Circuit Satisfiability and Constraint Satisfaction around Skolem Arithmetic

Christian Glaßer[1], Peter Jonsson[2⋆], and Barnaby Martin[3⋆⋆]

[1] Theoretische Informatik, Julius-Maximilians-Universität, Würzburg, Germany
[2] Dept. of Computer and Information Science, Linköpings Universitet, SE-581 83 Linköping, Sweden
[3] School of Science and Technology, Middlesex University, The Burroughs, Hendon, London NW4 4BT

Abstract. We study interactions between Skolem Arithmetic and certain classes of Circuit Satisfiability and Constraint Satisfaction Problems (CSPs). We revisit results of Glaßer et al. [16] in the context of CSPs and settle the major open question from that paper, finding a certain satisfiability problem on circuits—involving complement, intersection, union and multiplication—to be decidable. This we prove using the decidability of Skolem Arithmetic. Then we solve a second question left open in [16] by proving a tight upper bound for the similar circuit satisfiability problem involving just intersection, union and multiplication. We continue by studying first-order expansions of Skolem Arithmetic without constants, $(\mathbb{N}; \times)$, as CSPs. We find already here a rich landscape of problems with non-trivial instances that are in P as well as those that are NP-complete.

1 Introduction

Skolem Arithmetic is the weak fragment of first-order arithmetic involving only multiplication. Thoralf Skolem gave a quantifier-elimination technique and argued for decidability of the theory in [28]. However, his proof was rather vague and a robust demonstration was not given of this result until Mostowski [23]. Skolem Arithmetic is somewhat less fashionable than *Presburger Arithmetic*, which involves only addition, and was proved decidable by Presburger in [26]. Indeed, Mostowski's proof made use of a reduction from Skolem Arithmetic to Presburger Arithmetic through the notion of weak direct powers (an excellent survey on these topics is [3]). The central thread of this paper is putting to work results about Skolem Arithmetic from the past, to solve open and naturally arising problems from today. Many of our results, like that of Mostowski, will rely on the interplay between Skolem and Presburger Arithmetic.

A *constraint satisfaction problem* (CSP) is a computational problem in which the input consists of a finite set of variables and a finite set of constraints, and

⋆ The second author was partially supported by the *Swedish Research Council* (VR) under grant 621-2012-3239.
⋆⋆ The third author was supported by EPSRC grant EP/L005654/1.

where the question is whether there exists a mapping from the variables to some fixed domain such that all the constraints are satisfied. When the domain is finite, and arbitrary constraints are permitted in the input, the CSP is NP-complete. However, when only constraints from a restricted set of relations are allowed in the input, it can be possible to solve the CSP in polynomial time. The set of relations that is allowed to formulate the constraints in the input is often called the *constraint language*. The question which constraint languages give rise to polynomial-time solvable CSPs has been the topic of intensive research over the past years. It has been conjectured by Feder and Vardi [13] that CSPs for constraint languages over finite domains have a complexity dichotomy: they are either in P or NP-complete. This conjecture remains unsettled, although dichotomy is now known on substantial classes (e.g. structures with domains of size ≤ 3 [27, 9] and smooth digraphs [17, 2]). Various methods, combinatorial (graph-theoretic), logical and universal-algebraic have been brought to bear on this classification project, with many remarkable consequences. A conjectured delineation for the dichotomy was given in the algebraic language in [10].

By now the literature on infinite-domain CSPs is also beginning to mature. Here the complexity can be much higher (e.g. undecidable) but on natural classes there is often the potential for structured classifications, and this has proved to be the case for reducts of, e.g. the rationals with order [5], the random (Rado) graph [7] and the integers with successor [6]; as well as first-order (fo) expansions of linear program feasibility [4]. Skolem and Presburger Arithmetic represent perfect candidates for continuation in this vein. These natural classes around Skolem and Presburger Arithmetic have the property that their CSPs sit in NP and a topic of recent interest for the second and third authors has been natural CSPs sitting in higher complexity classes.

Meanwhile, a literature existed on satisfiability of circuit problems over sets of integers involving work of the first author [16], itself continuing a line of investigation begun in [30] and pursued in [32, 33, 22]. The problems in [16] can be seen as variants of certain functional CSPs whose domain is all singleton sets of the non-negative integers and whose relations are set operations of the form: complement, intersection, union, addition and multiplication (the latter two are defined set-wise, e.g. $A \times B := \{ab : a \in A \wedge b \in B\}$). An open problem was the complexity of the problem when the permitted set operators were precisely complement, intersection, union and multiplication. In this paper we resolve that this problem is in fact decidable, indeed in triple exponential space. We prove this result by using the decidability of the theory of Skolem Arithmetic with constants. We take here Skolem Arithmetic to be the non-negative integers with multiplication (and possibly constants). In studying this problem we are able to bring to light existing results of [16] as results about their related CSPs, providing natural examples with interesting super-NP complexities. In addition, we improve one of the upper bounds of [16] to a tight upper bound. This is the circuit satisfiability problem where the permitted set operators are just intersection, union and multiplication, and where we improve the bound from NEXP to

PSPACE. Interestingly, this result does not immediately translate to a similar upper bound for the corresponding functional CSP.

In the second part of the paper, Skolem Arithmetic takes centre stage as we initiate the study of the computational complexity of the CSPs of its reducts, i.e. those constraint languages whose relations have a fo-definition in $(\mathbb{N}; \times)$. $\mathrm{CSP}(\mathbb{N}; \times)$ is in P, indeed it is trivial. The object therefore of our early study is its fo-expansions. We show that $\mathrm{CSP}(\mathbb{N}; +, \neq)$ is NP-complete, as is $\mathrm{CSP}(\mathbb{N}; \times, c)$ for each $c > 1$. We further show that $\mathrm{CSP}(\mathbb{N}; \times, U)$ is NP-complete when U is any non-empty set of integers greater than 1 such that each has a prime factor p, for some prime p, but omits the factor p^2. Clearly, $\mathrm{CSP}(\mathbb{N}; \times, U)$ is in P (and is trivial) if U contains 0 or 1. As a counterpoint to our NP-hardness results, we prove that $\mathrm{CSP}(\mathbb{N}; \times, U)$ is in P whenever there exists $m > 1$ so that $U \supseteq \{m, m^2, m^3, \ldots\}$.

Related work. Apart from the research on circuit problems mentioned above there has been work on other variants like circuits over integers [31] and positive natural numbers [8], equivalence problems for circuits [15], functions computed by circuits [25], and equations over sets of natural numbers [18, 19].

2 Preliminaries

Let $\mathbb{N}$ be the set of non-negative integers, and let $\mathbb{N}^+$ be the set of positive integers. For $m \in \mathbb{N}$, let Div_m be the set of factors of m. Finally, let $\{\mathbb{N}\}$ be the set of singletons $\{\{x\} : n \in \mathbb{N}\}$. In this paper we use a version of the CSP permitting both relations and functions (and constants). Thus, a *constraint language* consists of a domain together with functions, relations and constants over that domain. One may thus consider a constraint language to be a first-order structure. A *homomorphism* from a constraint language Γ to a constraint language Δ, over the same signature, is a function f from the domain of Γ to the domain of Δ that preserves the relations, i.e. if $(x_1, \ldots, x_k) \in R^\Gamma$, then also $(f(x_1), \ldots, f(x_k)) \in R^\Delta$. A homomorphism from a constraint language to itself is an *endomorphism*. An endomorphism that also preserves the negations of relations is termed an *embedding* and a bijective embedding is an *automorphism*.

A constraint language is a *core* if all of its endomorphisms are embeddings (equivalently, if the domain is finite, automorphisms). The functional version of the CSP has previously been seen in, e.g., [12]. For a purely functional constraint language, a *primitive positive* (pp) sentence is the existential quantification of a conjunction of term equalities. More generally, and when relations present, we may have positive atoms in this conjunction. The problem $\mathrm{CSP}(\Gamma)$ takes as input a primitive positive sentence φ, and asks whether it is true on Γ. The problem $\mathrm{CSP}^c(\Gamma)$ is similar but allows input constants naming the domain elements. We will allow that the functions involved on φ be defined on a larger domain than the domain of Γ. This is rather *unheimlich*[4] but it allows the problems of [16] to be more readily realised in the vicinity of CSPs. For example, one such typical

[4] Weird. Thus spake Lindemann about Hilbert's non-constructive methods in the resolution of Gordon's problem (see [29]).

domain is $\{\mathbb{N}\}$, but we will allow functions such as $^-$ (complement), $\cup$ (union) and $\cap$ (intersection) whose domain and range is the set of all subsets of $\mathbb{N}$. We will also employ the operations of set-wise addition $A+B := \{a+b : a \in A \wedge b \in B\}$ and multiplication $A \times B := \{ab : a \in A \wedge b \in B\}$.

Σ_i^{P}, Π_i^{P}, and Δ_i^{P} are levels of the polynomial-time hierarchy, while Σ_i, Π_i, and Δ_i are levels of the arithmetical hierarchy. Moreover, we use the classes $\mathrm{NP} = \Sigma_1^{\mathrm{P}}$, $\mathrm{PSPACE} = \bigcup_{k\geq 1} \mathrm{DSPACE}(n^k)$, and $3\mathrm{EXPSPACE} = \bigcup_{k\geq 1} \mathrm{DSPACE}(2^{2^{2^{n^k}}})$. Where no SPACE is written explicitly, the complexity classes may be assumed to refer to time. For more on these complexity classes we refer the reader to [24].

For sets A and B we say that A is *polynomial-time many-one reducible* to B, in symbols $A \leq_{\mathrm{m}}^{\mathrm{p}} B$, if there exists a polynomial-time computable function f such that for all x it holds that $(x \in A \iff f(x) \in B)$. If f is even computable in logarithmic space, then A is *logspace many-one reducible* to B, in symbols $A \leq_{\mathrm{m}}^{\log} B$. A is *nondeterministic polynomial-time many-one reducible* to B, in symbols $A \leq_{\mathrm{m}}^{\mathrm{NP}} B$, if there is a nondeterministic Turing transducer M that runs in polynomial time such that for all x it holds that $x \in A$ if and only if there exists a y computed by M on input x with $y \in B$. The reducibility notions $\leq_{\mathrm{m}}^{\mathrm{p}}$, $\leq_{\mathrm{m}}^{\log}$, and $\leq_{\mathrm{m}}^{\mathrm{NP}}$ are transitive and NP is closed under these reducibilities.

A *circuit* $C = (V, E, g_C)$ is a finite, non-empty, directed, acyclic multi-graph (V, E) with a specified node $g_C \in V$. The graph does not need to be connected and only has multiple edges between two nodes when a binary operator is applied on both sides to a single set (e.g. $A \times A$). Let $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. The nodes in the graph (V, E) are topologically ordered, i.e., for all $v_1, v_2 \in V$, if $v_1 < v_2$, then there is no path from v_2 to v_1. Nodes are also called *gates*. Nodes with indegree 0 are called *input gates* and g_C is called the *output gate*. If there is an edge from gate u to gate v, then we say that u is a *predecessor* of v and v is a *successor* of u.

Let $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, \times\}$. An *$\mathcal{O}$-circuit with unassigned input gates* $C = (V, E, g_C, \alpha)$ is a circuit (V, E, g_C) whose gates are labeled by the labeling function $\alpha : V \to \mathcal{O} \cup \mathbb{N} \cup \{\star\}$ such that the following holds: Each gate has an indegree in $\{0, 1, 2\}$, gates with indegree 0 have labels from $\mathbb{N} \cup \{\star\}$, gates with indegree 1 have label $^-$, and gates with indegree 2 have labels from $\{\cup, \cap, +, \times\}$. Input gates with a label from $\mathbb{N}$ are called *assigned* (or constant) input gates; input gates with label $\star$ are called *unassigned* (or variable) input gates. An *$\mathcal{O}$-formula* is an $\mathcal{O}$-circuit that only contains nodes with outdegree one.

Let $u_1 < \cdots < u_n$ be the unassigned inputs in C and $x_1, \ldots, x_n \in \mathbb{N}$. By assigning value x_i to the input u_i, we obtain an *$\mathcal{O}$-circuit* $C(x_1, \ldots, x_n)$ whose input gates are all assigned. In this circuit, each gate g computes the following set $I(g)$: If g is an assigned input gate where $\alpha(g) \neq \star$, then $I(g) = \{\alpha(g)\}$. If $g = u_k$ is an unassigned input gate, then $I(g) = \{x_k\}$. If g has label $^-$ and predecessor g_1, then $I(g) = \mathbb{N} \setminus I(g_1)$. If g has label $\circ \in \{\cup, \cap, +, \times\}$ and predecessors g_1 and g_2, then $I(g) = I(g_1) \circ I(g_2)$. Finally, let $I(C(x_1, \ldots, x_n)) = I(g_C)$ be the set computed by the circuit $C(x_1, \ldots, x_n)$.

Definition 1 (membership, equivalence, and satisfiability problems of circuits and formulas).

Let $\mathcal{O} \subseteq \{\cup, \cap, {}^{-}, +, \times\}$.

$$\mathrm{MC}_{\mathbb{N}}(\mathcal{O}) = \{(C, b) \mid C \text{ is an } \mathcal{O}\text{-circuit without unassigned inputs and } b \in I(C)\}$$
$$\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) = \{(C_1, C_2) \mid C_1 \text{ and } C_2 \text{ are } \mathcal{O}\text{-circuits without unassigned inputs and we have } I(C_1) = I(C_2)\}$$
$$\mathrm{SC}_{\mathbb{N}}(\mathcal{O}) = \{(C, b) \mid C \text{ is an } \mathcal{O}\text{-circuit with unassigned inputs } u_1 < \cdots < u_n \text{ and there exist } x_1, \ldots, x_n \in \mathbb{N} \text{ such that } b \in I\big(C(x_1, \ldots, x_n)\big)\}$$

$\mathrm{MF}_{\mathbb{N}}(\mathcal{O})$, $\mathrm{EF}_{\mathbb{N}}(\mathcal{O})$, *and* $\mathrm{SF}_{\mathbb{N}}(\mathcal{O})$ *are the variants that deal with* $\mathcal{O}$*-formulas instead of* $\mathcal{O}$*-circuits.*

When an $\mathcal{O}$-circuit is used as input for an algorithm, then we use a suitable encoding such that it is possible to verify in deterministic logarithmic space whether a given string encodes a valid circuit.

In Section 3, for $i \in \mathbb{N}$, we often identify $\{i\}$ with i, where this can not cause a harmful confusion.

3 Circuit Satisfiability and Functional CSPs

We investigate the computational complexity of functional CSPs. In many cases we can translate known lower and upper bounds for membership, equivalence, and satisfiability problems of arithmetic circuits [22, 15, 16] to CSPs. Our main result is the decidability of $\mathrm{SC}_{\mathbb{N}}({}^{-}, \cup, \cap, \times)$ and $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cup, \cap, \times)$, which solves the main open question of the paper [16]. We emphasise that the domain of $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cup, \cap, \times)$ is the set of singletons that we defined as $\{\mathbb{N}\}$ and not, e.g., the set of subsets of all natural numbers. This would be a different CSP. Our unusual definition is motivated by the circuit problems whose relationship to CSPs we wish to formalise.

We start with the observation that the equivalence of arithmetic terms reduces to functional CSPs. This yields several lower bounds for the CSPs.

Proposition 1. *For* $\mathcal{O} \subseteq \{{}^{-}, \cup, \cap, +, \times\}$ *it holds that* $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \mathcal{O})$.

Corollary 1.

1. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cup, \cap, +)$ *and* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cup, \cap, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* PSPACE.
2. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \cap, +)$, $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \cap, \times)$, $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, +)$, *and* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* Π_2^{P}.

CSPs with $+$ and $\times$ can express diophantine equations, which implies the Turing-hardness of such CSPs.

Proposition 2. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, \times)$ *is* $\leq_{\mathrm{m}}^{\log}$*-hard for* Σ_1, $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \cap, +, \times) \in \Sigma_1$ *and* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cup, \cap, +, \times) \in \Sigma_2$.

We now show that the decidability of Skolem arithmetic [14] can be used to decide the satisfiability of arithmetic circuits without $+$. From this we obtain the decidability of CSPs where exactly one arithmetic operation is forbidden.

Theorem 1. $\mathrm{SC}_{\mathbb{N}}(^{-}, \cup, \cap, \times)$, $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; ^{-}, \cup, \cap, \times)$ *and* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; ^{-}, \cup, \cap, +)$ *are in* 3EXPSPACE.

The following proposition transfers the NP-hardness from satisfiability problems for arithmetic circuits to $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \times)$ and $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +)$.

Proposition 3. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \times)$ *and* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* NP.

The remaining results in this section show that certain functional CSPs belong to NP. This needs non-trivial arguments of the form: If a CSP can be satisfied, then it can be satisfied even with small values. These arguments are provided by the known results that integer programs, existential Presburger arithmetic, and existential Skolem arithmetic are decidable in NP.

Proposition 4. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; ^{-}, \cap, \cup)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* NP.

Proposition 5. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +) \in \mathrm{NP}$.

Proposition 6. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +) \leq_{\mathrm{m}}^{\mathrm{NP}} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, =, \neq)$ *and* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, \times) \leq_{\mathrm{m}}^{\mathrm{NP}} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \times, =, \neq)$. *Therefore,* $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +), \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, \times) \in \mathrm{NP}$.

A second open problem from [16]. We now improve another of the upper bounds of [16] to a tight upper bound. Here we have the circuit satisfiability problem where the permitted set operators are just intersection, union and multiplication, where we improve the bound from NEXP to PSPACE.

Theorem 2. $\mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times) \in \mathrm{PSPACE}$.

Table 1 summarizes the results obtained in Section 3 and shows open questions. In particular, we would like to improve the gap between the lower and upper bounds for $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \mathcal{O})$, where $\mathcal{O}$ contains $\cup$ and exactly one arithmetic operation ($+$ or $\times$).

4 CSPs over fo-expansions of Skolem Arithmetic

We now commence our exploration of the complexity of CSPs generated from the simplest expansions of $(\mathbb{N}; \times)$. Abandoning our set-wise definitions, we henceforth use $\times$ to refer to the syntactic multiplication of Skolem Arithmetic (which may additionally carry semantic content). When we wish to refer to multiplication in a purely semantic way, we prefer $\cdot$s or $\prod$. We will consider $\times$ as a ternary relation rather than a binary function. We will never use syntactic $\times$ in a non-standard way, i.e. holding on a triple of integers for which it does not already hold in natural arithmetic.

	$\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\mathcal{O})$	
$\mathcal{O}$	Lower Bound	Upper Bound
$^{-}\ \cup\ \cap\ +\ \times$	Σ_1	Σ_2
$^{-}\ \cup\ \cap\ +$	PSPACE	3EXPSPACE
$^{-}\ \cup\ \cap\ \ \times$	PSPACE	3EXPSPACE
$^{-}\ \cup\ \cap$	NP	NP
$\cup\ \cap\ +\ \times$	Σ_1	Σ_1
$\cup\ \cap\ +$	Π_2^{P}	3EXPSPACE
$\cup\ \cap\ \ \times$	Π_2^{P}	3EXPSPACE
$\cup\ \ +\ \times$	Σ_1	Σ_1
$\cup\ \ +$	Π_2^{P}	3EXPSPACE
$\cup\ \ \ \times$	Π_2^{P}	3EXPSPACE
$\cap\ +\ \times$	Σ_1	Σ_1
$\cap\ +$	NP	NP
$\cap\ \ \times$	NP	NP
$+\ \times$	Σ_1	Σ_1
$+$	NP	NP
$\times$	NP	NP

Table 1. Upper and lower bounds for $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\mathcal{O})$. All lower bounds are with respect to $\leq_{\mathrm{m}}^{\log}$-reductions.

Proposition 7. *Let Γ be a finite signature reduct of $(\mathbb{N};\times,1,2,\ldots)$. Then $\mathrm{CSP}(\Gamma)$ is in* NP*.*

Upper bounds. We continue with polynomial upper bounds. Note that constants are no longer assumed to necessary exist in our structures (in contrast to the situation in Proposition 7).

Lemma 1. *Let $U \subseteq N$ be non-empty and $U \cap \{0,1\} = \emptyset$. Then $\mathrm{CSP}(\mathbb{N};\times,U)$ is polynomial-time reducible to $\mathrm{CSP}(\mathbb{N}^+;\times,U)$.*

We now borrow the following slight simplification of Lemma 6 from [20].

Lemma 2 (Scalability [20]). *Let Γ be a finite signature constraint language with domain $\mathbb{R}$, whose relations are quantifier-free definable in $+,\leq$ and $<$, such that the following holds.*

- *Every satisfiable instance of $\mathrm{CSP}(\Gamma)$ is satisfied by some rational point.*
- *For each relation $R \in \Gamma$, it holds that if $\overline{x} := (x_1, x_2, \ldots, x_k) \in R$, then $(ax_1, ax_2, \ldots, ax_k) \in R$ for all $a \in \{y : y \in \mathbb{R}, y \geq 1\}$.*
- *$\mathrm{CSP}(\Gamma)$ is in* P*.*

Then $\mathrm{CSP}(\Delta)$ is in P*, where Δ is obtained from Γ by substituting the domain $\mathbb{R}$ by $\mathbb{Z}$.*

Lemma 3. *Arbitrarily choose $m > 1$ and $U \subseteq \mathbb{N}^+$ such that $\{m, m^2, m^3, \ldots\} \subseteq U$. Then, $\mathrm{CSP}(\mathbb{N}^+;\times,U)$ is in* P*.*

Proposition 8. *Arbitrarily choose $m > 1$ and $U \subseteq \mathbb{N}$ such that $\{m, m^2, m^3, \ldots\} \subseteq U$. Then,* $\mathrm{CSP}(\mathbb{N}; \times, U)$ *is in* P.

Cores. We say that an integer $m > 1$ has a *degree-one factor* p if and only if p is a prime such that $p|m$ and $p^2 \not| m$. Let Div_m be the set of divisors of m, pp-definable in $(\mathbb{N}; \times, m)$ by $\exists y\ x \times y = m$. We can pp-define the relation $\{1\}$ in $(\mathrm{Div}_m; \times, m)$ since $x = 1$ iff $x \times x = x$ (recalling $0 \notin \mathrm{Div}_m$). It follows that $\{1, m\}$ are contained in the core of $(\mathrm{Div}_m; \times, m)$.

Lemma 4. *Let $m > 1$ be an integer that has a degree-one factor p. Then $(\mathrm{Div}_m; \times, m)$ has a two-element core.*

Lemma 5. *Let m be an integer that does not have a degree-one factor. Then $(\mathrm{Div}_m; \times, m)$ does not have a two-element core.*

Lower bounds. We now move to lower bounds of NP-completeness.

Proposition 9. $\mathrm{CSP}(\mathbb{N}; \neq, \times)$ *is* NP*-complete.*

An operation $t : D^k \to D$ is a *weak near-unanimity* operation if t is idempotent and satisfies $t(y, x, \ldots, x) = t(x, y, x, \ldots, x) = \cdots = t(x, \ldots, x, y)$.

Theorem 3 ([1]). *Let Γ be a constraint language over a finite set D. If Γ is a core and does not have a weak near-unanimity polymorphism, then* $\mathrm{CSP}(\Gamma)$ *is* NP*-hard.*

Lemma 6. *Arbitrarily choose an $m > 1$ such that $m \neq k^n$ for all $k, n > 1$ together with a finite set $\{1, m\} \subseteq S \subseteq \mathbb{N} \setminus \{0\}$. If $(S; \times, m)$ is a core, then* $\mathrm{CSP}(S; \times, m)$ *is* NP*-hard.*

Note that the proof of this last lemma is made easier by our assumption that $\times$ is a relation and not a function. Were it a function we would need to prove the domain S is closed under it.

Theorem 4. $\mathrm{CSP}(\mathbb{N}; \times, m)$ *is* NP*-hard for every integer $m > 1$.*

Theorem 5. *Let U be any subset of $\mathbb{N} \setminus \{0, 1\}$ so that every $x \in U$ has a degree-one factor. Then* $\mathrm{CSP}(\mathbb{N}; \times, U)$ *is* NP*-hard.*

For $x \in \mathbb{N} \setminus \{0, 1\}$, define its *minimal exponent*, $\text{min-exp}(x)$, to be the smallest j such that x has a factor of p^j, for some prime p, but not a factor of p^{j+1}. Thus an integer with a degree-one factor has minimal exponent 1. Call $x \in \mathbb{N} \setminus \{0, 1\}$ *square-free* if it omits all repeated prime factors. For a set $U \subseteq \mathbb{N} \setminus \{0, 1\}$, define its *basis*, $\mathrm{basis}(U)$ to be the set $\{\text{min-exp}(x) : x \in U\}$.

Lemma 7. *Let $U \subseteq \mathbb{N} \setminus \{0, 1\}$, so that* $\mathrm{basis}(U)$ *is finite and* $\mathrm{basis}(U) \neq \{1\}$. *There is some set X pp-definable in $(\mathbb{N}; \times, U)$ so that* $\mathrm{basis}(X) = \{1\}$.

Theorem 6. *Let $U \subseteq \mathbb{N} \setminus \{0, 1\}$ be so that* $\mathrm{basis}(U)$ *is finite. Then* $\mathrm{CSP}(\mathbb{N}; \times, U)$ *is* NP*-complete.*

5 Final remarks

There are two major directions in which more work is necessary.

A perfunctory glance at the results of Section 3 shows that some of our bounds are not tight, and it would be great to see some natural CSPs in this region manifesting complexities such as PSPACE-complete. It is informative to compare our Table 1 with Table 1 in [16]. Our weird formulation of these CSPs belies the fact there are more natural versions where, for $\mathcal{O} \subseteq \{^-, \cap, \cup, +, \times\}$, we ask about $\mathrm{CSP}(\mathcal{P}(\mathbb{N}); \mathcal{O})$, where $\mathcal{P}(\mathbb{N})$ is the power set of $\mathbb{N}$, rather than the somewhat esoteric $\mathrm{CSP}(\{\mathbb{N}\}; \mathcal{O})$. Indeed, if we replace complement "$^-$" by set difference "$\setminus$", these questions could also be phrased for just the finite sets of $\mathcal{P}(\mathbb{N})$ (see recent work [11]).

Meanwhile, the results of Section 4 need to be extended to a classification of complexity for all $\mathrm{CSP}(\Gamma)$, where Γ is a reduct of Skolem Arithmetic $(\mathbb{N}; \times)$. We anticipate the first stage is to complete the classification for $\mathrm{CSP}(\mathbb{N}; \times, U)$ where U is fo-definable in $(\mathbb{N}; \times)$.

References

1. Barto, L., and Kozik, M. Constraint satisfaction problems of bounded width. In *FOCS* (2009), pp. 595–603.
2. Barto, L., Kozik, M., and Niven, T. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM Journal on Computing 38*, 5 (2009), 1782–1802.
3. Bès, A. A survey of arithmetical definability, 2002.
4. Bodirsky, M., Jonsson, P., and von Oertzen, T. Essential convexity and complexity of semi-algebraic constraints. *Logical Methods in Computer Science 8*, 4 (2012). Extended abstract titled *Semilinear Program Feasibility* at ICALP'10.
5. Bodirsky, M., and Kára, J. The complexity of temporal constraint satisfaction problems. *J. ACM 57*, 2 (2010).
6. Bodirsky, M., Martin, B., and Mottet, A. Constraint satisfaction problems over the integers with successor. In *ICALP I*, (2015), pp. 256–267.
7. Bodirsky, M., and Pinsker, M. Schaefer's theorem for graphs. In *Proceedings of STOC'11* (2011), pp. 655–664. Preprint of the long version available at arxiv.org/abs/1011.2894.
8. Breunig, H. The complexity of membership problems for circuits over sets of positive numbers. In *FCT* (2007), vol. 4639 of *Lecture Notes in Computer Science*, Springer, pp. 125–136.
9. Bulatov, A. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM 53*, 1 (2006), 66–120.
10. Bulatov, A., Krokhin, A., and Jeavons, P. G. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing 34* (2005), 720–742.
11. Dose, T. Complexity of Constraint Satisfaction Problems over Finite Subsets of Natural Numbers. *ECCC* (2016).
12. Feder, T., Madelaine, F. R., and Stewart, I. A. Dichotomies for classes of homomorphism problems involving unary functions. *Theor. Comput. Sci. 314*, 1-2 (2004), 1–43.

13. Feder, T., and Vardi, M. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing 28* (1999), 57–104.
14. Ferrante, J., and Rackoff, C. W. The computational complexity of logical theories. vol. 718 of *Lecture Notes in Mathematics*, Springer Verlag, 1979.
15. Glasser, C., Herr, K., Reitwiessner, C., Travers, S. D., and Waldherr, M. Equivalence problems for circuits over sets of natural numbers. *Theory of Computing Systems 46*, 1 (2010), 80–103.
16. Glasser, C., Reitwiessner, C., Travers, S. D., and Waldherr, M. Satisfiability of algebraic circuits over sets of natural numbers. *Discrete Applied Mathematics 158*, 13 (2010), 1394–1403.
17. Hell, P., and Nešetřil, J. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B 48* (1990), 92–110.
18. Jez, A., and Okhotin, A. Complexity of equations over sets of natural numbers. *Theoretical Computer Science 48*, 2 (2011), 319–342.
19. Jez, A., and Okhotin, A. Computational completeness of equations over sets of natural numbers. *Information and Computation 237* (2014), 56–94.
20. Jonsson, P., and Lööw, T. Computational complexity of linear constraints over the integers. *Artificial Intelligence 195* (2013), 44–62. An extended abstract appeared at IJCAI 2011.
21. Matiyasevich, Y. V. Enumerable sets are diophantine. *Doklady Akad. Nauk SSSR 191* (1970), 279–282. Translation in Soviet Math. Doklady, 11:354–357, 1970.
22. McKenzie, P., and Wagner, K. W. The complexity of membership problems for circuits over sets of natural numbers. *Computational Complexity 16*, 3 (2007), 211–244. Extended abstract appeared at STACS 2003.
23. Mostowski, A. On direct products of theories. *The Journal of Symbolic Logic 17*, 3 (1952), 1–31.
24. Papadimitriou, C. H. *Computational Complexity*. Addison-Wesley, 1994.
25. Pratt-Hartmann, I., and Düntsch, I. Functions definable by arithmetic circuits. In *Conference on Mathematical Theory and Computational Practice* (2009), vol. 5635 of *Lecture Notes in Computer Science*, Springer, pp. 409–418.
26. Presburger, M. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes Rendus du I congres de Mathématiciens des Pays Slaves* (1929), 92–101.
27. Schaefer, T. J. The complexity of satisfiability problems. In *Proceedings of STOC'78* (1978), pp. 216–226.
28. Skolem, T. Über gewisse satzfunktionen in der arithmetik. *Skr. Norske Videnskaps-Akademie i Oslo* (1930).
29. Smorynski, C. The incompleteness theorems. In *Handbook of Mathematical Logic*, J. Barwise, Ed. North-Holland, Amsterdam, 1977, pp. 821–865.
30. Stockmeyer, L. J., and Meyer, A. R. Word problems requiring exponential time: Preliminary report. In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, (STOC)* (1973), pp. 1–9.
31. Travers, S. D. The complexity of membership problems for circuits over sets of integers. *Theoretical Computer Science 369*, 1-3 (2006), 211–229.
32. Wagner, K. The complexity of problems concerning graphs with regularities. In *MFCS* (1984), pp. 544–552.
33. Yang, K. Integer circuit evaluation is Pspace-complete. *J. Comput. Syst. Sci. 63*, 2 (2001), 288–303. An extended abstract of appeared at CCC 2000.