Electronic Colloquium on Computational Complexity, Report No. 113 (2006)

On defining integers in the counting hierarchy and proving lower bounds in algebraic complexity

Peter Bürgisser*

August 25, 2006

Abstract

Let $\tau(n)$ denote the minimum number of arithmetic operations sufficient to build the integer n from the constant 1. We prove that if there are arithmetic circuits for computing the permanent of n by n matrices having size polynomial in n, then $\tau(n!)$ is polynomially bounded in $\log n$. Under the same assumption on the permanent, we conclude that the Pochhammer-Wilkinson polynomials $\prod_{k=1}^{n}(X-k)$ and the Taylor approximations $\sum_{k=0}^{n}\frac{1}{k!}X^k$ and $\sum_{k=1}^{n}\frac{1}{k}X^k$ of exp and log, respectively, can be computed by arithmetic circuits of size polynomial in $\log n$ (allowing divisions). This connects several so far unrelated conjectures in algebraic complexity.

Key words. algebraic complexity, permanent, factorials, integer roots of univariate polynomials

AMS subject classifications. Primary 68Q17; Secondary 11D45

1 Introduction

The investigation of the complexity to evaluate polynomials by straight-line programs (or arithmetic circuits) is a main focus in algebraic complexity theory. Let the *complexity* $L_K(f)$ of a polynomial $f \in K[X_1,\ldots,X_m]$ over a field K be the minimum number of arithmetic operations $+,-,*,/$ sufficient to compute f from the variables X_i and constants in K. We call a sequence $(f_n)_{n\in\mathbb{N}}$ of univariate polynomials *easy to compute* if $L_K(f_n) = (\log n)^{\mathcal{O}(1)}$, otherwise *hard to compute* (usually n stands for the degree of f_n). For example, the sequence $(G_n^{(r)})_{n\in\mathbb{N}}$ of univariate polynomials over $K=\mathbb{C}$

$$G_n^{(r)} := \sum_{k=1}^{n} k^r X^k$$

*Institute of Mathematics, University of Paderborn, D-33095 Paderborn, Germany. E-mail: pbuerg@upb.de. Partially supported by DFG grant BU 1371 and Paderborn Institute for Scientific Computation (PaSCo).

ISSN 1433-8092

is easy to compute, provided $r \in \mathbb{N}$. This is easily seen by computing the derivatives of the well-known formula $G_n^{(0)} = \frac{X^{n+1}-1}{X-1}$ -1 for the geometric series.

In a landmark paper [22], Strassen proved that various sequences (f_n) of specific polynomials like $f_n = \sum_{k=1}^{n} \exp(2\pi\sqrt{-1}/2^j)$ or $f_n = \sum_{k=1}^{n} 2^{2^k} X^k$ are hard to compute. Von zur Gathen and Strassen [13] showed that the sequence $(G_n^{(r)})$ is hard to compute if $r \in \mathbb{Q} \setminus \mathbb{Z}$. The complexity status of this sequence for negative integers r has ever since been an outstanding open problem, cf. Strassen [24, Problem 9.2]. More details and references on this can be found in [10, Chapter 9].

In 1994 Shub and Smale [20] discovered the following connection between the complexity of univariate integer polynomials and the $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$-hypothesis in the Blum-Shub-Smale model [6] over $\mathbb{C}$. For an integer polynomial $f \in \mathbb{Z}[X_1, \ldots, X_m]$, we define the *tau-complexity* $\tau(f)$ as $L_{\mathbb{Q}}(f)$, but allow only the constant 1 and disallow divisions. Clearly, $L_{\mathbb{Q}}(f) \leq \tau(f)$. The *$\tau$-conjecture* claims the following connection between the number $z(f)$ of distinct integer roots of an univariate $f \in \mathbb{Z}[X]$ and the complexity $\tau(f)$:

$$z(f) \leq (1 + \tau(f))^c \tag{1}$$

for some universal constant $c > 0$ (compare also [24, Problem 9.2]). Shub and Smale [20] proved that the τ-conjecture implies $\mathsf{P}_{\mathbb{C}} \neq \mathsf{NP}_{\mathbb{C}}$. In fact, their proof shows that in order to draw this conclusion, it suffices to prove that for all nonzero integers m_n, the sequence $(m_n n!)_{n\in\mathbb{N}}$ of multiples of the factorials is hard to compute. Hereby we say that a sequence $(a(n))$ of integers is *hard to compute* iff $\tau(a(n))$ is not polynomially bounded in $\log n$.

It is plausible that $(n!)$ is hard to compute, otherwise factoring integers could be done in (nonuniform) polynomial time, cf. [23] or [5, p.126]. Lipton [16] strengthened this implication by showing that if factoring integers is "hard on average" (a common assumption in cryptography), then a somewhat weaker version of the τ-conjecture follows.

Resolving the τ-conjecture appears under the title "Integer zeros of a polynomial of one variable" as the fourth problem in Smale's list [21] of the most important problems for the mathematicians in the 21st century. Our main result confirms the belief that solving this problem is indeed very hard. In fact we prove that the truth of τ-conjecture (as well as a hardness proof for the other problems mentioned before) would imply the truth of another major conjecture in algebraic complexity.

A quarter of a century ago, Valiant [26, 28] proposed an algebraic version of the P versus NP problem for explaining the hardness of computing the permanent. He defined the classes VP of polynomially computable and VNP of polynomially definable families of multivariate polynomials over a fixed field K and proved that the family (PER_n) of permanent polynomials is VNP-complete (if $\mathrm{char}K \neq 2$). We recall that the *permanent* of the matrix $[X_{ij}]_{1\leq i,j\leq n}$ is defined as

$$\mathrm{PER}_n = \sum_{\pi\in S_n} X_{1\pi(1)} \cdots X_{n\pi(n)},$$

where the sum is over all permutations π of the symmetric group. Valiant's completeness result implies that VP $\neq$ VNP iff $(\text{PER}_n) \notin$ VP. The latter statement is equivalent to the the hypothesis that $L_K(\text{PER}_n)$ is not polynomially bounded in n, which is often called *Valiant's hypothesis* over K. (For a detailed account we refer to [7]).

Our main result stated below refers to a somewhat weaker hypothesis claiming that $\tau(\text{PER}_n)$ is not polynomially bounded in n (compare however Corollary 4.2).

Theorem 1.1 *Each of the statements listed below implies that the permanent of n by n matrices cannot be computed by constant-free and division-free arithmetic circuits of size polynomial in n: that is, $\tau(\text{PER}_n)$ is not polynomially bounded in n.*

1. *The sequence of factorials $(n!)_{n\in\mathbb{N}}$ is hard to compute.*
2. *The τ-conjecture of Shub and Smale [20, 4] is true.*
3. *The sequence of Taylor approximations $(\sum_{k=0}^{n} \frac{1}{k!}T^k)_{n\in\mathbb{N}}$ of exp is hard to compute.*
4. *The sequence $(G_n^{(r)}) = (\sum_{k=1}^{n} k^r T^k)_{n\in\mathbb{N}}$ for a fixed negative integer r is hard to compute.*

This result gives some explanation why the attempts to prove the τ-conjecture or the hardness of the above specific sequences of integers or polynomials did not succeed. Astonishingly, the major open problems mentioned in Chapters 9 and 21 of [10] turn out to be closely related!

We remark that Bürgisser [9] proposed a strengthening of the τ-conjecture (L-conjecture) that claims that the number $N_d(f)$ of distinct irreducible factors of degree at most d of a polynomial $f \in K[X]$ over a number field K is bounded as $N_d(f) \leq (L_K(f) + d)^c$, where c is a constant only depending on K. Soon after, Cheng [11] observed that the L-conjecture directly implies a recent deep result in arithmetic geometry (torsion theorem for elliptic curves [18]) and even stronger statements, which are not (yet) known to be true. This indicates that a proof of the τ-conjecture (if true at all) should rely on very deep insights and techniques in arithmetic algebraic geometry, which are not yet developed and probably won't be so in the near future.

Theorem 1.1 was essentially conjectured by Bürgisser in [7, §8.3]. Koiran [15] proved the following weaker version of the statement regarding the factorials: if $(n!)$ is hard to compute, then $\text{VP}^0 \neq \text{VNP}^0$ or $\mathsf{P} \neq \mathsf{PSPACE}$. Hereby, VP^0 and VNP^0 denote complexity classes in the constant-free Valiant model, see §2.2 for definitions. (The statement $\text{VP}^0 \neq \text{VNP}^0$ seems a bit weaker than the assumption that $\tau(\text{PER}_n)$ is not polynomially bounded in n.) Koiran also proved that if either of the sequences $(\lfloor 2^n \log n \rfloor)$ or $(\lfloor 2^n \pi \rfloor)$ is hard to compute, then $\text{VP}^0 \neq \text{VNP}^0$. He then asked whether the same conclusion can be drawn for the sequences $(\lfloor 2^n e \rfloor)$, $(\lfloor 2^n \sqrt{2} \rfloor)$, or $(\lfloor (3/2)^n \rfloor)$. We prove that this is indeed the case (Corollary 4.3).

The main new idea for the proof of Theorem 1.1 is the consideration of the counting hierarchy CH, which was introduced by Wagner [30]. This is a complexity class lying between PP and PSPACE that bears more or less the same relationship to $\#\mathsf{P}$ as the polynomial hierarchy bears to NP. The key technical ingredient of our proof is the existence of Dlogtime-uniform threshold circuits of constant depth for iterated multiplication via Chinese remaindering. Whether Dlogtime-uniformity can be achieved was an outstanding issue since the paper by Beame et al. [3], that was finally resolved affirmatively in Hesse et al. [14]. Our statements on sequences of integers definable in the counting hierarchy, treated in §3, follow from Hesse et al. [14] in a rather straightforward way by "scaling up to the counting hierachy", see also Allender et al. [1].

Acknowledgements. I am very much grateful to Eric Allender for drawing my attention to the counting hierarchy and answering my questions about it. I thank Emmanuel Jeandel, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen for discussions.

2 Preliminaries

2.1 The counting hierarchy

The (polynomial) counting hierarchy was introduced by Wagner [30] with the goal of classifying the complexity of certain combinatorial problems where counting is involved. It is best defined by means of a counting operator $\mathbf{C}\cdot$ that can be applied to complexity classes.

We denote by $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*, (x,y) \mapsto \langle x,y\rangle$ a pairing function (e.g., by duplicating each bit of x and y and inserting 01 in between).

Definition 2.1 Let K be a complexity class. We define $\mathbf{C}\cdot K$ to be the set of all languages A such that there exist a language $B \in K$, a polynomial p, and a polynomial time computable function $f\colon \{0,1\}^* \to \mathbb{N}$ such that for all $x \in \{0,1\}^*$:

$$x \in L \iff |\{y \in \{0,1\}^{p(|x|)} \mid \langle x,y\rangle \in B\}| > f(x). \tag{2}$$

Remark 2.2 The operators $\exists\cdot$ and $\forall\cdot$ can be introduced in similar way by instead requiring $\exists y \in \{0,1\}^{p(|x|)}\ \langle x,y\rangle \in B$ and $\forall y \in \{0,1\}^{p(|x|)}\ \langle x,y\rangle \in B$, respectively. It is clear that $K \subseteq \exists\cdot K \subseteq \mathbf{C}\cdot K$ and $K \subseteq \forall\cdot K \subseteq \mathbf{C}\cdot K$.

By starting with the class $K = \mathsf{P}$ of languages decidable in polynomial time and iteratively applying the operator $\mathbf{C}\cdot$ we obtain the counting hierarchy.

Definition 2.3 The k-th level $\mathsf{C_kP}$ of the counting hierarchy is recursively defined by $\mathsf{C_0P} := \mathsf{P}$ and $\mathsf{C_{k+1}P} := \mathbf{C}\cdot\mathsf{C_kP}$ for $k \in \mathbb{N}$. One defines CH as the union of all classes $\mathsf{C_kP}$.

We recall that the classes of the polynomial hierarchy PH are obtained from the class P by iteratively applying the operators $\exists\cdot$ and $\forall\cdot$. It follows from Remark 2.2 that the union PH of these classes is contained in CH. Also it is not hard to see that CH is contained in the class PSPACE of languages decidable in polynomial space.

Modifying Definition 2.1 we define $\mathbf{C}' \cdot K$ of a complexity class K by requiring the majority condition

$$x \in L \Longleftrightarrow |\{y \in \{0,1\}^{p(|x|)} \mid \langle x, y\rangle \in B\}| > 2^{p(|x|)-1}.$$

instead of (2). It can be shown that this does not change the definition of the classes of the counting hierarchy $\mathsf{C_kP}$, cf. Torán [25]. In particular, we obtain for $k = 1$ the definition of the familiar class PP (probabilistic polynomial time).

We recall also that the counting complexity class $\#\mathsf{P}$ consists of all functions $g\colon \{0,1\}^* \to \mathbb{N}$ for which there exist a language $B \in \mathsf{P}$ and a polynomial p such that for all $x \in \{0,1\}^*$:

$$g(x) = |\{y \in \{0,1\}^{p(|x|)} \mid \langle x, y\rangle \in B\}|.$$

Hence functions in $\#\mathsf{P}$ can by evaluated in polynomial time by oracle calls to PP.

Torán [25] has obtained the following alternative characterization of the counting hierarchy, which is quite analogous to the corresponding characterization of the polynomial hierarchy:

$$\mathsf{C_{k+1}P} = \mathsf{PP}^{\mathsf{C_kP}}. \tag{3}$$

We recall the definition of the nonuniform version K/poly of a complexity class K by polynomial advice functions.

Definition 2.4 The nonuniform version K/poly of a complexity class K consists of all languages A for which there exists a language $B \in K$ and a function $\alpha\colon \mathbb{N} \to \{0,1\}^*$ with $\alpha(n)$ polynomially bounded in n, such that $x \in A$ iff $\langle x, \alpha(x)\rangle \in B$, for all $x \in \{0,1\}^*$.

Lemma 2.5 *The counting hierarchy collapses to* P *if* $\mathsf{PP} = \mathsf{P}$. *Moreover,* $\mathsf{PP} \subseteq \mathsf{P/poly}$ *implies* $\mathsf{CH} \subseteq \mathsf{P/poly}$.

Proof. Suppose $\mathsf{PP} \subseteq \mathsf{P/poly}$. We prove $\mathsf{C_kP} \subseteq \mathsf{P/poly}$ by induction on k. The start $k = 0$ being clear, let $A \in \mathsf{C_{k+1}P} = \mathbf{C}' \cdot \mathsf{C_kP}$. By definition, there exist $B \in \mathsf{C_kP}$ and a polynomial p such that for all $n \in \mathbb{N}$, $x \in \{0,1\}^n$,

$$x \in A \Longleftrightarrow |\{y \in \{0,1\}^{p(n)} \mid \langle x, y\rangle \in B\}| > 2^{p(n)-1}.$$

By induction hypothesis, we have $B \in \mathsf{P/poly}$. Hence there exists $D \in \mathsf{P}$ and an advice function $\alpha\colon \mathbb{N} \to \{0,1\}^*$ such that $z \in B$ iff $\langle z, \alpha(|z|)\rangle \in D$. Hence $x \in A$ iff

$$|\{y \in \{0,1\}^{p(n)} \mid \langle\langle x, y\rangle, \alpha(n + p(n))\rangle \in D\}| > 2^{p(n)-1}.$$

It follows that $A \in \mathsf{PP/poly}$. Hence $A \in \mathsf{P/poly}$. □

The counting hierarchy is closely tied to the theory of threshold circuits of bounded depth, cf. [2].

Recall that a majority gate outputs 1 iff the majority of its inputs have the value 1. A *threshold circuit* is a Boolean circuit consisting of majority gates only. The class of languages decidable by a family of threshold circuits of polynomial size and depth $\mathcal{O}(1)$ is denoted TC^0. This class is known to characterize the power of (iterated) integer multiplication. We refer to the textbook by Vollmer [29] for an introduction to this subject.

Beame et al. [3] presented parallel NC^1-algorithms for iterated multiplication and division of integers. Reif and Tate [19] observed that these algorithms can also be implemented by constant depth threshold circuits, placing these problems in the class TC^0. The question of the degree of uniformity required for these circuits was only recently solved in a satisfactory way by Hesse et al. [14], who showed that there are Dlogtime-uniform circuits performing these tasks. This result will be crucial in §3 for our study of sequences of integers definable in the counting hierarchy.

2.2 The constant-free Valiant model

An *arithmetic circuit* over the field $\mathbb{Q}$ is an acyclic finite digraph, where all nodes except the input nodes have fan-in 2 and are labelled by $+, -, \times$ or $/$. The circuit is called *division-free* if there are no division nodes. The input nodes are labelled by variables from $\{X_1, X_2, \ldots\}$ or by constants in $\mathbb{Q}$. If all constants belong to $\{-1, 0, 1\}$, then the circuit is said to be *constant-free*. We assume that there is exactly one output node, so that the circuit computes a rational function in the obvious way. By the *size* of a circuit we understand the number of its nodes different from input nodes.

Definition 2.6 The *L-complexity* $L(f)$ of a rational polynomial f is defined as the minimum size of an arithmetic circuit computing f. The *τ-complexity* $\tau(f)$ of an integer polynomial f is defined as the minimum size of a divison-free and constant-free arithmetic circuit computing f.

Note that $L(f) \leq \tau(f)$. While $L(c) = 0$ for any $c \in \mathbb{Q}$, it makes sense to consider the τ-complexity of an integer k. For instance, one can show that $\log\log k \leq \tau(k) \leq 2\log k$ for any $k \geq 2$, cf. [12].

In order to control the degree and the size of the coefficients of f we are going to put further restrictions on the circuits. The *(complete) formal degree* of a node is inductively defined as follows: input nodes have formal degree 1 (also those labelled by constants). The formal degree of an addition or subtraction node is the maximum of the formal degrees of the two incoming nodes, and the formal degree of a multiplication node is the sum of these formal degrees. The formal degree of a circuit is defined as the formal degree of its output node.

Valiant's algebraic model of NP-completeness [26, 28] (see also [7]) explains the hardness of computing the permanent polynomial in terms of an algebraic completeness result. For our purposes, it will be necessary to work with a variation of this model. This constant-free model has been systematically studied by Malod [17]. We briefly present the salient features following Koiran [15].

Definition 2.7 A sequence (f_n) of polynomials belongs to the complexity class VP^0 iff there exists a sequence (C_n) of division-free and constant-free arithmetic circuits such that C_n computes f_n and the size and the formal degree of C_n are polynomially bounded in n.

Clearly, if $(f_n) \in \mathrm{VP}^0$ then $\tau(f_n) = n^{\mathcal{O}(1)}$. Moreover, it is easy to see that the bitsize of the coefficients of f_n is polynomially bounded in n. When removing in the above definition the adjective "constant-free", the original class VP over the field $\mathbb{Q}$ is obtained [17]. The class VP^0 is universal in the sense that a family (g_n) is in VP iff there exists a family (f_n) in VP^0 such that g_n can be obtained from f_n by substituting some of the variables by constants in $\mathbb{Q}$.

The counterpart to VP^0 is the following class.

Definition 2.8 A sequence $(f_n(X_1, \ldots, X_{u(n)}))$ of polynomials belongs to the complexity class VNP^0 iff there exists a sequence $(g_n(X_1, \ldots, X_{v(n)}))$ in VP^0 such that

$$f_n(X_1, \ldots, X_{u(n)})) = \sum_{e \in \{0,1\}^{v(n)-u(n)}} g_n(X_1, \ldots, X_{u(n)}, e_1, \ldots, e_{v(n)-u(n)}).$$

(Hereby $u(n)$ and $v(n)$ are polynomially bounded functions of n.)

We note that by replacing VP^0 by VP in this definition, the original class VNP over $\mathbb{Q}$ is obtained.

Valiant developed the following useful criterion [26, Remark 1] for recognizing families in VNP^0, see also [7, Proposition 2.20] and [15, Theorem 2.3]. For instance, this criterion easily implies that the sequence $(\textsc{Per}_n)$ of permanent polynomials lies in the class VNP^0.

Proposition 2.9 *Consider a map $a\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}, (n, j) \mapsto a(n, j)$ that lies in the complexity class $\#\mathsf{P}/\mathsf{poly}$, when n is encoded in* unary *and j in binary. Let $p\colon \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function and let j_i denotes the bit of $0 \leq j < 2^{p(n)}$ of weight 2^{i-1}. Then the following sequence (f_n) of polynomials is in* VNP^0*:*

$$f_n(X_1, \ldots, X_{p(n)}) = \sum_{j=0}^{2^{p(n)}-1} a(n, j) X_1^{j_1} \cdots X_{p(n)}^{j_{p(n)}}.$$

Valiant's algebraic completeness result implies that VP = VNP iff $(\textsc{Per}_n) \in \mathrm{VP}$. The latter is equivalent to $L(\textsc{Per}_n) = n^{\mathcal{O}(1)}$. In the constant-free setting, the situation seems more complicated. It is not clear that $\mathrm{VP}^0 = \mathrm{VNP}^0$ is equivalent to the hypothesis $\tau(\textsc{Per}_n) = n^{\mathcal{O}(1)}$. Curiously, it is neither clear whether $(\textsc{Per}_n) \in \mathrm{VP}^0$ and $\mathrm{VP}^0 = \mathrm{VNP}^0$ are equivalent. However, it is known that they become equivalent when considering arithmetic circuits using the additional constant $\frac{1}{2}$, cf. Koiran [15, Theorem 4.3] and the result below.

Theorem 2.10 *Suppose $\tau(\textsc{Per}_n) = n^{\mathcal{O}(1)}$. Then for any family $(f_n) \in \mathrm{VNP}^0$ there exists a polyomially bounded sequence $(p(n))$ in $\mathbb{N}$ such that $\tau(2^{p(n)} f_n) = n^{\mathcal{O}(1)}$.*

Proof. An inspection of Valiant's algebraic completeness result (see for instance [7]) reveals that any family (f_n) in VNP^0 can be expressed as a projection $f_n = \textsc{Per}_{p(n)}(y_1, \ldots, y_{p(n)^2})$, where $p(n)$ is polynomially bounded in n and the y_i are either variables or constants taken from $\{-1, -1/2, 0, 1/2, 1\}$. By homogeneity of the permanent we get $2^{p(n)} f_n = \textsc{Per}_{p(n)}(2y_1, \ldots, 2y_{q(n)^2})$. This shows the first claim. □

Valiant's criterion (Proposition 2.9) has been "scaled down" by Koiran [15, Theorem 6.1] as follows.

Theorem 2.11 *Assume the map $a\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}, (n, j) \mapsto a(n, j)$ is in the complexity class $\#\mathsf{P}/\mathsf{poly}$, where n, j are encoded in binary. Let $p\colon \mathbb{N} \to \mathbb{N}$ be polynomially bounded and satisfying $p(n) \geq n$ for all n. Consider the polynomial*

$$F_n(X_1, \ldots, X_{\ell(n)}) = \sum_{j=0}^{p(n)} a(n, j) X_1^{j_1} \cdots X_{\ell(n)}^{j_{\ell(n)}},$$

where $\ell(n) = 1 + \lfloor \log p(n) \rfloor$ and j_i denotes the bit of j of weight 2^{i-1}. Then there exists a family $(G_r(X_1, \ldots, X_r, N_1, \ldots, N_r, P_1, \ldots, P_r))_{r\in\mathbb{N}}$ in VNP^0 that satisfies

$$F_n(X_1, \ldots, X_{\ell(n)}) = G_{\ell(n)}(X_1, \ldots, X_{\ell(n)}, n_1, \ldots, n_{\ell(n)}, p_1, \ldots, p_{\ell(n)}))$$

for all n, where n_i and p_i denote the bits of n and $p(n)$ of weight 2^{i-1}, respectively.

We will also need the following observation.

Lemma 2.12 *$\tau(\textsc{Per}_n) = n^{\mathcal{O}(1)}$ implies that $\mathsf{PP} \subseteq \mathsf{P}/\mathsf{poly}$.*

Proof. Suppose there is a family $(\mathcal{C}_n)$ of constant-free and division-free arithmetic circuits of polynomial size such that $\mathcal{C}_n$ computes the permanent $\textsc{Per}_n$. Let p_n be a prime such that $n! < p_n \leq 2^{n^{\mathcal{O}(1)}}$ (p_n is interpreted as a polynomial advice for input size n). On an input $A \in \{0, 1\}^{n\times n}$, we execute the arithmetic circuit $\mathcal{C}_n$ in the finite field $\mathbb{F}_{p_n}$. This computation can clearly be simulated by a Boolean circuit of polynomial size. Moreover, the result $\textsc{Per}(A) \bmod p_n$ the integer value of the permanent of A can be retrieved. Since the computation of the permanent of matrices with entries in $\{0, 1\}$ is #P-complete [27], we conclude $\mathsf{PP} \subseteq \mathsf{P}/\mathsf{poly}$. □

We reamrk that the proof of the above lemma can be extended to handle also arithmetic circuits using divisions.

3 Integers definable in the counting hierarchy

We consider sequences of integers $a(n,k)$ defined for $n,k \in \mathbb{N}$ and $0 \leq k \leq q(n)$, where q is polynomially bounded, such that

$$\forall n > 1 \; \forall k \leq q(n) \quad |a(n,k)| \leq 2^{n^c} \tag{4}$$

for some constant c. We shall briefly refer to such sequences $a = (a(n,k))$ as being of *polynomial bitsize*. The falling factorials $a(n,k) = n(n-1)\cdots(n-k+1)$ are an interesting example to keep in mind; note that $a(n,k) \leq 2^{n^2}$.

We shall write $|a| := (|a(n,k)|)$ for the sequence of absolute values of a. We assign to a sequence $a = (a(n,k))$ of polynomial bitsize the following languages with the integers n,k,j represented in binary (using $\mathcal{O}(\log n)$ bits):

$$\begin{aligned} \mathrm{Sgn}(a) &:= \{(n,k) \mid a(n,k) \geq 0\} \\ \mathrm{Bit}(|a|) &:= \{(n,k,j,b) \mid \text{ the } j\text{-th bit of } |a(n,k)| \text{ equals } b \,\}. \end{aligned}$$

The integer j can thus be interpreted as an address pointing to bits of $a(n,k)$. Because of (4), we have $j \leq n^c$ and thus $\log j = \mathcal{O}(\log n)$.

Definition 3.1 A sequence a of integers of polynomial bitsize is called *definable in the counting hierarchy* CH iff $\mathrm{Sgn}(a) \in \mathsf{CH}$ and $\mathrm{Bit}(|a|) \in \mathsf{CH}$. If both $\mathrm{Sgn}(a)$ and $\mathrm{Bit}(|a|)$ lie in $\mathsf{CH/poly}$ then we say that a is definable in $\mathsf{CH/poly}$.

This definition and all what follows extends to sequences $(a(n,k_1,\ldots,k_t))$ with a fixed number t of subordinate indices $k_1,\ldots k_t \leq n^{\mathcal{O}(1)}$ in a straightforward way. For the sake of simplifying notation we only state our results for the cases $t \in \{0,1\}$.

Remark 3.2 If $n \mapsto a(n)$ is computable in polynomial time, then clearly $\mathrm{Sgn}(a) \in \mathsf{P}$ and $\mathrm{Bit}(|a|) \in \mathsf{P}$. In particular, a is definable in CH. (Note that in this case $\log a(n) = (\log n)^{\mathcal{O}(1)}$.)

Our next goal is to find a useful criterion for showing that specific sequences are definable in CH. Let $m \bmod p \in \{0,\ldots,p-1\}$ denote the remainder of m upon division by the prime p. We assign to $a = (a(n,k))$ and a corresponding constant $c > 0$ satisfying (4) the *Chinese remainder language*

$$\mathrm{CR}(a) := \{(n,k,p,j,b) \mid \; p \text{ prime}, p < n^{2c}, \text{ the } j\text{-th bit of } a(n,k) \bmod p \text{ equals } b \,\}.$$

Again, the integers n,k,p,j are to represented in binary with $\mathcal{O}(\log n)$ bits. (We suppress the dependence of $\mathrm{CR}(a)$ on c to simplify notation.) Note that the absolute value $|a(n,k)| \leq 2^{n^c}$ is uniquely determined by the residues $a(n,k) \bmod p$ for the primes $p < n^{2c}$, since the product of these primes is larger than 2^{n^c} (for $n > 1$).

Lemma 3.3 *Suppose that the sequence $a = (a(n))$ of integers is easy to compute in the sense of Shub and Smale [20], that is, $\tau(a(n)) = (\log n)^{\mathcal{O}(1)}$. Then* $\mathrm{CR}(a) \in \mathsf{P/poly}$.

PROOF. By assumption, there are arithmetic circuits $\mathcal{C}_n$ of size $(\log n)^{\mathcal{O}(1)}$ computing $a(n)$. On input (n, k, p, j, b), given the advice $\mathcal{C}_n$, we evaluate $\mathcal{C}_n$ in the finite field $\mathbb{F}_p$ to obtain $a(n) \bmod p$. This is possible in time polynomial in $\log n$ as $\log p = \mathcal{O}(\log n)$. □

The following criterion for definability in CH turns out to be a rather straightforward consequence of the results in Hesse et al. [14] on uniform bounded-depth threshold circuits for division and iterated multiplication of integers.

Theorem 3.4 *Let a be a sequence of integers of polynomial bitsize. Then a is definable in CH iff $\mathrm{Sgn}(a) \in \mathsf{CH}$ and $\mathrm{CR}(a) \in \mathsf{CH}$. Moreover, a is definable in $\mathsf{CH/poly}$ iff $\mathrm{Sgn}(a) \in \mathsf{CH/poly}$ and $\mathrm{CR}(a) \in \mathsf{CH/poly}$.*

PROOF. We first show that for nonnegative sequences a of polynomial bitsize

$$a \text{ is definable in } \mathsf{CH} \iff \mathrm{CR}(a) \in \mathsf{CH} \tag{5}$$

and similarly for the nonuniform situation.

By the Chinese Remainder Representation (CRR) of an integer $0 \leq X \leq 2^n$ we understand the sequence of bits indexed (p, j) giving the j-th bit of $X \bmod p$, for each prime $p < n^2$. (The length of this sequence is $\mathcal{O}(n^2)$.)

It was shown by Hesse et al. [14, Theorem 4.1] that there are Dlogtime-uniform threshold circuits of polynomial size and depth bounded by a constant D that on input the Chinese Remainder Representation of $0 \leq X \leq 2^n$ compute the binary representation of X. Let this circuit family be denoted by $\{\mathcal{C}_n\}$.

Suppose that a is a sequence of nonnegative integers satisfying (4). For $d \in \mathbb{N}$ consider the language L_d consisting of the binary encodings of (n, k, F, b), where F is the name of a gate on level at most d of the threshold circuit $\mathcal{C}_{n^c}$ and F evaluates to b on input the CRR of $a(n, k)$.

Claim. $L_{d+1} \in \mathsf{PP}^{L_d}$ for $0 \leq d < D$.

We argue as in [1]. Due to the Dlogtime-uniformity of the circuits we can check in linear time whether two gates F and G are connected. Let F be a gate at level $d+1$. On input (n, k, F, b), we need to determine the majority of the gates G connected to F such that $(n, k, G, 1) \in L_d$. This is possible in PP^{L_d}, which proves the claim.

We can now show the direction from right to left of (5). Suppose that $\mathrm{CR}(a)$ is contained in the s-th level $\mathsf{C_sP}$ of the counting hierarchy. This means that $L_0 \in \mathsf{C_sP}$. Using the claim and (3) we conclude that $L_d \in \mathsf{C_{s+d}P} \subseteq \mathsf{C_{s+D}P}$. Applying this to the output gates of $\mathcal{C}_{n^c}$ we see that a is definable in CH. Similarly, if $\mathrm{CR}(a) \in \mathsf{C_sP/poly}$ we obtain $L_d \in \mathsf{C_{s+d}P/poly}$.

In order to show the direction from left to right of (5) we argue in the same way, using the fact that the reverse task of computing the CRR of $0 \leq X \leq 2^n$ from the binary representation of X can be accomplished by Dlogtime-uniform threshold circuits of polynomial size and constant depth, cf. [14, Lemma 4.1].

For completing the proof it now suffices prove that

$$\mathrm{Sgn}(a) \in \mathsf{CH} \text{ and } \mathrm{CR}(a) \in \mathsf{CH} \Longleftrightarrow \mathrm{Sgn}(a) \in \mathsf{CH} \text{ and } \mathrm{CR}(|a|) \in \mathsf{CH}$$

and similarly for the nonuniform situation. However, this follows from the fact that $-X \bmod p$ can be computed from $X \bmod p$ in AC^0, cf. [29]. □

Corollary 3.5 *If a and b are two sequences of nonnegative integers definable in* CH*, then so is $a - b$. Similarly in the nonuniform situation.*

Proof. By Theorem 3.4 we know that $\mathrm{CR}(a), \mathrm{CR}(b) \in \mathsf{CH}$. Using [14, Lemma 4.3] and proceeding as in the proof of Theorem 3.4 we conclude that $\mathrm{Sgn}(a - b) \in \mathsf{CH}$. Moreover it is obvious that $\mathrm{CR}(a - b) \in \mathsf{CH}$. Now apply again Theorem 3.4. In the nonuniform case similar arguments apply □

Corollary 3.6 *If the sequence $a = (a(n))$ of integers is easy to compute, then a is definable in* CH/poly.

Proof. Lemma 3.3 tells us that $\mathrm{CR}(a) \in \mathsf{P/poly} \subseteq \mathsf{CH/poly}$ if a is easy to compute. The nonnegative sequence $\widetilde{a}(n) := a(n) + 2^{\lceil n^c \rceil}$ is also easy to compute. We have $a(n) \geq 0$ iff $\widetilde{a}(n) \geq 2^{\lceil n^c \rceil}$. Corollary 3.5 thus implies that $\mathrm{Sgn}(a) \in \mathsf{CH/poly}$. (For a more precise statement we refer to Allender et al. [1].) The assertion follows with Theorem 3.4. □

From the above criterion we can derive the following closure properties with respect to iterated addition, iterated multiplication, and integer division.

Theorem 3.7 *1. Suppose $a = (a(n,k))_{n \in \mathbb{N}, k \leq q(n)}$ is definable in* CH*, where q is polynomially bounded. Consider*

$$b(n) := \sum_{k=0}^{q(n)} a(n,k), \quad d(n) := \prod_{k=0}^{q(n)} a(n,k).$$

Then $b = (b(n))$ and $d = (d(n))$ are definable in CH*. Moreover, if a is is definable in* CH/poly*, then so are b and d.*

2. Suppose $(s(n))_{n \in \mathbb{N}}$ and $(t(n))_{n \in \mathbb{N}}$ are definable in CH *and $t(n) > 0$ for all n. Then the sequence of quotients $(\lfloor s(n)/t(n) \rfloor)_{n \in \mathbb{N}}$ is definable in* CH*. The analogous assertion holds for* CH/poly*.*

PROOF. 1. Iterated addition is the problem to compute the sum of n integers $0 \le X_1, \ldots, X_n \le 2^n$ in binary. This problem is well known to be in Dlogtime-uniform TC^0, cf. [29]. By scaling up this result as in the proof of Theorem 3.4, we obtain the claim for b in the case where $a(n,k) \ge 0$.

The general case for b follows by applying this to each of two sums in

$$b(n) = \sum_{k=0}^{q(n)} a(n,k) \cdot 1_{\{a(n,k) \ge 0\}} - \sum_{k=0}^{q(n)} (-a(n,k)) \cdot 1_{\{a(n,k) < 0\}}$$

and by using Corollary 3.5.

The claim for the iterated multiplication will follow by scaling up the arguments in Hesse at al. [14] to the counting hierarchy. Those arguments are similar as in Beame et al. [3], except that the much stronger Dlogtime-uniformity condition was achieved in [14]. We need this uniformity condition for obtaining our result.

Suppose that a is definable in CH. First note that we can check for given n in CH whether all $a(n,k)$ are nonzero. We therefore assume w.l.o.g. that $a(n,k) \ne 0$ and write $a(n,k) = (-1)^{e(n,k)}|a(n,k)|$ with $e(n,k) \in \{0,1\}$. By definition, the sequence $(e(n,k))$ is definable in CH. We have

$$d(n) = (-1)^{s(n)} \prod_k |a(n,k)| \quad \text{where } s(n) = \sum_{k=0}^{q(n)} e(n,k).$$

According to the first claim of the theorem, $(s(n))$ is definable in CH. Hence it suffices to prove the second claim for a nonnegative sequence a.

By Theorem 3.4 we know $\mathrm{CR}(a) \in \mathsf{CH}$ and it suffices to prove that $\mathrm{CR}(d) \in \mathsf{CH}$. Suppose d satisfies (4) with the constant $c > 0$. Let a prime $p \le n^{2c}$ be given. We can find the smallest generator g of the cyclic group $\mathbb{F}_p^\times$ in P^{PH} by bisecting according to the following oracle in Σ_2 ($u < p$):

$$\exists\, 1 \le g < u\ \forall\, 1 \le i < p \quad g^i \ne 1.$$

Note that g^i can be computed by repeated squaring in polynomial time.

Similarly, for a given $u \in \mathbb{F}_p^\times$, we can compute the discrete logarithm $0 \le i < p$ defined by $u = g^i$ in P^{NP}.

For given $k \le q(n)$ let $\alpha(n,k)$ denote the discrete logarithm of $a(n,k) \bmod p$. By the previous reasonings we see that $(\alpha(n,k))$ is definable in CH. By part one of the theorem we conclude that $(\delta(n))$ defined by $\delta(n) = \sum_{k=0}^{q(n)} \alpha(n,k)$ is definable in CH. Hence $d(n) \bmod p = g^{\gamma(n)}$ is computable in CH. Similar arguments apply in the nonuniform case.

2. The claim for integer division follows as before by scaling up the arguments in Beame et al. [3] and Hesse et al. [14] to the counting hierarchy. □

Corollary 3.8 *The sequence of factorials* $(n!)$ *is definable in* CH. *More generally, the falling factorials* $(n(n-1)\cdots(n-k+1))_{k\leq n}$ *are definable in* CH.

PROOF. This follows from Theorem 3.7 and Remark 3.2. □

We denote by $\sigma_k(z_1,\ldots,z_n)$ the k-th elementary symmetric function in the variables $z_1,\ldots,z_n$ $(0\leq k\leq n)$.

Corollary 3.9 *The sequence* $(\sigma_k(1,2,\ldots,n))_{n\in\mathbb{N},k\leq n}$ *is definable in* CH.

PROOF. Starting from $(X+1)\cdots(X+n)=\sum_{k=0}^{n}\sigma_k(1,\ldots,n)X^{n-k}$ and substituting T by 2^{n^2} we get

$$d(n):=(2^{n^2}+1)\cdots(2^{n^2}+n)=\sum_{k=0}^{n}\sigma_k(1,\ldots,n)2^{n^2(n-k)}.$$

Since $\sigma_k(1,2,\ldots,n)<2^{n^2}$ there is no overlap of the bit representations, hence the bits of $\sigma_k(1,2,\ldots,n)$ can be read off the bit vector of $d(n)$. It is therefore sufficient to show that $(d(n))$ is definable in CH.

Using Theorem 3.7, it is enough to prove that the sequence $c(n,k)=2^{n^2}+k$ for $k\leq n, n\in\mathbb{N}$ is definable in CH. However, it is clear that $\mathrm{Bit}(c)\in\mathsf{P}$. □

4 Connecting Valiant's model to integers and univariate polynomials

We establish now the announced connection between Valiant's constant-free model and sequences of polynomials having coefficient sequences that are definable in the counting hierarchy.

Theorem 4.1 *Consider a sequence* $(a(n))_{n\in\mathbb{N}}$ *of integers definable in* $\mathsf{CH}/\mathsf{poly}$ *and sequences*

$$f_n=\sum_{k=0}^{q(n)}b(n,k)X^k\in\mathbb{Z}[X],\quad g_n=\frac{1}{d(n)}f_n\in\mathbb{Q}[X]$$

of integer and rational polynomials, respectively, such that $(b(n,k))_{n\in\mathbb{N},k\leq q(n)}$ *and* $(d(n))_{n\in\mathbb{N}}$ *are definable in* $\mathsf{CH}/\mathsf{poly}$ *(in particular,* q *is polynomially bounded).*

If $\tau(\mathrm{PER}_n)=n^{\mathcal{O}(1)}$, *then the following holds:*

1. $\tau(a(n))=(\log n)^{\mathcal{O}(1)}$.
2. $\tau(2^{e(n)}f_n)=(\log n)^{\mathcal{O}(1)}$ *for some polynomially bounded sequence* $(e(n))$ *in* $\mathbb{N}$.
3. $L(g_n)=(\log n)^{\mathcal{O}(1)}$.

PROOF. We assume that $\tau(\mathrm{PER}_n) = n^{\mathcal{O}(1)}$. By Lemma 2.12 this yields $\mathsf{PP} \subseteq \mathsf{P/poly}$. According to Lemma 2.5, this implies that $\mathsf{CH} \subseteq \mathsf{P/poly}$.

1. Let $a(n) = \sum_{j=0}^{p(n)} a(n,j)2^j$ be the binary representation of $a(n)$. Without loss of generality we may assume that the polynomially bounded function p satisfies $p(n) \geq n$. By assumption, we can decide $a(n,j) = b$ in $\mathsf{CH/poly}$, where n, j are given in binary. Because of the assumed collapse of the counting hierarchy we can decide $a(n,j) = b$ in $\mathsf{P/poly}$.

Consider the polynomial

$$A_n(Y_1, \ldots, Y_{\ell(n)}) = \sum_{j=0}^{p(n)} a(n,j) Y_1^{j_1} \cdots Y_{\ell(n)}^{j_{\ell(n)}},$$

where $\ell(n) = 1 + \lfloor \log p(n) \rfloor$ and j_i denotes the bit of j of weight 2^{i-1}. Note that

$$A_n(2^{2^0}, 2^{2^1}, \ldots, 2^{2^{\ell(n)-1}}) = a(n)$$

By Theorem 2.11 there is a family $(G_r(Y_1, \ldots, Y_r, N_1, \ldots, N_r, P_1, \ldots, P_r))$ in VNP^0 that satisfies for all n

$$A_n(Y_1, \ldots, Y_{\ell(n)}) = G_{\ell(n)}(Y_1, \ldots, Y_{\ell(n)}, n_1, \ldots, n_{\ell(n)}, p_1, \ldots, p_{\ell(n)}),$$

where n_i and p_i denote the bits of n and $p(n)$ of weight 2^{i-1}, respectively.

By Theorem 2.10 there exists a polynomially bounded sequence $(s(r))$ in $\mathbb{N}$ such that $\tau(2^{s(r)} G_r) = r^{\mathcal{O}(1)}$. This implies $\tau(2^{e(n)} G_{\ell(n)}) = (\log n)^{\mathcal{O}(1)}$, where $e(n) = s(\ell(n)) = (\log n)^{\mathcal{O}(1)}$. We conclude from the above that

$$2^{e(n)} a(n) = 2^{e(n)} G_{\ell(n)}(2^{2^0}, 2^{2^1}, \ldots, 2^{2^{\ell(n)-1}}, n_1, \ldots, n_{\ell(n)}, p_1, \ldots, p_{\ell(n)}),$$

hence

$$\tau(2^{e(n)} a(n)) \leq \tau(2^{e(n)} G_{\ell(n)}) + \ell(n) \leq (\log n)^{\mathcal{O}(1)}.$$

Lemma 4.4 in Koiran [15] implies $\tau(a(n)) \leq (2e(n)+3)\tau(2^{e(n)} a(n))$. Altogether, we obtain $\tau(a(n)) = (\log n)^{\mathcal{O}(1)}$.

2. Let $b(n,k) = \sum_{j=0}^{p(n)} b(n,k,j)2^j$ be the binary representation of $b(n,k)$ for $k \leq q(n)$. As before we assume $p(n) \geq n$ without loss of generality. Consider the polynomial

$$B_n(Y_1, \ldots, Y_{\ell(n)}, Z_1, \ldots, Z_{\lambda(n)}) = \sum_{j=0}^{p(n)} \sum_{k=0}^{q(n)} b(n,k,j) Y_1^{j_1} \cdots Y_{\ell(n)}^{j_{\ell(n)}} Z_1^{k_1} \cdots Z_{\lambda(n)}^{k_{\lambda(n)}},$$

where $\ell(n) = 1 + \lfloor \log p(n) \rfloor$, $\lambda(n) = 1 + \lfloor \log q(n) \rfloor$, and j_i, k_i denote the bit of j, k of weight 2^{i-1}, respectively. Note that

$$B_n(2^{2^0}, 2^{2^1}, \ldots, 2^{2^{\ell(n)-1}}, X^{2^0}, X^{2^1}, \ldots, X^{2^{2^{\lambda(n)-1}}}) = \sum_{k=0}^{q(n)} b(n,k) X^k = f_n.$$

By Theorem 2.11 there is a family $(G_r((X_1,\dots,X_r),(N_1,\dots,N_r),(P_1,\dots,P_r)))$ in VNP^0 that satisfies for all n

$$B_n(Y,Z) = G_{\ell(n)+\lambda(n)}((Y,Z),(n_1,\dots,n_{\ell(n)+\lambda(n)}),(p_1,\dots,p_{\ell(n)},q_1,\dots,q_{\lambda(n)})),$$

where $(Y,Z) = (Y_1,\dots,Y_{\ell(n)},Z_1,\dots,Z_{\lambda(n)})$ and n_i, p_i, and q_i denote the bits of n, $p(n)$, and $q(n)$ of weight 2^{i-1}, respectively. By Theorem 2.10 there exists a polynomially bounded sequence $(s(r))$ in $\mathbb{N}$ such that $\tau(2^{s(r)}G_r) = r^{\mathcal{O}(1)}$. This implies $\tau(2^{e(n)}G_{\ell(n)+\lambda(n)}) = (\log n)^{\mathcal{O}(1)}$, where $e(n) := s(\ell(n)+\lambda(n)) = (\log n)^{\mathcal{O}(1)}$. We conclude from the above that

$$\tau(2^{e(n)}f_n) \le \tau(2^{e(n)}G_{\ell(n)+\lambda(n)}) + \ell(n) + \lambda(n) \le (\log n)^{\mathcal{O}(1)}.$$

3. We know already that $\tau(2^{e(n)}f_n) = (\log n)^{\mathcal{O}(1)}$. By the first assertion, we have $\tau(d(n)) = (\log n)^{\mathcal{O}(1)}$. Using one division, we conclude that $L(g_n) = (\log n)^{\mathcal{O}(1)}$. □

We can also prove a conditional implication refering to the original Valiant hypothesis VP $\neq$ VNP over $\mathbb{C}$ (dealing with arithmetic circuits using divisions and arbitrary complex constants).

Corollary 4.2 *Assuming the generalized Riemann hypothesis, $L_{\mathbb{C}}(\mathrm{PER}_n) = n^{\mathcal{O}(1)}$ implies that $L_{\mathbb{C}}(g_n) = (\log n)^{\mathcal{O}(1)}$, where g_n is as in Theorem 4.1.*

Proof. Suppose that $L_{\mathbb{C}}(\mathrm{PER}_n) = n^{\mathcal{O}(1)}$. In Bürgisser [8] it was shown that this implies $\mathsf{PP} \subseteq \mathsf{NC/poly} \subseteq \mathsf{PP/poly}$, assuming the generalized Riemann hypothesis. Since (PER_n) is VNP-complete, we have $L_{\mathbb{C}}(f_n) = n^{\mathcal{O}(1)}$ for any $(f_n) \in \mathrm{VNP}$. Now we can argue as in the proof of Theorem 4.1 with $L_{\mathbb{C}}$ instead of τ. □

It is now easy to complete the proof our main result stated in the introduction.

Proof of Theorem 1.1. We suppose that $\tau(\mathrm{PER}_n) = n^{\mathcal{O}(1)}$. 1. The sequence of factorials $a(n) = n!$ is definable in CH according to Corollary 3.8. By Theorem 4.1(1) we get $\tau(n!) = (\log n)^{\mathcal{O}(1)}$.

2. Consider the Pochhammer-Wilkinson polynomial

$$f_n = \prod_{k=1}^{n}(X-k) = \sum_{k=0}^{n}(-1)^k\sigma_k(1,2,\dots,n)\,X^{n-k},$$

which has exactly n integer roots. Corollary 3.9 implies that its coefficient sequence is definable in CH. By Theorem 4.1(2) we have $\tau(2^{e(n)}f_n) = (\log n)^{\mathcal{O}(1)}$ for some $(e(n))$. The polynomial $2^{e(n)}f_n$ violates the τ-conjecture.

3. We have $g_n = \sum_{k=0}^{n}\frac{1}{k!}T^k = \frac{1}{n!}\sum_{k=0}^{n}n(n-1)\cdots(k+1)\;X^k$. According to Corollary 3.8 both the coefficient sequence and the sequence $(n!)$ of denominators are definable in CH. Theorem 4.1(3) implies that $L(g_n) = (\log n)^{\mathcal{O}(1)}$.

4. Similar to 3. □

We proceed with further applications of Theorem 4.1. The following result answers some questions posed by Koiran [15] in the affirmative. From the very general proof technique, it becomes obvious that this result actually holds for a large class of integer sequences, so the choice of the sequences below is for illustration and just motivated by Koiran's question. Of course, one could as well consider expansions in radix different from 2, like $(\lfloor 10^n e \rfloor)_{n\in\mathbb{N}}$.

Corollary 4.3 *If one of the following integer sequences is hard to compute, then then $\tau(\textsc{Per}_n)$ is not polynomially bounded in n:*

$$(\lfloor 2^n e \rfloor)_{n\in\mathbb{N}},\ (\lfloor 2^n \sqrt{2} \rfloor)_{n\in\mathbb{N}},\ (\lfloor (3/2)^n \rfloor)_{n\in\mathbb{N}}.$$

Proof. 1. A straightforward estimation shows that $e = \sum_{k=0}^{\infty} \frac{1}{k!} = \sum_{k=0}^{n+1} \frac{1}{k!} + \varepsilon_n$ with $0 < \varepsilon_n < 2^{-n}$. It follows that $b(n) \le \lfloor 2^n e \rfloor \le b(n) + n + 3$, where

$$b(n) := \sum_{k=0}^{n+1} \lfloor \frac{2^n}{k!} \rfloor.$$

Hence $\lfloor 2^n e \rfloor = b(n) + r(n)$ where $r(n)$ is an integer sequence satisfying $0 \le r(n) \le n + 3$.

The sequence $(r(n))$ is easy to compute since $\tau(m) \le 2 \log m$ for $m \ge 1$, cf. [5]. Hence $(\lfloor 2^n e \rfloor)_{n\in\mathbb{N}}$ is hard to compute iff $(b(n))$ is hard to compute. By Theorem 4.1 it is enough to prove that $(b(n))$ is definable in $\mathsf{CH/poly}$. We already know that (2^n) and $(k!)$ are definable in CH (cf. Corollary 3.8). By applying Theorem 3.7 first for the division and then for the iterated sum, we conclude that $(b(n))$ is indeed definable in CH.

2. The binomial expansion $(3/2)^n = (1 + \frac{1}{2})^n = \sum_{k=0}^{n} \binom{n}{k} 2^{-k}$ yields

$$\lfloor (3/2)^n \rfloor = \sum_{k=0}^{n} \lfloor \frac{n(n-1)\cdots(n-k+1)}{k! 2^k} \rfloor + r(n)$$

for some integers $r(n)$ satisfying $0 \le r(n) \le n+1$. The assertion follows by arguing as for the first claim.

3. We start with the binomial series expansion

$$\frac{3}{4}\sqrt{2} = \sqrt{\frac{18}{16}} = \sqrt{1 + \frac{1}{8}} = \sum_{k=0}^{\infty} \binom{\frac{1}{2}}{k} 8^{-k} = \sum_{k=0}^{n-1} \binom{\frac{1}{2}}{k} 8^{-k} + \varepsilon_n.$$

The error ε_n can be expressed with Lagrange's formula for the function $f(x) = (1+x)^{1/2}$ as follows: for some $\xi_n \in (1, 9/8)$ we have (using $n! \ge (n/e)^n$)

$$|\varepsilon_n| = \frac{1}{n!} |f^{(n)}(\xi_n)|\, 8^{-n} = \frac{1}{n!}\ \frac{1 \cdot 3 \cdot 5 \cdots (2n-3)}{2^n} \frac{1}{(1+\xi)^{\frac{2n-1}{2}}} 8^{-n} \le \left(\frac{e}{8}\right)^n < \frac{3}{4} 2^{-n}.$$

This implies, for some integer $r(n)$ satisfying $0 \leq r(n) \leq n+1$,

$$\lfloor 2^n \sqrt{2} \rfloor = \sum_{k=0}^{n-1} \lfloor \frac{4}{3} \binom{\frac{1}{2}}{k} \frac{2^n}{8^k} \rfloor + r(n).$$

The sequence $\frac{4}{3}\binom{\frac{1}{2}}{k}\frac{2^n}{8^k} = \frac{1 \cdot 3 \cdot 5 \cdots (2n-3) \cdot 4 \cdot 2^n}{k! \cdot 3 \cdot 8^k}$ is definable in CH by Theorem 3.7. The assertion follows now as before. □

References

[1] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. Miltersen. On the complexity of numerical analysis. In *Proc. 21st Ann. IEEE Conference on Computational Complexity*, pages 331–339, 2006.

[2] E. Allender and K.W. Wagner. Counting hierarchies: polynomial times and constant depth circuits. In G. Rozenberg and A. Salomaa, editors, *Current trends in Theoretical Computer Science*, pages 469–483. World Scientific, 1993.

[3] P.W. Beame, S.A. Cook, and H.J. Hoover. Log depth circuits for division and related problems. *SIAM J. Comput.*, 15(4):994–1003, 1986.

[4] L. Blum, F. Cucker, M. Shub, and S. Smale. Algebraic Settings for the Problem "$P \neq NP$?". In *The mathematics of numerical analysis*, number 32 in Lectures in Applied Mathematics, pages 125–144. Amer. Math. Soc., 1996.

[5] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer, 1998.

[6] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.

[7] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics.* Springer Verlag, 2000.

[8] P. Bürgisser. Cook's versus Valiant's hypothesis. *Theoret. Comp. Sci.*, 235:71–88, 2000.

[9] P. Bürgisser. On implications between P-NP-hypotheses: Decision versus computation in algebraic complexity. In J. Sgall, A. Pultr, and P. Kolman, editors, *Proc. 26th MFCS*, number 2136 in LNCS, pages 3–17. Springer Verlag, 2001.

[10] P. Bürgisser, M. Clausen, and M.A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften.* Springer Verlag, 1997.

[11] Qi Cheng. Straight-line programs and torsion points on elliptic curves. *Comput. Complexity*, 12(3-4):150–161, 2003.

[12] W. de Melo and B. F. Svaiter. The cost of computing integers. *Proc. Amer. Math. Soc.*, 124(5):1377–1378, 1996.

[13] J. von zur Gathen and V. Strassen. Some polynomials that are hard to compute. *Theoret. Comp. Sci.*, 11:331–336, 1980.

[14] W. Hesse, E. Allender, and D.A. Barrrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *J. Comput. System Sci.*, 65(4):695–716, 2002. Special issue on complexity, 2001 (Chicago, IL).

[15] P. Koiran. Valiant's model and the cost of computing integers. *Comput. Complexity*, 13(3-4):131–146, 2004.

[16] R.J. Lipton. Straight-line complexity and integer factorization. In *Algorithmic number theory*, number 877 in LNCS, pages 71–79. Springer Verlag, 1994.

[17] G. Malod. *Polynômes et coefficients*. Phd thesis, Université Claude Bernard - Lyon 1, 2003. `http://tel.ccsd.cnrs.fr/tel-00087399`.

[18] L. Merel. Bornes pour la torsion des courbes elliptiques sur les corps de nombres. *Invent. Math.*, 124(1-3):437–449, 1996.

[19] J.H. Reif and S.R. Tate. On threshold circuits and polynomial computation. *SIAM J. Comput.*, 21(5):896–908, 1992.

[20] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "NP $\neq$ P?". *Duke Math. J.*, 81:47–54, 1995.

[21] S. Smale. Mathematical problems for the next century. In *Mathematics: frontiers and perspectives*, pages 271–294. Amer. Math. Soc., Providence, RI, 2000.

[22] V. Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM J. Comp.*, 3:128–149, 1974.

[23] V. Strassen. Einige Resultate über Berechnungskomplexität. *Jahr. Deutsch. Math. Ver.*, 78:1–8, 1976.

[24] V. Strassen. Algebraic complexity theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 11, pages 634–672. Elsevier Science Publishers B. V., Amsterdam, 1990.

[25] J. Torán. Complexity classes defined by counting quantifiers. *J. Assoc. Comput. Mach.*, 38(3):753–774, 1991.

[26] L.G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM STOC*, pages 249–261, 1979.

[27] L.G. Valiant. The complexity of computing the permanent. *Theoret. Comp. Sci.*, 8:189–201, 1979.

[28] L.G. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic: an International Symposium held in honor of Ernst Specker*, volume 30, pages 365–380. Monogr. No. 30 de l'Enseign. Math., 1982.

[29] H. Vollmer. *Introduction to circuit complexity*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 1999. A uniform approach.

[30] K.W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Inform.*, 23(3):325–356, 1986.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/