)) \leq r$ for all n, then an automatic p-scheme for $\nu_p(A(n))$ can be easily obtained from an automatic p-scheme for $A(n)$ modulo p^r.

Moreover, it is not hard to see that, along the same lines, a (scaling) p-scheme for $\nu_p(A(n))$ can be obtained from a scaling p-scheme for $A(n)$ modulo p^r. Namely, suppose we have a scaling p-scheme for a sequence $A(n)$ modulo p^r. Let A_i be the states of this scheme. By construction, each transition is of the form

$$A_i(pn + k) \equiv \alpha_i^{(k)} A_{\sigma(i,k)}(n) \pmod{p^r}.$$

Replacing each transition factor $\alpha = \alpha_i^{(k)}$ with $p^{\nu_p(\alpha)}$, and likewise replacing each initial condition, we obtain a p-scheme that computes $p^{\nu_p(A(n))}$ modulo p^r. If $\nu_p(A(n)) \leq r$, the values $p^{\nu_p(A(n))}$ modulo p^r are in one-to-one correspondence with the values of $\nu_p(A(n))$, so that this scaling p-scheme characterizes the p-adic valuation of $A(n)$.

If we make the reasonable assumption that, in the algorithm described in Section 2.3, the cost of checking whether (8) is an existing state is essentially equal to the cost of checking whether (8) is a multiple of an existing state, then computing a scaling scheme is at least as fast as computing an automatic scheme. On the other hand, in many practical examples, such as the one described in Section 3.3, computing a scaling scheme is considerably faster and results in schemes with significantly reduced numbers of states. This makes scaling schemes particularly well suited for the purpose of computing schemes that describe p-adic valuations.

3.2 Reproving a conjecture on Motzkin numbers modulo 8

As an examplary application, we reprove the following result that was conjectured by Amdeberhan, Deutsch and Sagan [10, Conjecture 5.5] and (much more laboriously) proven by Eu, Liu and Yeh [11].

Theorem 3.1. *The* 2*-adic valuation of the Motzkin numbers* $M(n)$ *is*

$$\nu_2(M(n)) = \begin{cases} 2, & \text{if } n = (4i+1)4^{j+1} - 1 \text{ or } n = (4i+3)4^{j+1} - 2 \text{ with } i, j \in \mathbb{Z}_{\geq 0}, \\ 1, & \text{if } n = (4i+1)4^{j+1} - 2 \text{ or } n = (4i+3)4^{j+3} - 1 \text{ with } i, j \in \mathbb{Z}_{\geq 0}, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. We begin by following the approach of Rowland and Yassawi [19] who computed a finite state automaton representing Motzkin numbers modulo 8 and used it to conclude that no Motzkin number is 0 modulo 8. In particular, this implies $\nu_2(M(n)) < 3$ so that, by the argument given in Section 3.1, Rowland and Yassawi were able to conclude that the sequence of 2-adic valuations of Motzkin numbers is 2-automatic. Indeed, they provided a corresponding finite state automaton with 17 states in [19, Figure 5]. Theorem 3.1 could be derived by a careful analysis of this automaton.

However, we can slightly simplify this automaton (as well as the ensuing analysis) as follows. Starting with the constant term representation (4) of the Motzkin numbers, we use our implementation to compute the simplified finite state automaton with 10 states for $\nu_2(M(n))$ depicted in Figure 2 (see Example 4.10 for the details on this automatic computation).

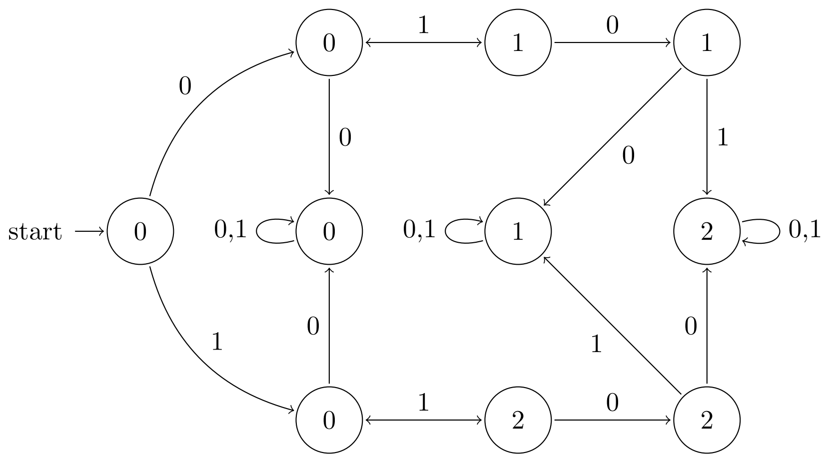

Figure 2: Congruence automaton for 2-adic valuations of the Motzkin numbers

We claim that the automaton in Figure 2 contains the same information as the formula in Theorem 3.1. In the sequel, we will give the details for the case $\nu_2(M(n)) = 2$, and omit those for $\nu_2(M(n)) = 1$ because the argument is the same.

The n with $\nu_2(M(n)) = 2$ are those with a binary expansion that when fed into the automaton in Figure 2 ends up in a state with label 2. Inspection of the automaton reveals that one way (namely moving along the upper part of the automaton) of ending up in a state with label 2 is to begin with 0, then 1, followed by $2j$ times the digit 1 where $j \geq 0$ is arbitrary, followed by 0, then 1, followed by any further sequence of digits. Suppose that the final further sequence of digits by itself represents the number i. Further suppose that the string of $2j+4$ digits preceding i (namely, $01^{2j+1}01$) represents the number $2^{2j+4} - 2^{2j+2} - 2 = 3 \cdot 4^{j+1} - 2$. Then the overall string of digits represents the number

$$n = 3 \cdot 4^{j+1} - 2 + 2^{2j+4} \cdot i = (4i+3)4^{j+1} - 2,$$

matching one of the two possibilities listed in the claimed formula. In the same manner, the other possibility, namely $n = (4i+1)4^{j+1} - 1$, corresponds to moving along the bottom part of the automaton to end up in a state with label 2 (and it is clear from the automaton that there is no further way of ending up in a state with label 2). $\square$

3.3 A conjecture on Motzkin numbers modulo p^2

The simple observation that scaling p-schemes are suitable for computing a p-scheme for the p-adic valuations of a sequence helps make computations feasible that were previously out of reach. We illustrate this in the case of an interesting open question posed by Rowland and Yassawi [19], which asks whether there exist infinitely many primes p such that $M(n) \not\equiv 0 \pmod{p^2}$ for all $n \in \mathbb{Z}_{\geq 0}$. By computing congruence schemes modulo 5^2 and 13^2, Rowland and Yassawi showed that $p = 5$ and $p = 13$ are two primes with this property and offered the following conjecture:

Conjecture 3.2 ([19, Conjecture 3.10])**.** *Let $p \in \{31, 37, 61\}$. For all $n \in \mathbb{Z}_{\geq 0}$, $M(n) \not\equiv 0 \pmod{p^2}$.*

We prove this conjecture and extend it to include three further cases.

Theorem 3.3. *Let $p \in \{5, 13, 31, 37, 61, 79, 97, 103\}$. For all $n \in \mathbb{Z}_{\geq 0}$, $M(n) \not\equiv 0 \pmod{p^2}$. For any other prime $p < 200$, there exists n such that p^2 divides $M(n)$.*

Proof. Let p be any prime number. Using the constant term representation (4), we proceed by computing a scaling p-scheme for the Motzkin numbers $M(n)$ modulo p^2. As described above, we then use this scheme to construct a (considerably simpler) p-scheme for $p^{\nu_p(M(n))}$ modulo p^2. Inspection of that scheme makes it straightforward to test whether there exists an index n such that $p^{\nu_p(M(n))} \equiv 0$ modulo p^2. The latter is equivalent to testing whether $M(n) \equiv 0$ modulo p^2. We implemented this approach in the computer algebra system Sage, described in Section 4, and carried out the computations for all primes $p < 200$. Further details of this computation are included in Example 4.11. □

As noted above, the cases $p = 5$ and $p = 13$ of Theorem 3.3 were already established in [19, Theorem 3.8 & 3.9]. Rowland and Yassawi report that the computation modulo 13^2 took about 40 minutes. On our basic laptop, Rowland's impressive and more recent implementation [18] reduces this time to about 2.5 minutes when using diagonals as in [19] and further to 30 seconds when using constant terms as in [20]. On the other hand, the computation described in the proof of Theorem 3.3 using a scaling 13-scheme only requires about half a second on the same laptop. Since performing our calculations, we have further learned that Rowland has independently established the cases $p = 31$ and $p = 37$ of Theorem 3.3 by using [18] to compute automatic p-schemes for $M(n)$ modulo p^2. These automatic p-schemes are rather complex with 28,081 and 44,173 states, respectively. On the other hand, the corresponding scaling p-schemes in our computation only have 125 and 149 states, respectively, making it possible to calculate them in less than a minute. It is this reduction, which becomes more pronounced as the size of p increases, in the number of states when using scaling over automatic schemes that made it feasible to compute p-schemes for $M(n)$ modulo p^2 for all primes below 200 (that arbitrary limit could be pushed further but we hope that it suffices to convince the reader of the utility of computing scaling schemes).

Example 3.4. For $p = 83$, the first Motzkin number that is divisible by p^2 is

$$M(5{,}139{,}193) = 2{,}051{,}827{,}558{,}749, \ldots\ldots, 008{,}702{,}105{,}903,$$

where the right-hand side is an integer with 2,452,009 decimal digits. This indicates the difficulty of predicting based on initial terms whether, given a prime p, there is a Motzkin number divisible by p^2. Of course, in the absence of further insight (such as an upper bound), computing initial terms by itself can only identify those primes p for which there exists a Motzkin number divisible by p^2. The computation of an automatic or scaling scheme modulo p^2, on the other hand, straightforwardly settles this question in either case.

It would be of interest to analyze the p-schemes for Motzkin numbers modulo p^2 in hopes of discovering a characterization of those p for which no Motzkin number is divisible by p^2. We do not pursue this here since our focus is on our ability to compute these p-schemes in practice. As we have demonstrated, using scaling p-schemes over automatic ones allows us to compute instances that were previously out of reach.

4 A computer algebra implementation

4.1 Basic usage

In order to perform the computations described in Section 3.3 (which to our knowledge are not within reach of previous implementations), we implemented the algorithm described in Section 2 in the open-source computer algebra system Sage [21]. Usage of this implementation is briefly described in this section. First, however, we note that Rowland and Zeilberger [20] provide an implementation in Maple for computing automatic and linear congruence schemes for the modulo p^r values of constant terms. Moreover, Rowland's powerful Mathematica package *IntegerSequences* [18] offers, among many other tools for working with k-regular sequences, methods for computing finite state automata representing the modulo p^r values of sequences (represented in various ways, including as constant terms or diagonals). A subset of the algorithms of [20] have also been implemented by Joel Henningsen in Sage as part of his master's thesis [14] under the direction of the author. The performance and design lessons learned from Henningsen's work have benefitted the present implementation which is freely available at:

`http://arminstraub.com/congruenceschemes`

To use the package from within a recent version of Sage, we need to import its functionality:

```
>>> from congruenceschemes import *
```

Before turning to more advanced applications, we illustrate the basic usage by showing how the congruence schemes from the introductory Examples 2.1, 2.2 and 2.3 can be computed.

Example 4.1. Using the constant term representation (3) for the Catalan numbers, we can compute a linear 3-scheme for the Catalan numbers $C(n)$ modulo 3 as follows:

```
>>> R.<x> = LaurentPolynomialRing(Zmod(3))
>>> S = CongruenceScheme(1/x+2+x, 1-x); S
  Linear 3-scheme with 2 states over Ring of integers modulo 3
```

The resulting 3-scheme is the one described in Example 2.1. In the implementation, the initial conditions and transitions (spelled out explicitly in Example 2.1) are encoded as follows:

```
>>> S.initial_conds()
  [1, 0]
>>> S.transitions()
  [[{0: 1, 1: 1}, {0: 1, 1: 1}, {0: 2, 1: 1}],
   [{}, {0: 1, 1: 1}, {0: 1, 1: 2}]]
```

Note that the transitions consist of two lists (one on each line in the above output), corresponding to the two states A_0, A_1. Each list has three entries corresponding to the transitions $3n + j$ for $j \in \{0, 1, 2\}$. For instance, the entry `{0: 2, 1: 1}` encodes the transition $A_0(3n+2) = 2A_0(n)+A_1(n)$.

Example 4.2. The automatic 3-scheme from Example 2.2 can be likewise computed:

```
>>> S = CongruenceSchemeAutomatic(1/x+2+x, 1-x)
>>> S.initial_conds()
  [1, 1, 2, 2]
>>> S.transitions()
  [[{1: 1}, {1: 1}, {2: 1}], [{1: 1}, {3: 1}, {}],
   [{3: 1}, {}, {2: 1}], [{3: 1}, {1: 1}, {}]]
```

In contrast to the previous example, we now have four states rather than two. The four lists (two on each line in the above final output) correspond directly to the transitions spelled out in Example 2.2.

Example 4.3. In the same manner, we can compute the three-state scaling 3-scheme from Example 2.3:

```
>>> S = CongruenceSchemeScaling(1/x+2+x, 1-x)
>>> S.initial_conds()
  [1, 1, 1]
>>> S.transitions()
  [[{1: 1}, {1: 1}, {2: 2}], [{1: 1}, {1: 2}, {}], [{1: 1}, {}, {2: 1}]]
```

4.2 Numbers of states

The present implementation tends to produce congruence schemes with fewer states than the Maple implementation accompanying [20] because it implements certain ad-hoc optimizations such as, most notably, the exploitation of symmetry described in Example 2.8. A valuable avenue for future work would be to systematically study and implement further optimizations.

Example 4.4. For instance, automatic 2-schemes for the Motzkin numbers modulo 2^r are computed in [20], for $r \in \{1, 2, \dots, 5\}$, with Table 1 listing the number of states of the resulting schemes.

r	1	2	3	4	5	6	7	8
implementation in [20]	4	24	128	801	5093	$> 10^4$		
present implementation	4	14	24	76	225	701	2810	8090

Table 1: Number of states in automatic schemes modulo 2^r for Motzkin numbers.

Table 1 also lists the number of states of the schemes when computed using our implementation. These numbers can be obtained (in about 90 seconds on a basic laptop) as follows:

```
>>> R.<x> = LaurentPolynomialRing(ZZ)
>>> P, Q = 1/x+1+x, 1-x^2
>>> schemes = [CongruenceSchemeAutomatic(P, Q, p=2, r=r) for r in [1..8]]
>>> [S.nr_states() for S in schemes]
  [4, 14, 24, 76, 225, 701, 2810, 8090]
```

As pointed out in Example 2.2, the corresponding finite state automata may have one additional state (representing 0). The counts for these automata are:

```
>>> [S.nr_states_automaton() for S in schemes]
  [5, 15, 24, 76, 225, 701, 2810, 8090]
```

These counts match the number of states of the finite state automata for Motzkin numbers modulo 2^r that we computed using Rowland's Mathematica package [18]. Indeed, these numbers of states are best possible because both Rowland's Mathematica implementation and our Sage implementation minimize the computed finite state automata in an additional (optional) post-processing phase (our implementation presently employs Moore's algorithm [16] for this purpose).

Question 4.5. *Can we give an exact (or asymptotic) formula for the sequence* 5*,* 15*,* 24*,* 76*,* 225*,* 701*,* 2810*,* 8090*, … of the minimal numbers of states for finite state automata for Motzkin numbers modulo* 2^r*?*

The corresponding question for linear (or scaling) 2-schemes for Motzkin numbers modulo 2^r is equally interesting and, possibly, more natural. In this direction, we recall from Example 2.7 that there is a linear 2-scheme for the Motzkin numbers modulo 2^r with at most 2^r+1 many states (and, for small r, such a scheme can be computed using our implementation). It is natural to wonder whether it is possible to further reduce the number of states needed for these schemes.

In order to investigate such questions systematically, it would be valuable to extend the minimization of automatic schemes to the case of scaling and linear schemes, as well as to analyze the computational cost of doing so. Especially in the case of scaling schemes, Moore's algorithm [16] (and other known minimization algorithms) can likely be adapted for this purpose but we do not pursue this question here.

4.3 Fast evaluation of sequences modulo m

As pointed out by Rowland and Zeilberger [20], one application of congruence schemes is the fast evaluation of the underlying sequence modulo p^r (and, hence, modulo any m by virtue of the Chinese remainder theorem).

Example 4.6. As an example, it is shown in [20] that $M(10^{100})$, the googol-th Motzkin number, is 12 modulo 25. The following confirms this computation:

```
>>> R.<x> = LaurentPolynomialRing(Zmod(25))
>>> S = CongruenceScheme(1/x+1+x, 1-x^2)
>>> S.nth_term(10^100)
  12
```

Example 4.7. If we are able to evaluate a sequence modulo prime powers in a fast manner, the Chinese remainder theorem allows us to evaluate the sequence modulo any modulus. For illustration, it is computed in [20] in logarithmic time that $M(10^{100}) \equiv 187$ modulo 1000, extending the computation of the previous example. We further extend this computation and determine $M(10^{100})$ modulo 10^5:

```
>>> R.<x> = LaurentPolynomialRing(ZZ)
>>> S2 = CongruenceScheme(1/x+1+x, 1-x^2, p=2, r=5)
>>> S5 = CongruenceScheme(1/x+1+x, 1-x^2, p=5, r=5)
>>> S2.nth_term(10^100).crt(S5.nth_term(10^100))
  27187
```

Accordingly, the last five decimal digits of the googol-th Motzkin number are 27187. These computations took about a minute, with all but a fraction of a second spent on the computation of the congruence scheme modulo 5^5.

4.4 Determining forbidden residues

Rowland and Yassawi [19] give several intriguing examples of sequences that avoid certain residues modulo p^r. Such results are often rather hard to obtain by hand but are automatic to prove by computing an automatic p-scheme for the sequence of interest modulo p^r (or can be deduced with a little more effort from a scaling p-scheme). Here, we restrict ourselves to reproducing, and in one case slightly extending, two of these results using our implementation.

Example 4.8. Chowla, J. Cowles and M. Cowles [6] conjectured, and Gessel [12] proved, that the Apéry numbers (5) associated to $\zeta(3)$ are periodic modulo 8 alternating between the values 1 and 5. Based on the constant term representation (5), the following confirms that the Apéry numbers $A(n)$ modulo 8 only take the values 1 and 5:

```
>>> R.<x,y,z> = LaurentPolynomialRing(ZZ)
>>> P = ((x+y)*(1+z)*(x+y+z)*(1+y+z))/x/y/z
>>> S = CongruenceSchemeAutomatic(P, p=2, r=3); S
  Linear 2-scheme with 3 states over Ring of integers modulo 8
>>> S.possible_values()
  {1, 5}
```

Moreover, as is done in [19], an inspection of the (particularly simple) automaton immediately reveals that $A(2n) \equiv 1$ and $A(2n+1) \equiv 5$ modulo 8.

In general, however, as is illustrated by the next example, no such simple characterizations of the values modulo p^r are possible. Still, automatic (or scaling) congruence schemes can readily be used to determine exactly which residues modulo p^r are attained by a given sequence.

Example 4.9. Rowland and Yassawi [19] observe that certain residues modulo 2^r are never attained by the Catalan numbers $C(n)$. For instance:

- $C(n) \not\equiv 3 \pmod{4}$,
- $C(n) \not\equiv 9 \pmod{16}$,
- $C(n) \not\equiv 17, 21, 26 \pmod{32}$,
- $C(n) \not\equiv 10, 13, 33, 37 \pmod{64}$.

These results can be confirmed with the following computation:

```
>>> R.<x> = LaurentPolynomialRing(Zmod(2^6))
>>> S = CongruenceSchemeScaling(1/x+2+x, 1-x)
>>> S.impossible_values()
  {3, 7, 9, 10, 11, 13, 15, 17, 19, 21, 23, 25, 26, 27, 31, 33, 35, 37, 39,
  41, 43, 47, 49, 51, 53, 55, 57, 58, 59, 63}
>>> len(S.impossible_values())
  30
```

This shows, in particular, that the Catalan numbers do not attain $30/2^6 = 46.875\%$ of the residues modulo 2^6. Based on the corresponding computations modulo 2^r for $r \leq 9$, Rowland and Yassawi [19] pose the question whether the proportion of residues that are not attained by the Catalan numbers modulo 2^r tends to 1 as $r \to \infty$. The proportions for $r \leq 14$ are recorded in Table 2 labeled as $P(r)$. Further listed are the total number $N(r)$ of residues not attained by the Catalan numbers modulo 2^r as well as the number $A(r) = N(r) - 2N(r-1)$ of additional residues not attained (observe that, if $C(n) \not\equiv a$ modulo 2^{r-1}, then we necessarily have $C(n) \not\equiv a$ modulo 2^r; $A(r)$ counts those residues not covered by this observation; for instance, $A(6) = 4$ corresponding to the residues $10, 13, 33, 37$ not attained modulo 2^6, as listed above).

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$P(r)$	0	.25	.25	.31	.41	.47	.54	.59	.65	.69	.73	.76	.79	.82
$N(r)$	0	1	2	5	13	30	69	152	332	710	1502	3133	6502	13,394
$A(r)$	0	1	0	1	3	4	9	14	28	46	82	129	236	390

Table 2: The proportions and numbers of residues not attained by Catalan numbers modulo 2^r.

The values for $r \leq 9$ match those computed by Rowland and Yassawi [19], while we computed the new values for $10 \leq r \leq 14$ using our implementation (in about 3 hours).

Rowland and Yassawi [19] further pose the question whether there exist any residues modulo 3^r that are not attained by the Catalan numbers. Proceeding as above, we are able to compute a scaling 3-scheme for the Catalan numbers modulo 3^9 (in about 20 hours). That scheme then allows us to deduce that the Catalan numbers attain all residues modulo 3^9.

4.5 Computing valuation schemes

Let us demonstrate how to compute the automatic 2-scheme for the 2-adic valuation of the Motzkin numbers that we employed in the proof of Theorem 3.1.

Example 4.10. First, we compute an automatic 2-scheme for the Motzkin numbers modulo 8 as follows:

```
>>> R.<x> = LaurentPolynomialRing(Zmod(8))
>>> S = CongruenceSchemeAutomatic(1/x+1+x, 1-x^2); S
  Linear 2-scheme with 24 states over Ring of integers modulo 8
>>> S.impossible_values()
  {0}
```

The output is a scheme with 24 states that certifies that no Motzkin number $M(n)$ is divisible by 8, as conjectured by Amdeberhan, Deutsch and Sagan [10, Conjecture 5.5] and proven by Eu, Liu and Yeh [11] as well as Rowland and Yassawi [19]. On the other hand, every other value modulo 8 is achieved. We then derive from this scheme, as described in Section 3.1, a scheme for $2^{\nu_2(M(n))}$:

```
>>> V = S.valuation_scheme(); V
  Linear 2-scheme with 10 states over Ring of integers modulo 8
>>> V.initial_conds()
  [1, 1, 1, 1, 2, 4, 2, 4, 2, 4]
>>> V.transitions()
```

```
[[{1: 1}, {2: 1}], [{3: 1}, {4: 1}], [{3: 1}, {5: 1}], [{3: 1}, {3: 1}],
 [{6: 1}, {1: 1}], [{7: 1}, {2: 1}], [{8: 1}, {9: 1}], [{9: 1}, {8: 1}],
 [{8: 1}, {8: 1}], [{9: 1}, {9: 1}]]
```

Relabeling the values of the initial conditions from $1, 2, 4$ to $0, 1, 2$, respectively, results in an automatic 2-scheme for the 2-adic valuations of the Motzkin numbers. Indeed, this scheme directly translates into the finite state automaton in Figure 2 (where the four vertically centered states correspond to states $0, 3, 8, 9$ in the above scheme), which we used to prove Theorem 3.1.

As another application of computing valuation schemes, let us demonstrate how to prove Theorem 3.3 for the prime $p = 13$.

Example 4.11. The case $p = 13$ of Theorem 3.3 claims that no Motzkin number is divisible by 13^2. Rowland and Yassawi [19] prove this claim using an automatic 13-scheme for the Motzkin numbers modulo 13^2. To perform this calculation using our implementation we can compute:

```
>>> R.<x> = LaurentPolynomialRing(Zmod(13^2))
>>> S = CongruenceSchemeAutomatic(1/x+1+x, 1-x^2); S
  Linear 13-scheme with 2097 states over Ring of integers modulo 169
>>> S.impossible_values()
  {0}
```

The last output confirms that, indeed, $M(n) \not\equiv 0$ modulo 13^2 for all n.

In principle, the same approach could be used for any prime. However, the above computation, which takes a little over 10 seconds on a typical laptop (a slight improvement on the 30 seconds we needed for the same computation on the same laptop using Rowland's Mathematica implementation [18], which considerably improves on the 40 minutes reported in [19]), for larger primes p quickly becomes impractical even on much more powerful machines. Instead, as described in the proof of Theorem 3.3, we first compute a scaling 13-scheme for the Motzkin numbers modulo 13^2, which only takes about half a second:

```
>>> S = CongruenceSchemeScaling(1/x+1+x, 1-x^2); S
  Linear 13-scheme with 48 states over Ring of integers modulo 169
```

We could again determine the impossible values from here but, especially for larger primes, it is more efficient to derive from the above scheme a scheme for $13^{\nu_{13}(M(n))}$ modulo 13^2:

```
>>> V = S.valuation_scheme(); V
  Linear 13-scheme with 5 states over Ring of integers modulo 169
>>> V.possible_values()
  {1, 13}
```

The final output certifies that $13^{\nu_{13}(M(n))}$ only takes the values 1 or 13 modulo 13^2. Accordingly, $M(n) \not\equiv 0$ modulo 13^2.

To prove Theorem 3.3, we performed these computations for all primes $p < 200$. As indicated, for larger primes it becomes computationally imperative to initially compute a scaling (rather than an automatic) p-scheme for the Motzkin numbers modulo p^2. For $p = 61$, the first previously open case, the computation took about 10 minutes on a basic laptop, while the case $p = 197$ required about 3 days of computation.

5 Conclusion

For the sake of exposition, we have focused on constant term sequences (1) though the general ideas, such as the utility of scaling schemes, apply in the same manner to sequences that are diagonals of rational functions. Constant term sequences, that is, sequences of the form $a(n) = \mathrm{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$ for Laurent polynomials $P, Q \in \mathbb{Z}[\boldsymbol{x}^{\pm 1}]$, can always be expressed as diagonals of rational functions. As Zagier [23, p. 769, Question 2] and Gorodetsky [13] do in the case $Q = 1$, it is therefore natural to ask which diagonal sequences are constant term sequences (1). This appears to be a difficult problem, even for specific sequences. As an initial challenge, we invite the interested reader to consider the following:

Question 5.1. *Can the Fibonacci numbers F_n, the diagonal sequence of $x/(1-x-x^2)$, be expressed as a constant term sequence? That is, are there $P, Q \in \mathbb{Z}[\boldsymbol{x}^{\pm 1}]$ such that $F_n = \mathrm{ct}[P(\boldsymbol{x})^n Q(\boldsymbol{x})]$?*

We observe that the Fibonacci numbers cannot be so expressed with $Q = 1$ (because they fail to satisfy the Gauss congruences [4]).

On the other hand, diagonals of rational functions are somewhat better understood due to recent results by Bostan, Lairez and Salvy [5] who show, among other results, that these can be characterized as sequences expressible as multiple binomial sums.

One of the motivations for being able to efficiently compute congruence schemes is that it enables us to observe new phenomena which would otherwise be more difficult to observe. For instance, even in the case of the very well-studied Catalan numbers, Rowland and Yassawi [19] reveal intriguing new questions by computing congruence schemes. For instance, as indicated in Example 4.9, Rowland and Yassawi [19] pose the question whether the proportion of residues that are not attained by the Catalan numbers modulo 2^r tends to 1 as $r \to \infty$.

Rowland and Yassawi [19] further note that some residues are only attained finitely many times. For instance, $C(n) \not\equiv 1 \pmod{8}$ for $n \geq 2$, and $C(n) \not\equiv 5, 10 \pmod{16}$ for $n \geq 6$. On the other hand, we presently lack the tools to establish similar results modulo m if m is not a prime power. This is illustrated, in the case $m = 10$, by the following conjecture due to Alin Bostan, observed in 2015 and popularized at the 80th Séminaire Lotharingien de Combinatoire in 2018.

Conjecture 5.2 (Bostan, 2015)**.**

(a) For all $n \geq 0$, $C(n) \not\equiv 3 \pmod{10}$.

(b) For sufficiently large n, $C(n) \not\equiv 1, 7, 9 \pmod{10}$.

In particular, this conjecture implies that the last digit of any sufficiently large odd Catalan number is always 5.

Acknowledgements

The author thanks Alin Bostan for interesting discussions on diagonals and constant terms, as well as for sharing the final conjecture. Support through a Collaboration Grant from the Simons Foundation (#514645) is gratefully acknowledged. The author further thanks the anonymous referees for their careful and helpful comments and suggestions.

Data availability statement

All data generated or analysed as part of this work has been produced, and can be reproduced, using the author's Sage package that is freely available at:
`http://arminstraub.com/congruenceschemes`

References

[1] J.-P. Allouche and J. Shallit. The ring of k-regular sequences. *Theoretical Computer Science*, 98(2):163–197, May 1992.

[2] J.-P. Allouche and J. Shallit. *Automatic Sequences: Theory, Applications, Generalizations.* Cambridge University Press, Cambridge, 2003.

[3] R. Apéry. Irrationalité de $\zeta(2)$ et $\zeta(3)$. *Astérisque*, 61:11–13, 1979.

[4] F. Beukers, M. Houben, and A. Straub. Gauss congruences for rational functions in several variables. *Acta Arithmetica*, 184:341–362, 2018.

[5] A. Bostan, P. Lairez, and B. Salvy. Multiple binomial sums. *Journal of Symbolic Computation*, 80.2:351–386, 2017.

[6] S. Chowla, J. Cowles, and M. Cowles. Congruence properties of Apéry numbers. *Journal of Number Theory*, 12(2):188–190, May 1980.

[7] G. Christol. Globally bounded solutions of differential equations. In K. Nagasaka and E. Fouvry, editors, *Analytic Number Theory*, number 1434 in Lecture Notes in Mathematics, pages 45–64. Springer Berlin Heidelberg, Jan. 1990.

[8] G. Christol, T. Kamae, M. M. France, and G. Rauzy. Suites algébriques, automates et substitutions. *Bulletin de la Société Mathématique de France*, 108(4):401–419, 1980.

[9] J. Denef and L. Lipshitz. Algebraic power series and diagonals. *Journal of Number Theory*, 26(1):46–67, May 1987.

[10] E. Deutsch and B. E. Sagan. Congruences for Catalan and Motzkin numbers and related sequences. *Journal of Number Theory*, 117(1):191–215, Mar. 2006.

[11] S.-P. Eu, S.-C. Liu, and Y.-N. Yeh. Catalan and Motzkin numbers modulo 4 and 8. *European Journal of Combinatorics*, 29(6):1449–1466, Aug. 2008.

[12] I. M. Gessel. Some congruences for Apéry numbers. *Journal of Number Theory*, 14(3):362–368, June 1982.

[13] O. Gorodetsky. New representations for all sporadic Apéry-like sequences, with applications to congruences. *Experimental Mathematics*, 2021. DOI:10.1080/10586458.2021.1982080.

[14] J. A. Henningsen. Sequences modulo primes and finite state automata. Master's thesis, University of South Alabama, 2019.

[15] J. A. Henningsen and A. Straub. Generalized Lucas congruences and linear p-schemes. *Preprint*, Nov. 2021. arXiv:2111.08641.

[16] E. F. Moore. Gedanken-experiments on sequential machines. In *Automata studies*, Annals of Mathematics Studies, no. 34, pages 129–153. Princeton University Press, Princeton, N.J., 1956.

[17] A. v. d. Poorten. A proof that Euler missed ... Apéry's proof of the irrationality of $\zeta(3)$. *Mathematical Intelligencer*, 1(4):195–203, 1979.

[18] E. Rowland. IntegerSequences: A package for computing with k-regular sequences. In J. H. Davenport, M. Kauers, G. Labahn, and J. Urban, editors, *Mathematical Software – ICMS 2018*, volume 10931, pages 414–421. Springer, Cham, 2018.

[19] E. Rowland and R. Yassawi. Automatic congruences for diagonals of rational functions. *Journal de Théorie des Nombres de Bordeaux*, 27(1):245–288, 2015.

[20] E. Rowland and D. Zeilberger. A case study in meta-automation: automatic generation of congruence automata for combinatorial sequences. *Journal of Difference Equations and Applications*, 20(7):973–988, 2014.

[21] Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3)*, 2021. `https://www.sagemath.org`.

[22] R. P. Stanley. *Catalan Numbers*. Cambridge University Press, Cambridge, 2015.

[23] D. B. Zagier. The arithmetic and topology of differential equations. In V. Mehrmann and M. Skutella, editors, *Proceedings of the European Congress of Mathematics, Berlin, 18-22 July, 2016*, pages 717–776. European Mathematical Society, 2018.