Circuit Satisfiability and Constraint Satisfaction around Skolem Arithmetic[☆]

Christian Glaßer[a], Peter Jonsson[b,*], Barnaby Martin[c,**]

[a]*Theoretische Informatik, Julius-Maximilians-Universität, Würzburg, Germany*
[b]*Department of Computer and Information Science, Linköpings Universitet, SE-581 83 Linköping, Sweden*
[c]*School of Engineering & Computing Sciences, Durham University, Durham, UK.*

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. [1] 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 [1] 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.

Keywords: Circuit Satisfiability, Constraint Satisfaction, Skolem Arithmetic, Computational Complexity

[☆] An extended abstract of this paper appeared at the 12th Conference on Computability in Europe (CiE) 2016.

[*] 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.

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 [2]. However, his proof was rather vague and a robust demonstration was not given of this result until Mostowski [3]. Skolem Arithmetic is somewhat less fashionable than *Presburger Arithmetic*, which involves only addition, and was proved decidable by Presburger in [4]. 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 [5]). 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 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. When the structure of the variables within the constraints in the instance is restricted or the constraints come from a restricted set of relations, it can be possible to solve the CSP in polynomial time. Structural restrictions on the variables usually take the form of some restriction to the *Gaifman Graph* built from the variables as vertices, with an edge between variables if they appear in a single constraint. When the CSP has instances whose corresponding Gaifman Graph comes from a class of graphs of bounded-treewidth (we will not define this class save to say its members are somewhat tree-like), then this restricted CSP can be solved in polynomial time [6]. This result is tight, under a certain assumption from Parameterized Complexity Theory [7]. 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 [8] that CSPs for constraint languages over finite domains have a complexity dichotomy: they are either in P or NP-complete (it is known they are always in NP). This conjecture is now known on substantial classes (e.g. structures with domains of size ≤ 3 [9, 10] and smooth digraphs [11, 12]). Further, there are three papers on arxiv claiming to prove it in full generality [13, 14, 15]. 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 [16], and it is this version of the Feder-Vardi Conjecture which the three papers claim to settle.

By now the literature on infinite-domain CSPs is also beginning to mature. Here the complexity can be much higher (e.g. undecidable [17]) but on natural classes there is often the potential for structured classifications, and this has proved to be the case for reducts of (i.e. structures whose relations admit a first-order definition in), e.g., the rationals with order [18], the random (Rado) graph [19] and the integers with successor [20]; as well as first-order expansions of linear program feasibility [21]. Skolem and Presburger Arithmetic represent perfect candidates for continuation in this vein. These natural classes around Skolem and Presburger Arithmetic have the property that, for constraint languages that are their reducts, the corresponding CSPs sit in NP.

Meanwhile, a literature existed on satisfiability of circuit problems over sets of integers involving work of the first author [1], itself continuing a line of investigation begun in [22] and pursued in [23, 24, 25]. The circuits typically compute some set of integers at their unique output node and one asks for satisfiability in terms of evaluations of free set-variables at their input nodes. The problems in [1] 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. $A \times B := \{ab : a \in A \wedge b \in B\}$). Here, by functional CSPs we mean CSPs over a functional signature, and not CSPs that compute a function (for example, a satisfying assignment or counting the number of satisfying assignments). An open problem in the area 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 [1] as results about their related CSPs, providing natural examples with interesting super-NP complexities (remember the constraint language here is not a reduct of Skolem Arithmetic but rather built from singleton sets of non-negative integers). In addition, we improve one of the upper bounds of [1] 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. For minor technical reasons which we will come back to, we here consider $\times$ to be a ternary relation, rather than a binary function.[1] $\mathrm{CSP}(\mathbb{N}; \times)$, that is the problem of model-checking over Skolem Arithmetic a positive first-order sentence involving just $\exists$ and $\wedge$, is in P; indeed it is trivial. The object therefore of our early study is its first-order expansions. We show that $\mathrm{CSP}(\mathbb{N}; +, \neq)$ is NP-complete, as is $\mathrm{CSP}(\mathbb{N}; \times, c)$ for each $c > 1$. As an example of another non-trivial hard class, we 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 (Theorem 5). 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 [26] and positive natural numbers [27], equivalence problems for circuits [28], functions computed by circuits [29], and equations over sets of natural numbers [30, 31]. Typically, the complexity of membership of circuits is similar to the corresponding equivalence of circuits problem, though the latter may be slightly higher and belies some imperfect bounds[2]. The complexity of the satisfiability of circuits is generally higher than these other circuit problems. Some interesting recent work on the multiplicative theory of numbers appears

[1] In this paper $\times$ is overloaded, meaning both a binary function on sets and a ternary relation on non-negative integers. Since these are of distinct types, this should not cause too much confusion.

[2] Tables detailing these complexities can be found together at `https://en.wikipedia.org/wiki/Circuits_over_sets_of_natural_numbers`

in [32].

2. Preliminaries

Let $\mathbb{N}$ be the set of non-negative integers, and let $\mathbb{N}^+$ be the set of positive integers. When numbers appear as part of the input to a computational problem, we will always assume they are encoded in binary. For $m \in \mathbb{N}$, let Div_m be the set of factors of m. Finally, let $\{\mathbb{N}\}$ be the set of singletons $\{\{x\} : n \in \mathbb{N}\}$.

2.1. Constraint Satisfaction Problems

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, whose *signature* describes the arities of the relations and functions involved. 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). Every finite-domain constraint language has a unique induced substructure which is a core. For infinite-domain constraint languages, the situation is more complex [33], though often cores still do exist. The functional version of the CSP has previously been seen in, e.g., [34]. 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 are present, we may have positive atoms in this conjunction. The problem $\mathrm{CSP}(\Gamma)$ takes as input a pp sentence ϕ, and asks whether it is true on Γ. The problem $\mathrm{CSP}^{\mathrm{c}}(\Gamma)$ is the same except that we enrich Γ with constants naming the elements of the domain, that may now be used in pp sentences. In the finite-domain case, cores are important because they allow pp definition (i.e. in a pp formula) of the constants (at least up to automorphism). Again, in the infinite-domain case, the situation is more complex [33]. Definitions by pp formulas play a key role in the study of CSPs due to the following observation, that holds generally and not just in the finite-domain case, and which we will freely use without specific reference.

Proposition 1 ([35]). *If Γ is a constraint language and R is a relation pp definable in Γ, then there is a polynomial time reduction from CSP$(\Gamma; R)$ to CSP(Γ).*

In light of this proposition, we see that if Γ is finite-domain, then CSP(Γ) and CSP$^{c}(\Gamma)$ have the same complexity, up to polynomial time reductions. We will allow that the functions involved on ϕ be defined on a larger domain than the domain of Γ. This is rather *unheimlich*[3] but it allows the problems of [1] to be more readily realised in the vicinity of CSPs. For example, one such typical 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 $A \times B := \{ab : a \in A \wedge b \in B\}$. Our main results are for constraint languages with an infinite domain, though some of our subsidiary results involve those that are finite.

2.2. Computational Complexity

Definable sets of the *Arithmetical Hierarchy* are built from formulas in the language of Peano Arithmetic (see [37] for more on these), the union of Presburger and Skolem Arithmetics, involving both addition and multiplication. Formulas of first-order logic with only bounded quantification of the form $\exists x \leq z$ or $\forall x \leq z$ define $\Sigma_0 = \Pi_0$ sets at the base of the Arithmetical Hierarchy. These are the sets whose membership is decidable by a Turing Machine. Note that we conflate formulas and definable sets in the notation Σ_0 and Π_0. A first-order formula over Peano Arithmetic is considered Σ_{i+1} if it begins with a block of (bounded or unbounded) existential quantifiers and then alternates between (bounded or unbounded) universal and existential quantifiers i times before culminating with a $\Sigma_0 = \Pi_0$ formula. Again, we conflate formulas and definable sets to obtain levels Σ_{i+1} of the Arithmetical Hierarchy. Definable sets Π_{i+1} are defined similarly but with outermost universal quantification. Finally, we define $\Delta_i = \Sigma_i \cap \Pi_i$.

[3] Weird. Thus spake Lindemann about Hilbert's non-constructive methods in the resolution of Gordon's problem (see [36]).

The *Polynomial Hierarchy* PH (see [38]) is the subrecursive analog of the Arithmetical Hierarchy in which polynomial time P takes the role of the decidable sets $\Sigma_0 = \Pi_0$, and alternation corresponds to looking for a choice in a non-deterministic computation (existential) or considering all choices (universal). Hence we define $\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}$ and $\Sigma_1^{\mathrm{P}} = \mathrm{NP}$. The Polynomial Hierarchy sits within $\mathrm{PSPACE} = \bigcup_{k \geq 1} \mathrm{DSPACE}(n^k)$. Finally, we introduce the classes $2\mathrm{EXPSPACE} = \bigcup_{k \geq 1} \mathrm{DSPACE}(2^{2^{n^k}})$ and $3\mathrm{EXPSPACE} = \bigcup_{k \geq 1} \mathrm{DSPACE}(2^{2^{2^{n^k}}})$. Presburger Arithmetic is decidable in 2EXPSPACE [39] and Skolem Arithmetic is decidable in 3EXPSPACE [40]. Where no SPACE is written explicitly, the complexity classes may be assumed to refer to time.

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. For more on these complexity classes we refer the reader to [38].

2.3. Circuit Problems

A *circuit* $C = (V, E, g_C)$ is a finite, non-empty, directed, acyclic multigraph (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, i.e. 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. *Let $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, \times\}$.*

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

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

Note that $\mathrm{MC}_{\mathbb{N}}$, $\mathrm{EC}_{\mathbb{N}}$ and $\mathrm{SC}_{\mathbb{N}}$ intimate Membership of a Circuit, Equivalence of Circuits and Satisfiability of a Circuit, respectively. 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. The reader interested only in arithmetic circuits may jump to Section 3.2. In many cases we can translate known lower and upper bounds for membership, equivalence, and satisfiability problems of arithmetic circuits [25, 28,

	$\mathrm{CSP^c}(\{\mathbb{N}\}; \mathcal{O})$			
$\mathcal{O}$	Lower Bound		Upper Bound	
$^- \cup \cap + \times$	Σ_1	P3	Σ_2	P5
$^- \cup \cap +$	PSPACE	C1	3EXPSPACE	C4
$^- \cup \cap \times$	PSPACE	C1	3EXPSPACE	C3
$^- \cup \cap$	NP	P8	NP	P8
$\cup \cap + \times$	Σ_1	P3	Σ_1	P4
$\cup \cap +$	Π_2^{P}	C1	3EXPSPACE	C4
$\cup \cap \times$	Π_2^{P}	C1	3EXPSPACE	C3
$\cup + \times$	Σ_1	P3	Σ_1	P4
$\cup +$	Π_2^{P}	C1	3EXPSPACE	C4
$\cup \times$	Π_2^{P}	C1	3EXPSPACE	C3
$\cap + \times$	Σ_1	P3	Σ_1	P4
$\cap +$	NP	P7	NP	C2
$\cap \times$	NP	P6	NP	C2
$+ \times$	Σ_1	P3	Σ_1	P4
$+$	NP	P7	NP	P9
$\times$	NP	P6	NP	C2

Table 1: Upper and lower bounds for $\mathrm{CSP^c}(\{\mathbb{N}\}; \mathcal{O})$, together with the proposition or corollary in which they are proved. All lower bounds are with respect to $\leq_{\mathrm{m}}^{\log}$-reductions.

1] to CSPs. Our main result is the decidability of $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, \times)$ and $\mathrm{CSP^c}(\{\mathbb{N}\}; ^-, \cup, \cap, \times)$, which solves the main open question of the paper [1]. We emphasise that the domain of $\mathrm{CSP^c}(\{\mathbb{N}\}; ^-, \cup, \cap, \times)$ 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. Table 1 summarizes the results obtained in this section.

3.1. Recasting circuit results for CSPs

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

Proposition 2. *Whenever* $\mathcal{O} \subseteq \{^-, \cup, \cap, +, \times\}$ *it holds that* $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{CSP^c}(\{\mathbb{N}\}; \mathcal{O})$.

Proof. An $\mathrm{EF}_{\mathbb{N}}(\mathcal{O})$-instance (F_1, F_2) is mapped to the $\mathrm{CSP^c}(\{\mathbb{N}\}; \mathcal{O})$-instance $F_1 = F_2$.

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}*.*

Proof. The statements follow from Proposition 2 and the following facts [28]: $\mathrm{EF}_{\mathbb{N}}({}^-, \cup, \cap, +)$ and $\mathrm{EF}_{\mathbb{N}}({}^-, \cup, \cap, \times)$ are $\leq_{\mathrm{m}}^{\log}$-complete for PSPACE. $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, +)$, $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, \times)$, $\mathrm{EF}_{\mathbb{N}}(\cup, +)$, and $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$ are $\leq_{\mathrm{m}}^{\log}$-complete for Π_2^{P}.

CSPs with + and × can express Diophantine equations, which implies the hardness under (polynomial time) Turing reductions of such CSPs.

Proposition 3. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, \times)$ *is* $\leq_{\mathrm{m}}^{\log}$*-hard for* Σ_1*.*

Proof. By the Matiyasevich-Robinson-Davis-Putnam theorem [41, 42], there exists an $n \in \mathbb{N}$ and a multivariate polynomial p with integer coefficients such that for every $A \in \Sigma_1$ there exists an $a \in \mathbb{N}$ such that

$$x \in A \iff \exists y \in \mathbb{N}^n, p(a, x, y) = 0.$$

In the equation $p(a, x, y) = 0$ we can move negative monomials and negative constants to the right-hand side. This yields multivariate polynomials l and r with coefficients from $\mathbb{N}$ such that

$$x \in A \iff \exists y \in \mathbb{N}^n, l(a, x, y) = r(a, x, y).$$

The right-hand side is an instance of $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, \times)$. Hence, for every $A \in \Sigma_1$, $A \leq_{\mathrm{m}}^{\log} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, \times)$.

Proposition 4. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \cap, +, \times) \in \Sigma_1$.

Proof. It is decidable whether a given assignment satisfies an instance of $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \cap, +, \times)$. Hence testing the existence of a satisfying assignment is in Σ_1.

Proposition 5. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, +, \times) \in \Sigma_2$.

Proof. By Glaßer et al. [28], $\mathrm{EC}_{\mathbb{N}}({}^{-},\cup,\cap,+,\times) \in \Delta_2$. Consider an arbitrary $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};{}^{-},\cup,\cap,+,\times)$-instance $\phi := \exists y \in \mathbb{N}^n[t_0 = t_1 \wedge \cdots \wedge t_{2m} = t_{2m+1}]$, where each t_i is a term or (equivalently) a tree-like circuit. It holds that

$$\phi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};{}^{-},\cup,\cap,+,\times) \iff \exists y \in \mathbb{N}^n[\mathrm{EC}_{\mathbb{N}}(t_0,t_1)\wedge\cdots\wedge\mathrm{EC}_{\mathbb{N}}(t_{2m},t_{2m+1})].$$

The right-hand side is a Σ_2 predicate.

The following propositions transfer 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 6. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\times)$ *is* $\leq_{\mathrm{m}}^{\mathrm{log}}$*-hard for* NP.

Proof. It is known that $3\mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{log}} \mathrm{SC}_{\mathbb{N}}(\cap,\times)$ [1]. The reduction has the additional property that it outputs pairs (C,b) where the circuit C is connected in the sense that from each gate there exists a path to the output gate. Hence it suffices to construct a $\leq_{\mathrm{m}}^{\mathrm{log}}$-reduction that works on $\mathrm{SC}_{\mathbb{N}}(\cap,\times)$-instances (C,b) where C is connected.

For such a pair (C,b) we construct a $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\times)$-instance where each gate g is represented by the variable g. Moreover, each gate g causes the following constraints: If g is an assigned input gate with value $k \in \mathbb{N}$, then we add the constraint $g = k$. For unassigned input gates no additional constraints are needed. If g is a $\times$-gate with predecessors g_1 and g_2, then we add the constraint $g = g_1 \cdot g_2$. If g is a $\cap$-gate with predecessors g_1 and g_2, then we add the constraints $g = g_1$ and $g = g_2$. If g is the output gate, then this causes the additional constraint $g = b$. Finally, if $g_1,\ldots,g_n$ are the gates in C and $c_1,\ldots,c_m$ are the constraints described above, then the reduction outputs the $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\times)$-instance $\varphi := \exists g_1,\ldots,g_n[c_1 \wedge \cdots \wedge c_m]$.

It remains to argue that for connected C it holds that

$$(C,b) \in \mathrm{SC}_{\mathbb{N}}(\cap,\times) \iff \varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\times).$$

Assume $(C,b) \in \mathrm{SC}_{\mathbb{N}}(\cap,\times)$ and consider an assignment that produces $\{b\}$ at the output gate. Since C is connected, each gate g_i computes a singleton $\{a_i\}$. Hence $a_1,\ldots,a_n$ is a satisfying assignment for φ, which shows $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\times)$.

Assume $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};\times)$. Let $a_1,\ldots,a_n$ be a satisfying assignment for φ and let l be the number of C's input gates. The constraints in φ make sure that $C(a_1,\ldots,a_l)$ produces $\{a_i\}$ at gate g_i. In particular, $C(a_1,\ldots,a_l)$ produces $\{b\}$ at the output gate, which shows $(C,b) \in \mathrm{SC}_{\mathbb{N}}(\cap,\times)$.

Proposition 7. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};+)$ *is* $\leq_{\mathrm{m}}^{\log}$*-hard for* NP.

PROOF. It suffices to show $\mathrm{SC}_{\mathbb{N}}(\{+\}) \leq_{\mathrm{m}}^{\log} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};+)$ [1]. The proof is similar to the proof of Proposition 6, but easier, since we have no $\cap$-gates and hence we do not need the assumption that C is connected.

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

PROOF. Consider a $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};{}^{-},\cap,\cup)$-instance $\varphi := \exists x_1,\dots,x_n[t_1 = t_1' \wedge \cdots \wedge t_m = t_m']$. We show that if $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};{}^{-},\cap,\cup)$, then it has a satisfying assignment $d = (d_1,\dots,d_n)$ such that $d_1,\dots,d_n \in \{0,\dots,n-1\}$. Since $\mathrm{MC}_{\mathbb{N}}({}^{-},\cap,\cup) \in \mathrm{P}$ from [25], we deduce $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};{}^{-},\cap,\cup) \in \mathrm{NP}$.

Assume $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\};{}^{-},\cap,\cup)$ and choose a satisfying assignment $a = (a_1,\dots,a_n)$ such that $a' = \max\{a_1,\dots,a_n\}$ is minimal. Assume that $a' \geq n$, we will show a contradiction. Let $b' = \min(\mathbb{N} - \{a_1,\dots,a_n\})$ and note that $b' < n$ by a simple counting argument. Let b be the assignment that is obtained from a if all occurrences of a' are replaced with b'. For any term t in φ, the sets computed by t under the assignments a and b are denoted by t_a and t_b, respectively. Observe that for all terms t in φ and all $x \in \mathbb{N} - \{a', b'\}$ it holds that:

$$\begin{aligned} x \in t_a &\iff x \in t_b \\ a' \in t_a &\iff b' \in t_b \\ b' \in t_a &\iff a' \in t_b \end{aligned}$$

It follows that for all atoms $t = t'$ in φ it holds that

$$t_a = t'_a \iff t_b = t'_b.$$

Therefore, b is a satisfying assignment that is smaller than a, which contradicts the minimal choice of a.

For the NP-hardness it suffices to show 3SAT $\leq_{\mathrm{m}}^{\log}$ $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cap, \cup)$. On input of a 3CNF-formula $t = t(x_1, \ldots, x_n)$ the reduction outputs the instance of $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^{-}, \cap, \cup)$

$$\varphi := \exists x_1, \ldots, x_n [t' \cap \{1\} = \{1\}],$$

where t' is obtained from t by replacing $\neg, \wedge, \vee$ with ${}^{-}, \cap, \cup$, respectively. Every satisfying assignment for t also satisfies φ. Conversely, if $a_1, \ldots, a_n$ is a satisfying assignment for φ, then we obtain a satisfying assignment for t if values greater than 1 are replaced with 0.

Proposition 9. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +) \in \mathrm{NP}$.

PROOF. Consider a $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +)$-instance $\varphi := \exists x_1, \ldots, x_n [s_1 = t_1 \wedge \cdots \wedge s_m = t_m]$. Each atom $s_i = t_i$ can be written as $0 = t_i - s_i = a_{i,1}x_1 + \cdots + a_{i,n}x_n + c_i$ where $a_{i,j}, c_i \in \mathbb{Z}$. Let

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix} \quad \text{and} \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix}.$$

Hence $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +)$ if and only if there exists an $x = (x_1, \ldots, x_n) \in \mathbb{N}^n$ such that $Ax + c = 0$. The latter of these is an integer program that can be decided in NP [43, 44].

Proposition 10.

1. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +) \leq_{\mathrm{m}}^{\mathrm{NP}} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, =, \neq)$.
2. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, \times) \leq_{\mathrm{m}}^{\mathrm{NP}} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \times, =, \neq)$.

PROOF. We show the first statement, the proof of the second one is analogous.

For a term t, let t' be the term obtained from t if every subterm of the form $s_1 \cap s_2$ is replaced with s_1.

We describe the $\leq_{\mathrm{m}}^{\mathrm{NP}}$-reduction on input of a $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +)$-instance

$$\varphi := \exists x_1, \ldots, x_n [t_0 = t_1 \wedge \cdots \wedge t_{2m} = t_{2m+1}].$$

For each atom $t_{2i} = t_{2i+1}$, we guess nondeterministically whether $t_{2i} = t_{2i+1} \in \{\mathbb{N}\}$ or $t_{2i} = t_{2i+1} = \emptyset$. If we guessed $t_{2i} = t_{2i+1} \in \{\mathbb{N}\}$, then replace t_{2i} with

t'_{2i}, replace t_{2i+1} with t'_{2i+1}, and for every subterm $s_1 \cap s_2$ that appears in t_{2i} or t_{2i+1} add the constraint $s'_1 = s'_2$. If we guessed $t_{2i} = t_{2i+1} = \emptyset$, then guess a subterm $u_1 \cap u_2$ in t_{2i}, guess a subterm $u_3 \cap u_4$ in t_{2i+1}, remove the atom $t_{2i} = t_{2i+1}$, and add the constraints $u'_1 \neq u'_2$ and $u'_3 \neq u'_4$. The obtained formula ψ is the result of the $\leq_{\mathrm{m}}^{\mathrm{NP}}$-reduction.

We argue that the described $\leq_{\mathrm{m}}^{\mathrm{NP}}$-reduction reduces the $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +)$ to the $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, =, \neq)$.

Assume $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +)$ and fix some satisfying assignment $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$. Consider the nondeterministic path of the reduction that for all atoms correctly guesses whether $t_{2i} = t_{2i+1} \in \{\mathbb{N}\}$ or $t_{2i} = t_{2i+1} = \emptyset$, and that for all $t_{2i} = t_{2i+1} = \emptyset$ guesses subterms $u_1 \cap u_2$ in t_{2i} and $u_3 \cap u_4$ in t_{2i+1} such that $u_1, u_2, u_3, u_4 \in \{\mathbb{N}\}$, $u_1 \neq u_2$, and $u_3 \neq u_4$. If $t_{2i} = t_{2i+1} \in \{\mathbb{N}\}$, then $t'_{2i} = t_{2i} = t_{2i+1} = t'_{2i+1}$ and hence the formula is still satisfied after replacing t_{2i} with t'_{2i} and t_{2i+1} with t'_{2i+1}. Moreover, the added constraints $s'_1 = s'_2$ are satisfied, since in t_{2i} and t_{2i+1} all subterms $s_1 \cap s_2$ must be nonempty. If $t_{2i} = t_{2i+1} = \emptyset$, then after removing the atom $t_{2i} = t_{2i+1}$ and after adding the constraints $u'_1 \neq u'_2$ and $u'_3 \neq u'_4$ the formula is still satisfied. So at the described nondeterministic path the reduction outputs a formula $\psi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, =, \neq)$.

Assume there is a nondeterministic path where the reduction outputs a formula $\psi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, =, \neq)$. Consider a satisfying assignment a for ψ, we claim that a satisfies φ. If this is not true, then φ must have an atom $t_{2i} = t_{2i+1}$ that is not satisfied by a.

Case 1: At the path that produced ψ we guessed that $t_{2i} = t_{2i+1} \in \{\mathbb{N}\}$. In this case we added the constraints $s'_1 = s'_2$, which ensure that $t_{2i} = t'_{2i}$ and $t_{2i+1} = t'_{2i+1}$. Hence under the assignment a it holds that $t_{2i} = t'_{2i} = t'_{2i+1} = t_{2i+1}$, which contradicts the assumption that $t_{2i} = t_{2i+1}$ is not satisfied by a.

Case 2: At the path that produced ψ we guessed that $t_{2i} = t_{2i+1} = \emptyset$. Here we added the constraints $u'_1 \neq u'_2$ and $u'_3 \neq u'_4$, which are satisfied by a. Hence under the assignment a we have $t_{2i} = t_{2i+1} = \emptyset$, which contradicts the assumption that $t_{2i} = t_{2i+1}$ is not satisfied by a.

It follows that a satisfies φ and hence $\varphi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +)$.

Corollary 2. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, +), \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cap, \times) \in \mathrm{NP}$.

Proof. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; +, =, \neq)$-instances and $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \times, =, \neq)$-instances are formulas of existential Presburger arithmetic and existential Skolem arithmetic, which are both decidable in NP [45, 46]. Now the statement follows from Proposition 10.

3.2. New circuit results

We now show that the decidability of Skolem Arithmetic [40] can be used to decide the satisfiability of arithmetic circuits without $+$. This solves the main open question of the paper [1] and at the same time implies the decidability of the corresponding CSPs. Note that we consider Skolem Arithmetic with constants, which are not included in the treatment of [40]. For an explicit extension of decidability to the case with constants, see [47].

Our construction is motivated by the following idea. Consider a $\{^-, \cup, \times\}$-circuit C with n unassigned input gates. For every gate g in C, we construct a formula $\phi_g(x_1, \ldots, x_n, z)$ that expresses the predicate

$$C(x_1, \ldots, x_n) \text{ produces at gate } g \text{ a set that contains } z.$$

The definition of ϕ_g is straightforward (g_1, g_2 denote g's predecessors): If g is the i-th unassigned input gate, then $\phi_g(x_1, \ldots, x_n, z) := (z = x_i)$. If g is an assigned input gate with label $l \in \mathbb{N}$, then $\phi_g(x_1, \ldots, x_n, z) := (z = l)$. If g is a complement gate, then $\phi_g(x_1, \ldots, x_n, z) := \neg\phi_{g_1}(x_1, \ldots, x_n, z)$. If g is a $\cup$-gate, then $\phi_g(x_1, \ldots, x_n, z) := \phi_{g_1}(x_1, \ldots, x_n, z) \vee \phi_{g_2}(x_1, \ldots, x_n, z)$. If g is a $\times$-gate, $\phi_g(x_1, \ldots, x_n, z) := \exists f_1, f_2(\phi_{g_1}(x_1, \ldots, x_n, f_1) \wedge \phi_{g_2}(x_1, \ldots, x_n, f_2) \wedge z = f_1 f_2)$. For the output gate g_C it holds that

$$(C, z) \in \mathrm{SC}_{\mathbb{N}}(^-, \cup, \times) \iff \exists x_1, \ldots, x_n \; \phi_{g_C}(x_1, \ldots, x_n, z).$$

The right-hand side is a first-order sentence of Skolem Arithmetic.

By the above construction, each $\cup$-gate and each $\times$-gate double the size of the formula, which results in a formula ϕ_{g_C} of exponential size. Therefore, in the proof of Theorem 1 we introduce further variables to ϕ_g, which allow us to reuse the formula. For example, if g is a $\times$-gate, we can reuse the formula as follows.

$$\begin{aligned} \phi_g(x_1, \ldots, x_n, z) \quad := \quad & \exists f_1, f_2 \;\; \forall e \;\; \exists i, v \;\; [(f_1 \cdot f_2 = z) \wedge \\ & (e = 0 \rightarrow (i = g_1 \wedge v = f_1)) \wedge \\ & (e = 1 \rightarrow (i = g_2 \wedge v = f_2)) \wedge \\ & \phi_i(x_1, \ldots, x_n, v)] \end{aligned}$$

Here e acts like a switch: $e = 0$ expresses $\phi_{g_1}(x_1, \ldots, x_n, f_1)$ and $e = 1$ expresses $\phi_{g_2}(x_1, \ldots, x_n, f_2)$. This formula is technically more involved, but ϕ_i appears just once, which results in a formula ϕ_{g_C} of polynomial size.

Theorem 1. $\mathrm{SC}_{\mathbb{N}}(^{-}, \cup, \cap, \times) \in$ 3EXPSPACE.

PROOF. Let C be a circuit with gates $g_1, \ldots, g_r$ such that $g_1, \ldots, g_n$ are the unassigned input gates, $g_{n+1}, \ldots, g_m$ are the assigned input gates, and g_r is the output gate. We will reduce C to a first-order sentence ϕ such that $g_r \in I(C(g_1, \ldots, g_n))$ iff $(\mathbb{N}; \times, 0, 1, \ldots) \models \phi$. Without loss of generality we may assume that C does not have $\cap$-gates (recall that $A \cap B = {}^{-}({}^{-}A \cup {}^{-}B)$). For every gate g_k we define a formula $\varphi_k := \varphi_k(x_1, \ldots, x_n, i_k, v_k, b_k)$ in Skolem arithmetic such that the following holds.

$(*)$ For $a_1, \ldots, a_n, v \in \mathbb{N}$, $b \in \{0, 1\}$, and $i = 1, \ldots, k$ it holds that $\varphi_k(a_1, \ldots, a_n, 0, v, b)$ is true and

- $\varphi_k(a_1, \ldots, a_n, i, v, b)$ is true IFF
- ($b = 1$ iff $C(a_1, \ldots, a_n)$ produces at g_i a set that contains v).

Let $\varphi_0 := b_0 \vee \neg b_0 \vee (x_1 \cdot \ldots \cdot x_n \cdot i_0 \cdot v_0 = 0)$, which is always true and which has the free variables $x_1, \ldots, x_n, i_0, v_0, b_0$. For $1 \leq k \leq n$, the formula φ_k which corresponds to the k-th unassigned input gate g_k is defined as

$$\begin{aligned}\varphi_k := {} & \exists i_{k-1}, v_{k-1}, b_{k-1} \\ & [(i_k = k \wedge b_k = 0) \rightarrow (x_k \neq v_k \wedge i_{k-1} = 0)] \wedge \\ & [(i_k = k \wedge b_k = 1) \rightarrow (x_k = v_k \wedge i_{k-1} = 0)] \wedge \\ & [i_k \neq k \rightarrow (i_{k-1} = i_k \wedge v_{k-1} = v_k \wedge b_{k-1} = b_k)] \wedge \\ & \varphi_{k-1}.\end{aligned}$$

Observe that the free variables of φ_k are the variables $x_1, \ldots, x_n, i_k, v_k, b_k$, i.e., $\varphi_k = \varphi_k(x_1, \ldots, x_n, i_k, v_k, b_k)$. For $n + 1 \leq k \leq m$, the formula φ_k which corresponds to the assigned input gate g_k is defined analogously, just by replacing x_k with the label of the gate g_k. An induction on k shows that $(*)$ holds for all φ_k where $0 \leq k \leq m$.

Now define the formulas φ_k for the inner gates g_k where $m < k \leq r$. Here d_k, e_k, f_k, f'_k, h_k, and h'_k are used as auxiliary variables.

If g_k is a complement gate with predecessor g_p, then let

$$\begin{aligned}\varphi_k := {} & \exists i_{k-1}, v_{k-1}, b_{k-1} \\ & [i_k = k \rightarrow (i_{k-1} = p \wedge v_{k-1} = v_k \wedge (b_k = 1 \rightarrow b_{k-1} = 0) \wedge (b_k = 0 \rightarrow b_{k-1} = 1))] \wedge \\ & [i_k \neq k \rightarrow (i_{k-1} = i_k \wedge v_{k-1} = v_k \wedge b_{k-1} = b_k)] \wedge \\ & \varphi_{k-1}.\end{aligned}$$

If g_k is a $\cup$-gate with predecessors g_p and g_q, then let

$$\begin{aligned}\varphi_k := \exists f_k, h_k \forall e_k \exists i_{k-1}, v_{k-1}, b_{k-1}\\ &[(i_k = k \wedge e_k = 0) \rightarrow (i_{k-1} = p \wedge v_{k-1} = v_k \wedge b_{k-1} = f_k)] \wedge\\ &[(i_k = k \wedge e_k \neq 0) \rightarrow (i_{k-1} = q \wedge v_{k-1} = v_k \wedge b_{k-1} = h_k)] \wedge\\ &[(i_k = k \wedge b_k = 1) \rightarrow (f_k = 1 \vee h_k = 1)] \wedge\\ &[(i_k = k \wedge b_k = 0) \rightarrow (f_k = 0 \wedge h_k = 0)] \wedge\\ &[i_k \neq k \rightarrow (i_{k-1} = i_k \wedge v_{k-1} = v_k \wedge b_{k-1} = b_k)] \wedge\\ &\varphi_{k-1}.\end{aligned}$$

If g_k is a $\times$-gate with predecessors g_p and g_q, then let

$$\begin{aligned}\varphi_k := \exists f_k, f'_k \forall e_k \forall h_k, h'_k \exists d_k \exists i_{k-1}, v_{k-1}, b_{k-1}\\ &[(i_k = k \wedge b_k = 1 \wedge e_k = 0) \rightarrow (f_k \cdot f'_k = v_k \wedge i_{k-1} = p \wedge v_{k-1} = f_k \wedge b_{k-1} = 1)] \wedge\\ &[(i_k = k \wedge b_k = 1 \wedge e_k \neq 0) \rightarrow (f_k \cdot f'_k = v_k \wedge i_{k-1} = q \wedge v_{k-1} = f'_k \wedge b_{k-1} = 1)] \wedge\\ &[(i_k = k \wedge b_k = 0 \wedge h_k \cdot h'_k = v_k \wedge d_k = 0) \rightarrow (i_{k-1} = p \wedge v_{k-1} = h_k \wedge b_{k-1} = 0)] \wedge\\ &[(i_k = k \wedge b_k = 0 \wedge h_k \cdot h'_k = v_k \wedge d_k \neq 0) \rightarrow (i_{k-1} = q \wedge v_{k-1} = h'_k \wedge b_{k-1} = 0)] \wedge\\ &[i_k \neq k \rightarrow (i_{k-1} = i_k \wedge v_{k-1} = v_k \wedge b_{k-1} = b_k)] \wedge\\ &\varphi_{k-1}.\end{aligned}$$

Again it holds that φ_k's free variables are $x_1, \dots, x_n, i_k, v_k, b_k$ and an induction on k shows that $(*)$ holds for all φ_k where $0 \leq k \leq r$. So for the output gate g_r we obtain

$$(C, v) \in \mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, \times)) \iff \exists a_1, \dots, a_n\ \varphi_r(a_1, \dots, a_n, r, v, 1).$$

The right-hand side is a first-order sentence of Skolem arithmetic. On input (C, v) this sentence can be computed in polynomial time, which shows that $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, \times)$ is $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducible to Skolem arithmetic. The latter is decidable in 3EXPSPACE [40] (see Corollary 2.6 on page 137).

Historical note. The reader may be curious as to why decidability was previously known for $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, +)$ yet not for $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, \times)$. The authors of [1] had initially approached the circuit satisfiability problems with machinery developed for circuit memberships problems. Following this approach, the result for $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, +)$ is readily found, but that for $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, \times)$ is more elusive. Had a logic-oriented approach been initiated to $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, +)$, based on Presburger Arithmetic, then the approach to $\mathrm{SC}_{\mathbb{N}}(^-, \cup, \cap, \times)$ based on Skolem Arithmetic would have been more obvious.

Corollary 3. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, \times) \in$ 3EXPSPACE

Proof. By Theorem 1, it suffices to show that $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, \times) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{SC}_{\mathbb{N}}({}^-, \cup, \cap, \times)$. We describe the reduction on the input of an instance of $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, \times)$ given by $\phi := \exists y \in \mathbb{N}^n \bigwedge_{i=0}^{m}(t_{2i} = t_{2i+1})$. Observe that

$$\begin{aligned}
\bigwedge_{i=0}^{m}(t_{2i} = t_{2i+1}) &\iff \bigwedge_{i=0}^{m}(t_{2i} \cap \overline{t_{2i+1}}) \cup (\overline{t_{2i}} \cap t_{2i+1}) = \emptyset \\
&\iff \bigcup_{i=0}^{m}[(t_{2i} \cap \overline{t_{2i+1}}) \cup (\overline{t_{2i}} \cap t_{2i+1})] \ = \ \emptyset \\
&\iff 0 \ \in \ \underbrace{\overline{0 \times \bigcup_{i=0}^{m}[(t_{2i} \cap \overline{t_{2i+1}}) \cup (\overline{t_{2i}} \cap t_{2i+1})]}}_{C:=}.
\end{aligned}$$

So $\phi \in \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, \times)$ if and only if $(C, 0) \in \mathrm{SC}_{\mathbb{N}}({}^-, \cup, \cap, \times)$.

Corollary 4. $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, +) \in$ 3EXPSPACE

Proof. By Corollary 3, it suffices to show that we have $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, +) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, \times)$. Consider a $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, +)$-instance

$$\phi := \exists y \in \mathbb{N}^n \bigwedge_{i=0}^{m}(t_{2i} = t_{2i+1}).$$

We may assume that 0 and 1 are the only constants that occur in ϕ. We can do this, since constants $c > 1$ can be removed as follows: Let $l = \lfloor \log c \rfloor$, replace c with a new variable z, and add constraints

$$(z_0 = \{2\}) \wedge (z_1 = z_0 + z_0) \wedge \cdots \wedge (z_l = z_{l-1} + z_{l-1}) \wedge (z = \sum_{i \in I} z_i),$$

where $z_0, \ldots, z_l$ are new variables and

$$I = \{i \mid \text{the } i\text{-th bit in } c\text{'s binary representation is } 1\}.$$

Note that removing constants in this way can be done in polynomial time.

Consider the term $q := \overline{\left(\overline{\overline{\{0,1\}} \times \overline{\{0,1\}}} \cap \overline{\{0,1,2\}}\right) \times \overline{\{0\} \cap \{1\}}}$. Since $\overline{\{0,1\}} \times \overline{\{0,1\}}$ indicates all composite positive integers, its complement is the set of primes. Then $\left(\overline{\overline{\{0,1\}} \times \overline{\{0,1\}}} \cap \overline{\{0,1,2\}}\right)$ is the set of odd primes, and $\left(\overline{\overline{\{0,1\}} \times \overline{\{0,1\}}} \cap \overline{\{0,1,2\}}\right) \times \overline{\{0\} \cap \{1\}}$ is the set of positive integers involving an odd prime factor. Finally, we see the whole expression generates the set $\{2^i \mid i \in \mathbb{N}\}$.

For every term t, let t' be the term that is obtained from t if every constant c is replaced with 2^c, every $+$ operation is replaced with $\times$, and every complement operation $\overline{s}$ is replaced with $(\overline{s} \cap q)$. The computation of t' is possible in polynomial time, since only the constants 0 and 1 can appear.

The reduction outputs the $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; {}^-, \cup, \cap, \times)$-instance

$$\phi' := \exists y \in \mathbb{N}^n \bigwedge_{i=0}^{m} (t'_{2i} = t'_{2i+1}) \wedge \bigwedge_{i=1}^{n} (y_i \cup q = q).$$

Observe that for all terms t and all $e = (e_1, \ldots, e_n) \in \mathbb{N}^n$ it holds that

$$t'(2^{e_1}, \ldots, 2^{e_n}) = \{2^i \mid i \in t(e_1, \ldots, e_n)\}. \tag{1}$$

It remains to show that ϕ and ϕ' are equivalent.

If $e = (e_1, \ldots, e_n) \in \mathbb{N}^n$ is a satisfying assignment for ϕ, then by equation (1), $z = (2^{e_1}, \ldots, 2^{e_n})$ is a satisfying assignment for ϕ' (note that $\bigwedge_{i=1}^{n} (y_i \cup q = q)$ holds, since $y_i = 2^{e_i} \in q$).

If $z = (z_1, \ldots, z_n) \in \mathbb{N}^n$ is a satisfying assignment for ϕ', then because of the constraints $\bigwedge_{i=1}^{n} (y_i \cup q = q)$, $z_1 = 2^{e_1}, \ldots, z_n = 2^{e_n}$ for $e = (e_1, \ldots, e_n) \in \mathbb{N}^n$ and by (1), e is a satisfying assignment for ϕ.

A second open problem from [1]: $\mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times) \in \mathrm{PSPACE}$.

We now improve another of the upper bounds of [1] 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.

The absolute value of an integer x is denoted by $\mathrm{abs}(x)$. For $v = (v_1, \ldots, v_n) \in \mathbb{Z}^n$ and $A = (a_{i,j}) \in \mathbb{Z}^{m \times n}$ let $|v|_\infty = \max\{\mathrm{abs}(v_1), \ldots, \mathrm{abs}(v_n)\}$ and $|A|_\infty = \max\{\mathrm{abs}(a_{i,j}) \mid 1 \leq i \leq m \text{ and } 1 \leq j \leq n\}$. For a circuit C, let $|C|$ be the size of the circuit, i.e., the length of the encoding of the circuit.

An estimation from [48] yields the following bound for the size of small elements in $\mathbb{N}^n$ that solve a system of linear equations.

Lemma 1. *Let $k, m, n \in \mathbb{N}^+$, $A = (a_{i,j}) \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$ such that $|A|_\infty, |b|_\infty \leq k$. If there exists $y \in \mathbb{N}^n$ such that $Ay = b$, then there exists $z \in \mathbb{N}^n$ such that $Az = b$ and $|z|_\infty \leq (32k)^{12n^4}$.*

Proof. We adopt the following definitions from [48]. The size of a rational number $r = p/q$ where p and q are relatively prime integers is $\mathrm{size}(r) = 1 + \lceil \log_2(\mathrm{abs}(p) + 1)\rceil + \lceil \log_2(\mathrm{abs}(q) + 1)\rceil$. The size of a rational vector $v = (v_1, \dots, v_n) \in \mathbb{Q}^n$ is $\mathrm{size}(v) = n + \mathrm{size}(v_1) + \cdots + \mathrm{size}(v_n)$. The size of a rational matrix $A = (a_{i,j}) \in \mathbb{Q}^{m\times n}$ is $\mathrm{size}(A) = mn + \sum_{i,j} \mathrm{size}(a_{i,j})$. The size of a system $Ax \le b$ of rational linear inequalities is $\mathrm{size}(Ax \le b) = 1 + \mathrm{size}(A) + \mathrm{size}(b)$. A rational polyhedron is a set $\{x \in \mathbb{R}^n \mid Ax \le b\}$ for some $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$. The facet complexity of a rational polyhedron $P \subseteq \mathbb{R}^n$ is the smallest number φ such that $\varphi \ge n$ and there exists a system $Ax \le b$ of rational linear inequalities defining P, where each inequality in $Ax \le b$ has size at most φ, i.e., there exist $m \in \mathbb{N}$, $A = (a_{i,j}) \in \mathbb{Q}^{m\times n}$, and $b = (b_1, \dots, b_m) \in \mathbb{Q}^m$ such that $P = \{x \in \mathbb{R}^n \mid Ax \le b\}$ and $\forall i\, [1 + n + \mathrm{size}(b_i) + \sum_{j=1}^{n} \mathrm{size}(a_{i,j}) \le \varphi]$.

Let $C = \begin{pmatrix} A \\ -A \\ -I_n \end{pmatrix} \in \mathbb{Z}^{(2m+n)\times n}$ and $d = \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix} \in \mathbb{Z}^{2m+n}$, where I_n denotes the identity matrix of size n and 0 the zero element in $\mathbb{Z}^n$. Consider the rational polyhedron $P = \{x \in \mathbb{R}^n \mid Cx \le d\}$ and let φ be its facet complexity. Observe that $\varphi \le 1 + n + \mathrm{size}(k) + n \cdot \mathrm{size}(k) \le (n + 1) \cdot \log_2 16(k + 1)$. By definition, $Cx \le d$ if and only if $Ax \le b$ and $-Ax \le -b$ and $-I_n x \le 0$ if and only if $Ax = b$ and $x \in (\mathbb{R}^{\ge 0})^n$. Therefore, $P = \{x \in (\mathbb{R}^{\ge 0})^n \mid Ax = b\}$. By assumption, $y \in P \cap \mathbb{Z}^n$ and hence $P \cap \mathbb{Z}^n \ne \emptyset$. By Corollary 17.1b in [48], there exists $z \in P \cap \mathbb{Z}^n = P \cap \mathbb{N}^n$ such that $\mathrm{size}(z) \le 6n^3\varphi$. So $\mathrm{size}(z) \le 6n^3(n + 1) \cdot \log_2 16(k + 1) \le 12n^4 \cdot \log_2 32k$ and hence $|z|_\infty \le (32k)^{12n^4}$.

We use the following problem which asks for the solvability of a *system of monomial equations*.

Name: MonEq
Instance: A list of equations of the following form.

$$\begin{aligned} x^5 z^7 &= 5^9 y^3 z^2 \\ y z^2 &= 2^3 x^5 \\ x^2 y^4 z^3 &= 3^{11} \end{aligned}$$

Question: Is this system of equations solvable over the natural numbers?

Formally, the problem MonEq is defined as follows, where 0^0 is defined as 1.

MonEq $= \{(A, B, C, D) \mid A = (a_{i,j}) \in \mathbb{N}^{m\times n},\ B = (b_{i,j}) \in \mathbb{N}^{m\times n},\ C = (c_1, \dots, c_m) \in \mathbb{N}^m,\ D = (d_1, \dots, d_m) \in \mathbb{N}^m$, and there exist $x_1, \dots, x_n \in \mathbb{N}$ such that for all $i \in \{1, \dots, m\}$, $\prod_{j=1}^n x_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n x_j^{b_{i,j}}\}$

Let us consider the variant of MonEq that restricts to positive constant factors and positive solutions. We show that if a system of such monomial equations has a solution over $\mathbb{N}^+$, then it has a solution that consists of small numbers in $\mathbb{N}^+$.

Lemma 2. *Let $k, m, n \in \mathbb{N}^+$, $A = (a_{i,j}) \in \mathbb{N}^{m\times n}$, $B = (b_{i,j}) \in \mathbb{N}^{m\times n}$, $C = (c_1, \dots, c_m) \in (\mathbb{N}^+)^m$, and $D = (d_1, \dots, d_m) \in \mathbb{N}^m$ such that $|A|_\infty, |B|_\infty, |C|_\infty, |D|_\infty \le k$ and $|\{p \mid p \text{ is prime factor of } c_1 \cdots c_m\}| \le k$. If there exists $y = (y_1, \dots, y_n) \in (\mathbb{N}^+)^n$ such that $\bigwedge_{i=1}^m (\prod_{j=1}^n y_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n y_j^{b_{i,j}})$, then there exists $z = (z_1, \dots, z_n) \in (\mathbb{N}^+)^n$ such that $\bigwedge_{i=1}^m (\prod_{j=1}^n z_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n z_j^{b_{i,j}})$ and $|z|_\infty \le 2^{(32k^2)^{13n^4}}$.*

Proof. Let $\{p_1, \dots, p_l\}$ be the set of prime factors of $c_1 \cdots c_m$. So $c_i = p_1^{e_{i,1}} p_2^{e_{i,2}} \cdots p_l^{e_{i,l}}$ for suitable exponents $e_{i,j} \in \mathbb{N}$. By assumption, there exist $y_1, \dots, y_n \in \mathbb{N}^+$ such that

$$\bigwedge_{i=1}^m (\prod_{j=1}^n y_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n y_j^{b_{i,j}}). \tag{2}$$

We may assume that $y_i = p_1^{s_{i,1}} p_2^{s_{i,2}} \cdots p_l^{s_{i,l}}$ for suitable exponents $s_{i,j} \in \mathbb{N}$, since other prime factors can be deleted in y_i. The equalities in (2) hold with respect to each prime factor, i.e., for all $r \in \{1, \dots, l\}$,

$$\bigwedge_{i=1}^m \ \prod_{j=1}^n (p_r^{s_{j,r}})^{a_{i,j}} = (p_r^{e_{i,r}})^{d_i} \cdot \prod_{j=1}^n (p_r^{s_{j,r}})^{b_{i,j}}. \tag{3}$$

So for fixed $r \in \{1, \dots, l\}$ it holds that $\bigwedge_{i=1}^m \sum_{j=1}^n (a_{i,j} - b_{i,j}) s_{j,r} = d_i e_{i,r}$. Hence $A' s_r = t_r$, where $A' = A - B \in \mathbb{Z}^{m\times n}$, $s_r = (s_{1,r}, \dots, s_{n,r}) \in \mathbb{N}^n$, and $t_r = (d_1 e_{1,r}, \dots, d_m e_{m,r}) \in \mathbb{N}^m$. Note that $|A'|_\infty \le k$ and $|t_r|_\infty \le k^2$, since $e_{i,r} \le p_r^{e_{i,r}} \le c_i \le k$. By Lemma 1, there exists $x_r = (x_{1,r}, \dots x_{n,r}) \in \mathbb{N}^n$ such that $A'x_r = t_r$ and $|x_r|_\infty \leq (32k^2)^{12n^4}$. So $\bigwedge_{i=1}^m \sum_{j=1}^n (a_{i,j} - b_{i,j})x_{j,r} = d_i e_{i,r}$ and hence

$$\bigwedge_{i=1}^{m} \prod_{j=1}^{n} (p_r^{x_{j,r}})^{a_{i,j}} = (p_r^{e_{i,r}})^{d_i} \cdot \prod_{j=1}^{n} (p_r^{x_{j,r}})^{b_{i,j}}. \tag{4}$$

Since this equation holds for all primes p_r, we obtain

$$\bigwedge_{i=1}^{m} \prod_{j=1}^{n} (p_1^{x_{j,1}} \cdots p_l^{x_{j,l}})^{a_{i,j}} = (p_1^{e_{i,1}} \cdots p_l^{e_{i,l}})^{d_i} \cdot \prod_{j=1}^{n} (p_1^{x_{j,1}} \cdots p_l^{x_{j,l}})^{b_{i,j}}. \tag{5}$$

We define $z_j = p_1^{x_{j,1}} \cdots p_l^{x_{j,l}} \in \mathbb{N}^+$ and obtain

$$\bigwedge_{i=1}^{m} \prod_{j=1}^{n} z_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^{n} z_j^{b_{i,j}}. \tag{6}$$

By assumption, $l, |C|_\infty \leq k$ and hence $p_1, \ldots, p_l \leq k$. So $z_j \leq k^{x_{j,1}+\cdots+x_{j,l}} \leq k^{k\cdot(32k^2)^{12n^4}} \leq 2^{(32k^2)^{13n^4}}$.

We extend Lemma 2 to the case where the vectors C, y, and z can contain 0's.

Corollary 5. *Let $k, m, n \in \mathbb{N}^+$, $A = (a_{i,j}) \in \mathbb{N}^{m\times n}$, $B = (b_{i,j}) \in \mathbb{N}^{m\times n}$, $C = (c_1, \ldots, c_m) \in \mathbb{N}^m$, and $D = (d_1, \ldots, d_m) \in \mathbb{N}^m$ such that $|A|_\infty, |B|_\infty, |C|_\infty, |D|_\infty \leq k$ and $|\{p \mid p \text{ is prime factor of } c_1 \cdots c_m\}| \leq k$. If there exists $y = (y_1, \ldots, y_n) \in \mathbb{N}^n$ such that $\bigwedge_{i=1}^m (\prod_{j=1}^n y_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n y_j^{b_{i,j}})$, then there exists $z = (z_1, \ldots, z_n) \in \mathbb{N}^n$ such that $\bigwedge_{i=1}^m (\prod_{j=1}^n z_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n z_j^{b_{i,j}})$ and $|z|_\infty \leq 2^{(32k^2)^{13n^4}}$.*

Proof. We may assume that ($c_i = 0 \implies d_i \neq 0$) for all $i \in \{1, \ldots, m\}$, since otherwise $c_i^{d_i} = 1 = 1^{d_i}$ and we can use $c_i = 1$ instead of $c_i = 0$. Consider the equations $\prod_{j=1}^n y_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^n y_j^{b_{i,j}}$ for $i \in \{1, \ldots, m\}$. If such an equation is 0, then we delete it, i.e., we delete the i-th row in A and B, and the i-th component of C and D. This results in matrices $A' = (a'_{i,j}), B' = (b'_{i,j}) \in \mathbb{N}^{m'\times n}$ and vectors $C' = (c'_i), D' = (d'_i) \in \mathbb{N}^{m'}$. Note that $C' \in (\mathbb{N}^+)^{m'}$, since if $c'_i = 0$, then by our assumption $d'_i \neq 0$ and hence equation i is 0 and was deleted. Let $y' = (y'_1, \ldots, y'_n) \in (\mathbb{N}^+)^n$ be the vector that is obtained from y by replacing all 0's with 1's. Observe that

$$\bigwedge_{i=1}^{m'} (\prod_{j=1}^{n} {y'}_j^{a'_{i,j}} = {c'}_i^{d'_i} \cdot \prod_{j=1}^{n} {y'}_j^{b'_{i,j}}), \tag{7}$$

since if $y_j = 0$, then $\bigwedge_{i=1}^{m'} a'_{i,j} = b'_{i,j} = 0$ (otherwise equation i is 0 and was deleted) and hence $y_j^{a'_{i,j}} = y_j^{b'_{i,j}} = 1 = {y'}_j^{a'_{i,j}} = {y'}_j^{b'_{i,j}}$. By Lemma 2, there exists $z' = (z'_1, \dots, z'_n) \in (\mathbb{N}^+)^n$ such that

$$\bigwedge_{i=1}^{m'} (\prod_{j=1}^{n} {z'}_j^{a'_{i,j}} = {c'}_i^{d'_i} \cdot \prod_{j=1}^{n} {z'}_j^{b'_{i,j}}) \tag{8}$$

and $|z'|_\infty \leq 2^{(32k^2)^{13n^4}}$. Let $J = \{j \mid 1 \leq j \leq n \text{ and } y_j = 0\}$ be the set of positions of 0's in y. Let $z = (z_1, \dots, z_n) \in \mathbb{N}^n$ be the vector that is obtained from z' by replacing the j-th component with 0 for all $j \in J$. It follows that

$$\bigwedge_{i=1}^{m'} (\prod_{j=1}^{n} z_j^{a'_{i,j}} = {c'}_i^{d'_i} \cdot \prod_{j=1}^{n} z_j^{b'_{i,j}}), \tag{9}$$

since for $j \in J$ it holds that $y_j = 0$ and hence $\bigwedge_{i=1}^{m'} a'_{i,j} = b'_{i,j} = 0$ and ${z'}_j^{a'_{i,j}} = {z'}_j^{b'_{i,j}} = 1 = z_j^{a'_{i,j}} = z_j^{b'_{i,j}}$.

It remains to argue that z also satisfies each deleted equation i. Assume $\prod_{j=1}^{n} y_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^{n} y_j^{b_{i,j}} = 0$. From $\prod_{j=1}^{n} y_j^{a_{i,j}} = 0$ it follows that there exists $j \in \{1, \dots, n\}$ such that $y_j^{a_{i,j}} = 0$ and hence $y_j = 0 = z_j$ and $a_{i,j} \neq 0$, which shows $z_j^{a_{i,j}} = 0$ and $\prod_{j=1}^{n} z_j^{a_{i,j}} = 0$. From $c_i^{d_i} \cdot \prod_{j=1}^{n} y_j^{b_{i,j}} = 0$ it follows that $c_i^{d_i} = 0$ or there exists $j \in \{1, \dots, n\}$ such that $y_j^{b_{i,j}} = 0$. As above this implies $c_i^{d_i} \cdot \prod_{j=1}^{n} z_j^{b_{i,j}} = 0$. This shows that z also satisfies all deleted equations. Together with (9) we obtain

$$\bigwedge_{i=1}^{m} (\prod_{j=1}^{n} z_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^{n} z_j^{b_{i,j}}). \tag{10}$$

Finally, note that $|z|_\infty \leq |z'|_\infty \leq 2^{(32k^2)^{13n^4}}$.

Lemma 3. *Let C_0 be a $\{\cup, \cap, \times\}$-circuit with n unassigned input gates. If $(C_0, 0) \in \mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times)$, then there exists $z = (z_1, \dots, z_n) \in \mathbb{N}^n$ such that $0 \in C_0(z_1, \dots, z_n)$ and $|z|_\infty \leq 2^{2^{91 \cdot |C_0|^5}}$.*

Proof. Let $1, \dots, n$ be the unassigned input gates. Moreover, let $n+1, \dots, n+r$ be the assigned input gates and let $b_{n+1}, \dots, b_{n+r} \in \mathbb{N}$ be their labels. We may assume that in C_0 all gates are connected to the output gate. Assume $0 \in C_0(y_1, \ldots, y_n)$ for suitable $y_1, \ldots, y_n \in \mathbb{N}$.

Let C_1 be the $\{\cup, \cap, \times\}$-circuit that is obtained by recursively unfolding C_0 such that all inner gates have outdegree 1 (i.e., if one disregards the input gates, then C_1 is a tree). More precisely, C_1 is obtained from C_0 by duplicating all inner gates with outdegree greater than 1, where the gates are processed recursively from bottom to top. Observe that

$$\forall x_1, \ldots, x_n \in \mathbb{N}, C_1(x_1, \ldots, x_n) = C_0(x_1, \ldots, x_n) \text{ and} \tag{11}$$
$$|C_1| \leq 2^{|C_0|} - 1, \tag{12}$$

where the inequality is shown by an induction on the number of gates. Now for each union gate in C_1, we will cut either the left or the right input (hence making union gates trivial) and then delete all inner gates that are not connected to the output gate anymore. Observe that in this way it is possible to obtain a $\{\cap, \times\}$-circuit C_2 such that

$$C_2(y_1, \ldots, y_n) = \{0\}. \tag{13}$$

Moreover, it holds that

$$\forall x_1, \ldots, x_n \in \mathbb{N}, C_2(x_1, \ldots, x_n) \subseteq C_0(x_1, \ldots, x_n) \text{ and} \tag{14}$$
$$|C_2| \leq 2^{|C_0|}. \tag{15}$$

Let $I(g)$ be the set computed by gate g of the circuit $C_2(y_1, \ldots, y_n)$. From $C_2(y_1, \ldots, y_n) \neq \emptyset$ it follows that $|I(g)| = 1$ for all gates g.

We recursively attach a monomial of the form $x_1^7 x_2^{23} \cdots x_{n+r}^5$ to each gate of C_2: For $g \in \{1, \ldots, n+r\}$ attach the monomial x_i to the input gate g. Let g be a gate with the direct predecessors g_1 and g_2 such that the monomial M_1 is attached to g_1 and M_2 is attached to g_2. If g is a $\times$-gate, then attach the monomial $M_1 \cdot M_2$ to g (where the product is simplified such that multiple occurrences of the same variable are combined). If g is a $\cap$-gate, then attach the monomial M_1 to g. Apart from the input gates, C_2 is a tree and each input gate has outdegree $\leq |C_2| \leq 2^{|C_0|}$. Therefore, the exponents in the monomials are less than or equal to $2^{|C_0|}$.

Consider the following system of monomial equations: For each $\cap$-gate g we take the equation $M_1 = M_2$ to the system, where M_1 and M_2 are the monomials that are attached to g's direct predecessors. For each assigned input gate $g \in \{n+1, \ldots, n+r\}$ we take the equation $x_g = b_g$ to the system.

Finally, we take the equation $M = 0$ to the system, where M is the monomial attached to the output gate.

For a monomial M attached to some gate, let $M(a_1, \ldots, a_n)$ denote the number that is obtained when M is evaluated for $x_1 = a_1, \ldots, x_n = a_n$ and $x_{n+1} = b_{n+1}, \ldots, x_{n+r} = b_{n+r}$. An induction on the structure of C_2 yields the following.

Claim 1. *If gate g in C_2 has the monomial M attached, then for all $a_1, \ldots, a_n \in \mathbb{N}$, the gate g of the circuit $C_2(a_1, \ldots, a_n)$ either computes $\emptyset$ or computes the set $\{M(a_1, \ldots, a_n)\}$.*

Next we show that the solutions of our system of monomial equations are exactly the assignments that produce $\{0\}$ at the output gate.

Claim 2. *For all $a_1, \ldots, a_{n+r} \in \mathbb{N}$ it holds that $(a_1, \ldots, a_{n+r})$ is a solution for our system of monomial equations if and only if $C_2(a_1, \ldots, a_n) = \{0\}$ and $\bigwedge_{g=n+1}^{n+r} a_g = b_g$.*

Assume $(a_1, \ldots, a_{n+r})$ is a solution for our system of monomial equations. Because of the equations $x_g = b_g$ it must hold that $\bigwedge_{g=n+1}^{n+r} a_g = b_g$. Let us show that $I(C_2(a_1, \ldots, a_n)) \neq \emptyset$: Otherwise there exists a $\cap$-gate g with direct predecessors g_1 and g_2 such that $I(g_1) \neq \emptyset$, $I(g_2) \neq \emptyset$, and $I(g_1) \neq I(g_2)$. Let M, M_1, and M_2 be the monomials attached to g, g_1, and g_2, respectively. By Claim 1, $I(g_1) = \{M_1(a_1, \ldots, a_n)\}$ and $I(g_2) = \{M_2(a_1, \ldots, a_n)\}$. The equation $M_1 = M_2$ appears in our system of monomial equations and hence for the assignment $x_1 = a_1, \ldots, x_{n+r} = a_{n+r}$, the value of M_1 equals the value of M_2. Together with $\bigwedge_{g=n+1}^{n+r} a_g = b_g$ this shows $M_1(a_1, \ldots, a_n) = M_2(a_1, \ldots, a_n)$ and hence $I(g_1) = I(g_2)$, which is a contradiction. This shows $I(C_2(a_1, \ldots, a_n)) \neq \emptyset$. Now let M denote the monomial attached to the output gate and recall that the equation $M = 0$ appears in our system of monomial equations. Together with $\bigwedge_{g=n+1}^{n+r} a_g = b_g$ this implies $M(a_1, \ldots, a_n) = 0$. By Claim 1, $I(C_2(a_1, \ldots, a_n)) = \{M(a_1, \ldots, a_n)\} = \{0\}$.

Conversely, assume $I(C_2(a_1, \ldots, a_n)) = \{0\}$ and $\bigwedge_{g=n+1}^{n+r} a_g = b_g$. We show that $x_1 = a_1$, …, $x_n = a_n$, $x_{n+1} = b_{n+1}$, …, $x_{n+r} = b_{n+r}$ is a solution for our system of monomial equations. In the circuit $C_2(a_1, \ldots, a_n)$, each $\cap$-gate g computes a nonempty set. So if g_1 and g_2 are the predecessors of g, then $I(g) = I(g_1) = I(g_2)$. Let M, M_1, and M_2 be the monomials attached to g, g_1, and g_2, respectively. From Claim 1 it follows that $M_1(a_1, \ldots, a_n) = M_2(a_1, \ldots, a_n)$. So all equations of the form $M_1 = M_2$ are satisfied. The additional equations of the form $x_{n+i} = b_{n+i}$ are also satisfied. From $I(C_2(a_1, \dots, a_n)) = \{0\}$ and Claim 1 it follows that $M(a_1, \dots, a_n) = 0$ where M is the monomial attached to C_2's output gate. This shows that all equations of our system are satisfied by the solution $(a_1, \dots, a_{n+r})$. This proves Claim 2.

Our system of monomial equations can be formally written as

$$\bigwedge_{i=1}^{m} \left(\prod_{j=1}^{n'} x_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^{n'} x_j^{b_{i,j}}\right), \tag{16}$$

where $m = 1 + r +$ (number of $\cap$-gates), $n' = n + r$, $a_{i,j}, b_{i,j}, c_i \in \mathbb{N}$, and $d_i = 1$. The factors $c_i^{d_i}$ are used to express the right-hand side of the equations $x_g = b_g$ and $M = 0$. Define $A = (a_{i,j}) \in \mathbb{N}^{m \times n'}$, $B = (b_{i,j}) \in \mathbb{N}^{m \times n'}$, $C = (c_1, \dots, c_m) \in \mathbb{N}^m$, $D = (d_1, \dots, d_m) \in \mathbb{N}^m$, and $k = 2^{|C_0|}$. Since the variables $a_{i,j}$ and $b_{i,j}$ describe exponents in monomials attached to gates in C_2, they are less than or equal to k. Moreover, $c_i \leq k$, since these variables are 0 or they describe labels of assigned input gates in C_2, which are also labels of assigned input gates in C_0. Hence $|A|_\infty, |B|_\infty, |C|_\infty, |D|_\infty \leq k$. Moreover, $|\{p \mid p \text{ is prime factor of } c_1 \cdots c_m\}| \leq k$, since $\{p \mid p \text{ is prime factor of } c_1 \cdots c_m\}$ is a subset of

$$\{p \mid p \text{ is prime factor of the label of an assigned input in } C_0\}.$$

By (13), $C_2(y_1, \dots, y_n) = \{0\}$ and from Claim 2 it follows that $y = (y_1, \dots, y_{n'}) := (y_1, \dots, y_n, b_{n+1}, \dots, b_{n+r})$ is a solution for our system of monomial equations, i.e., $\bigwedge_{i=1}^{m}(\prod_{j=1}^{n'} y_j^{a_{i,j}} = c_i^{d_i} \cdot \prod_{j=1}^{n'} y_j^{b_{i,j}})$. From Corollary 5 it follows that there exists a solution $z' = (z_1, \dots, z_{n'}) \in \mathbb{N}^{n'}$ such that $|z'|_\infty \leq 2^{(32k^2)^{13n'^4}}$. Let $z = (z_1, \dots, z_n)$ and observe that

$$|z|_\infty \leq |z'|_\infty \leq 2^{(32k^2)^{13n'^4}} \leq 2^{(32(2^{2|C_0|}))^{13|C_0|^4}} \leq 2^{2^{91 \cdot |C_0|^5}}. \tag{17}$$

By Claim 2, $C_2(z_1, \dots, z_n) = \{0\}$. By (14), $C_2(z_1, \dots, z_n) \subseteq C_0 = (z_1, \dots, z_n)$ and hence $0 \in C_0(z_1, \dots, z_n)$.

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

Proof. Let C be a $\{\cup, \cap, \times\}$-circuit with n unassigned input gates. It suffices to decide whether or not $(C, 0) \in \mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times)$, since for every $b \in$

$\mathbb{N}$ it holds that $(C, b) \in \mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times)$ if and only if $((C \cap \{b\}) \times 0, 0) \in \mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times)$. Assume $(C, 0) \in \mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times)$, i.e., there exist $y_1, \ldots, y_n \in \mathbb{N}$ such that $0 \in C(y_1, \ldots, y_n)$. By Lemma 3, there exists $z = (z_1, \ldots, z_n) \in \mathbb{N}^n$ such that $0 \in C(z_1, \ldots, z_n)$ and $|z|_\infty \leq 2^{2^{91 \cdot |C|^5}}$.

Let $P = \{p_1, \ldots, p_l\}$ be the set of prime factors of the labels of the assigned inputs. For $x \in \mathbb{N}^+$ let $\pi(x)$ be the number obtained from x, if one removes all prime factors that are not in P, and let $\pi(0) = 0$. An induction on the structure of C shows the following.

Claim 3. *Let $x_1, \ldots, x_n \in \mathbb{N}$, $I(g)$ the set computed by gate g of the circuit $C(x_1, \ldots, x_n)$, and $I'(g)$ the set computed by gate g of the circuit $C(\pi(x_1), \ldots, \pi(x_n))$. Then it holds that $\pi(I(g)) \subseteq I'(g)$.*

Let $z' = (z'_1, \ldots, z'_n)$ with $z'_i = \pi(z_i)$. By Claim 3, we have $0 \in C(z'_1, \ldots, z'_n)$ and $|z'|_\infty \leq |z|_\infty \leq 2^{2^{91 \cdot |C|^5}}$. Hence $z'_1, \ldots, z'_n \in S := \{0\} \cup \{p_1^{e_1} \cdots p_l^{e_l} \mid e_1, \ldots, e_l \leq 2^{91 \cdot |C|^5}\}$. So each positive z'_i can be described by a vector $(e_{i,1}, \ldots, e_{i,l})$, which has polynomial size in $|C|$. This yields the following nondeterministic algorithm, which decides whether $(C, 0) \in \mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times)$.

- determine the set $P = \{p_1, \ldots, p_l\}$ of prime factors of the labels of the assigned inputs
- nondeterministically choose $z'_1, \ldots, z'_n \in S$, where positive z'_i are represented by vectors $(e_{i,1}, \ldots, e_{i,l})$
- build the circuit $C' = C(z'_1, \ldots, z'_n)$, where each number $z'_1, \ldots, z'_n$ is generated by a small $\{\times\}$-circuit using square and multiply (note that these numbers can have exponential size)
- if $(C', 0) \in \mathrm{MC}_{\mathbb{N}}(\cup, \cap, \times)$, then accept, otherwise reject

It is known that $\mathrm{MC}_{\mathbb{N}}(\cup, \cap, \times) \in \mathrm{PSPACE}$ [25]. So our algorithm shows $\mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times) \in \mathrm{PSPACE}$.

It is not clear how to turn the upper bound $\mathrm{SC}_{\mathbb{N}}(\cup, \cap, \times) \in \mathrm{PSPACE}$ given in Theorem 2 into an upper bound for $\mathrm{CSP}^{\mathrm{c}}(\{\mathbb{N}\}; \cup, \cap, \times)$. The reason is that with CSPs we can formulate queries such as "there exists an assignment such that two expressions generate the same set". We do not know how to formulate this with circuits.

From Table 1 we can recover some interesting 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 first-order 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). By *syntactic*, we mean it belongs to the signature (language) of Skolem Arithmetic. When we wish to refer to multiplication in a purely semantic way, we prefer the dot notation $\cdot$s or $\prod$. We will consider $\times$ as a ternary relation rather than a binary function. This would only affect our complexity-theoretic results up to a quadratic factor, since the functional and relational notations are equivalent under $\times(x, y) = z$ (prefix functional) iff $\exists z \ \times(x, y, z)$ (prefix relational). We use the prefix variant here to avoid ambiguity, in our exposition we always favour infix. The technical advantage in using the relational form can be exemplified by our argument in the upcoming Lemma 4, in which we want to take the atomic constraints from the instance. In the functional formulation these may have nested applications of the function operator, and turning this to an equivalent relational formulation adds potentially quadratically many constraints. 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.

Proposition 11. *Let* Γ *be a finite signature reduct of* $(\mathbb{N}; \times, 1, 2, \ldots)$. *Then* $\mathrm{CSP}(\Gamma)$ *is in* NP.

Proof. It is known that Skolem Arithmetic admits quantifier-elimination and that the existential theory in this language is in NP [46]. The result follows when one considers that we can substitute quantifier-free definitions for each among our finite set of first-order definable relations.

Upper bounds. We continue with polynomial upper bounds. Note that constants are no longer assumed to necessarily exist in our structures (in contrast to the situation in Proposition 11).

Lemma 4. *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* $\mathrm{CSP}(\mathbb{N}^+; \times, U)$.

Proof. Let ϕ be an arbitrary instance of $\mathrm{CSP}(\mathbb{N}; \times, U)$ involving a set of atoms C on variables V. We construct a set $V' \subseteq V$ incrementally by repeating the following three steps until a fixed point is reached.

- if $U(v) \in C$, then $V' := V' \cup \{v\}$,
- if $(x \times y = z) \in C$ and $z \in V'$, then $V' := V' \cup \{x, y\}$, and
- if $(x \times y = z) \in C$ and $x, y \in V'$, then $V' := V' \cup \{z\}$.

Note that if $v \in V'$, then any solution to ϕ must satisfy $s(v) \neq 0$. Let $V_0 = V \setminus V'$. Construct ϕ', an instance for $\mathrm{CSP}(\mathbb{N}; \times, U, 0)$, with atoms C' and variables V' by replacing each variable $v \in V_0$ with the constant 0. Note the following:

1. if $U(0) \in C'$, then the variable that was replaced by 0 is a member of V' so this case cannot occur.
2. if $(0 \times y = z) \in C'$ or $(x \times 0 = z) \in C'$, then the variable replaced by 0 is not a member of V' while z is a member of V'. This situation cannot occur.
3. if $(x \times y = 0) \in C'$, then note that $x, y \in V'$ so the variable that was replaced with 0 also was a member of V'. Hence, this case cannot occur.

Thus, 0 can only appear in three cases: $(x \times 0 = 0)$, $(0 \times x = 0)$, and $(0 \times 0 = 0)$. Let ϕ'' be an instance for $\mathrm{CSP}(\mathbb{N}^+; \times, U)$ built from atoms C'' and variables V' where C'' is obtained from C' by removing these kinds of constraints. Note that $(\mathbb{N}; \times, U, 0) \models \phi'$ iff $(\mathbb{N}; \times, U) \models \phi''$. Also note that if ϕ'' has a solution, then it has a solution $s : V' \to \mathbb{N}^+$. This implies that ϕ' is satisfiable on $(\mathbb{N}; \times, U)$ iff it is satisfiable on $(\mathbb{N}^+; \times, U)$

The transformation above can obviously be carried out in polynomial time. In order to prove the lemma, it remains to show that $(\mathbb{N}; \times, U) \models \phi$ iff $(\mathbb{N}; \times, U, 0) \models \phi'$.

Assume first that ϕ has a solution $s : V \to \mathbb{N}$. Since $C'' \subseteq C$, it follows immediately that s is a solution to ϕ'', too.

Assume instead that ϕ'' has a solution $s' : V' \to \mathbb{N}^+$. We claim that the function $s : V \to \mathbb{N}$ defined by $s(v) = s'(v)$ for $v \in V'$ and $s(v) = 0$

otherwise is a solution to ϕ. If the variable $v \in V$ appears in an atom $U(v)$, then $v \in V'$ and $U(v)$ is satisfied by s. Consider an atom $(x \times y = z) \in C$. If $\{x, y, z\} \subseteq V'$, then s satisfies the atom since $(x \times y = z) \in C''$. Assume $x \notin V'$. Then $z \notin V'$, $s(x) = s(z) = 0$ and the atom is satisfied by s. The same reasoning applies when $y \notin V'$. Assume finally that $z \notin V'$. Then at least one of $x, y \notin V'$ so $s(x) = 0$ and/or $s(y) = 0$. Combining this with the fact that $s(z) = 0$ implies that the atom is satisfied by s.

We now borrow the following slight simplification of Lemma 6 from [49].

Lemma 5 (Scalability [49]). *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 6. *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*.*

Proof. For natural numbers x, define $\ell(x)$ to be the number of factors of m in x, i.e. the number of times one may successively divide x by m without leaving a remainder. Now define $h(x) = m^{\ell(x)}$. Let $D = \{1, m, m^2, m^3, \ldots\}$. The function h is a homomorphism from $(\mathbb{N}^+; \times, U)$ to $(D; \times, U \cap D)$. Clearly, $h(U) = U \cap D$. Suppose $a \cdot b = c$ where $a, b, c \in \mathbb{N}^+$. We see that

$$h(a) \cdot h(b) = m^{\ell(a)} \cdot m^{\ell(b)} = m^{\ell(a)+\ell(b)} = m^{\ell(a \cdot b)} = m^{\ell(c)} = h(c).$$

Define $h'(1) = 0$ and $h'(m^k) = k$. Note that h' is a homomorphism from $(D; \times, U \cap D)$ to $(\mathbb{N}; +, x \geq 1)$. We know that $\mathrm{CSP}(\mathbb{R}; +, x \geq 0, x \geq 1)$ is in P (via linear programming) and this implies tractability of $\mathrm{CSP}(\mathbb{N}; +, x \geq 1)$ through $\mathrm{CSP}(\mathbb{Z}; +, x \geq 1, x \geq 0)$ by Lemma 5.

Proposition 12. *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.

Proof. Combine Lemma 4 with Lemma 6.

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

Proof. Consider the function $e : \mathrm{Div}_m \to \mathrm{Div}_m$ uniquely defined by $e(1) = 1$, $e(p) = m$, $e(p_1) = \cdots = e(p_k) = 1$ (i.e. all the other prime divisors map to 1), and the rule $e(x \cdot x') = e(x) \cdot e(x')$. We claim that e is an endomorphism of $(\mathrm{Div}(m); \times, m)$. Clearly, $e(m) = m$. Arbitrarily choose a tuple $(x, y, z) \in (x \times y = z)$. Let $x = x_1^{\alpha_1} \cdot \ldots \cdot x_a^{\alpha_a}$ and $y = y_1^{\beta_1} \cdot \ldots \cdot y_b^{\beta_b}$ be prime factorisations. Note that at most one of $x_1, \ldots, x_a, y_1, \ldots, y_b$ can equal p and, if so, the corresponding exponent must equal one. If none of the factors equal p, then $e(x) = e(y) = e(z) = 1$ and $e(x) \times e(y) = e(z)$. Otherwise, assume without loss of generality that $x_1 = p$. Then we have $e(x) = e(z) = m$ and $e(y) = 1$. Once again $e(x) \times e(y) = e(z)$ and e is indeed an endomorphism of $(\mathrm{Div}_m; \times, m)$. It follows that $(\{1, m\}; \times, m)$ is the core of $(\mathrm{Div}_m; \times, m)$.

Lemma 8. *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.*

Proof. Assume m has the prime factorization $m = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}$ and note that $\alpha_1, \ldots, \alpha_k > 1$. Assume $e : \mathrm{Div}_m \to \{a, b\}$ is an endomorphism to a two-element core, i.e. the range of e is $\{1, m\}$. Since multiplication is determined by the action of the primes, we can see that for one prime $p \in \{p_1, \ldots, p_k\}$ we must have $e(p) = m$. Consider $p \times p = p^2$. If we apply the endomorphism e to this tuple, we end up with $e(p) \times e(p) = e(p)^2$ which is not possible. Hence, e does not exist and $(\mathrm{Div}_m; \times, m)$ does not admit a two-element core.

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

Proposition 13. CSP($\mathbb{N}; \neq, \times$) *is* NP*-complete.*

Proof. NP membership follows from Proposition 11. For NP-hardness, we will encode the CSP of a certain Boolean constraint language, i.e. with domain $\{0,1\}$, with two relations: $\neq$ and $R_1 := \{0,1\}^3 \setminus \{(1,1,1)\}$. This CSP is NP-hard because $\neq$ omits constant and semilattice polymorphisms and R_1 omits majority and minority polymorphisms (an algebraic reformulation of Schaefer's Theorem [9] in the spirit of [35]).

To encode our Boolean CSP, ensure all variables v satisfy $v \times v = v$, which enforces the domain $\{0,1\}$. Consider 0 to be false and 1 to be true. 0 is pp-definable by $x \times x = x \wedge \exists y\ y \neq x \wedge x \times y = x$. For $\{0,1\}^3 \setminus \{(1,1,1)\}$ take $R_1(x,y,z)$ to be $\exists w\ x \times y = w \wedge w \times z = 0$; and for $\neq$ take $\neq$. The reduction may now be done by local substitution and the result follows.

An operation $t : D^k \to D$ is a *weak near-unanimity* operation if t satisfies

$$t(x,\ldots,x) = x, \text{ and}$$
$$t(y,x,\ldots,x) = t(x,y,x,\ldots,x) = \cdots = t(x,\ldots,x,y).$$

Theorem 3 ([50]). *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* CSP(Γ) *is* NP*-hard.*

Lemma 9. *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* CSP($S; \times, m$) *is* NP*-hard.*

Proof. Assume $(S; \times, m)$ admits a weak near-unanimity operation $t : S^k \to S$. The relation $\prod_{i=1}^{k} x_i = x_{k+1}$ is pp-definable in $(S; \times, m)$ and so is the relation

$$R = \{(x_1,\ldots,x_k) \in S^k \mid \prod_{i=1}^{k} x_i = m\}.$$

The relation R contains the tuples

$$\begin{gathered}(m,1,\ldots,1)\\(1,m,1,\ldots,1)\\\vdots\\(1,\ldots,1,m).\end{gathered}$$

Applying t component-wise (i.e. vertically) to these tuples yields a tuple $(a, \dots, a)$ for some $a \in D$. However, R does not contain a tuple $(a, \dots, a)$ for any $a \in D$ since this would imply that $m = a^k$ for some $a, k > 1$. We conclude that $\mathrm{CSP}(S; \times, m)$ is NP-hard by Theorem 3.

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* $m > 1$.

Proof. If $m = k^n$ for some $k, n > 1$, then we can pp-define the constant relation $\{k\}$ since $x = k \Leftrightarrow \prod_{i=1}^{k} x = m$. Hence, we assume without loss of generality that $m \neq k^n$ for all $k, n > 1$.

We further know that Div_m is pp-definable in $(\mathbb{N}; \times, m)$, i.e. there is polynomial time reduction from $(\mathrm{Div}_m; \times, m)$ to $(\mathbb{N}; \times, m)$. The core of $(\mathrm{Div}_m; \times, m)$ is some $(S; \times, m)$, where $\{1, m\} \subseteq S \subseteq \mathrm{Div}_m$ and the result follows from Lemma 9.

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.*

Proof. From Lemma 7, for each $x \in U$, the core of $(\mathbb{N}; \times, x)$ is the same (up to isomorphism). Fix some $m \in U$. We claim there is a polynomial time reduction from $\mathrm{CSP}(\mathrm{Div}_m; \times, m)$ to $\mathrm{CSP}(\mathbb{N}; \times, U)$, whereupon the result follows from Theorem 4.

To see the claim, take an instance ϕ of $\mathrm{CSP}(\mathrm{Div}_m; \times, m)$ and build an instance ψ of $\mathrm{CSP}(\mathbb{N}; \times, U)$ by adding an additional variable v_m, now substituting instances of m for v_m, and adding the constraint $U(v_m)$. Correctness of the reduction is easy to see and the result follows.

For $x \in \mathbb{N} \setminus \{0, 1\}$, define its *minimal exponent*, 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*, basis(U) to be the set $\{\text{min-exp}(x) : x \in U\}$.

Lemma 10. *Let* $U \subseteq \mathbb{N} \setminus \{0, 1\}$*, so that* basis$(U)$ *is finite and* basis$(U) \neq \{1\}$*. There is some set* X *pp-definable in* $(\mathbb{N}; \times, U)$ *so that* basis$(X) = \{1\}$*.*

Proof. Let $r = \max(\mathrm{basis}(U))$ and take an element x that witnesses this, of the form $q^r \cdot p_1^{a_1} \cdots p_k^{a_k}$, where $p_1, \ldots, p_k$ are prime and each is coprime to q (which is square-free), and where $a_1, \ldots, a_k > r$. Set

$$\xi(y) := \exists z, x_1, \ldots, x_k \; U(z) \wedge y^r \cdot x_1^{a_1} \cdots x_k^{a_k} = z.$$

We claim that ξ defines a set of integers X so that $\mathrm{basis}(X)$ has the desired property. The non-emptiness is clear since $1 \in \mathrm{basis}(X)$ by construction.

Firstly, we will by contradiction argue that $0 \notin \mathrm{basis}(X)$. Assume $0 \in \mathrm{basis}(X)$. This implies that $1 \in X$. Hence, $\exists z, x_1, ..., x_k \; U(z) \wedge 1^r \cdot x_1^{a_1} \cdots x_k^{a_k} = z$, so $x_1^{a_1} \cdots x_k^{a_k} \in U$. It follows that $d = \min\{a_1, \ldots, a_k\} \in \mathrm{basis}(U)$. Now, $a_1, \ldots, a_k > r$ so $d > r$. This contradicts the fact that $r = \max(\mathrm{basis}(U))$.

We will now argue by contradiction that $1 < s \notin \mathrm{basis}(X)$. Assume $s \in \mathrm{basis}(X)$. Then there exists a $t = q^s \cdot p_1^{a_1} \cdots p_k^{a_k} \in X$ where $s < a_1, \ldots, a_k$. Since $t \in X$, we know that $\exists z, x_1, \ldots, x_k \; U(z) \wedge t^s \cdot x_1^{a_1} \cdots x_k^{a_k} = z$. Let $e = t^r \cdot x_1^{a_1} \cdots x_k^{a_k} \in U$ as above. Let's expand t: $e = (q^s \cdot p_1^{a_1} \cdots p_k^{a_k})^r \cdot x_1^{a_1} \cdots x_k^{a_k}$. We see that $\text{min-exp}(e) > r$ which contradicts the choice of r.

Example 1. *We provide an example of the construction of the previous lemma* in vivo. *Let* $U := \{p^2, p^2q^3, p^4q^4r^8 : \; p, q, r \textit{ primes}\}$, *so that* $\mathrm{basis}(U) = \{2, 4\}$. *Then* $\xi(y) := \exists z, x \; U(z) \wedge y^4 \cdot x^8 = z$. *We can now deduce* $X := \{p, pq^2 : \; p, q \textit{ primes}\}$ *and* $\mathrm{basis}(X) = \{1\}$.

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

Proof. Membership of NP follows from Proposition 11. We use the construction of the previous lemma to pp-define X with $\mathrm{basis}(X) = \{1\}$. This allows us to polynomially reduce $\mathrm{CSP}(\mathbb{N}; \times, X)$ to $\mathrm{CSP}(\mathbb{N}; \times, U)$ by local substitution. NP-hardness for the former comes from Theorem 5 and the result follows.

5. Final remarks

In this paper we have provided a solution to the major open question from [1] as well as begun the investigation of CSPs associated with Skolem Arithmetic. However, the thrust of our work must be considered exploratory and 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 [1]. 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 CSP($\mathcal{P}(\mathbb{N}); \mathcal{O}$), where $\mathcal{P}(\mathbb{N})$ is the power set of $\mathbb{N}$, rather than the somewhat esoteric 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 [51]).

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

5.1. Acknowledgements

We are grateful for the input of several anonymous reviewers for suggesting ways to improve the readability of our article.

References

[1] C. Glaßer, C. Reitwießner, S. D. Travers, M. Waldherr, Satisfiability of algebraic circuits over sets of natural numbers, Discrete Applied Mathematics 158 (13) (2010) 1394–1403.

[2] T. Skolem, Über gewisse Satzfunktionen in der Arithmetik, Skr. Norske Videnskaps-Akademie i Oslo.

[3] A. Mostowski, On direct products of theories, The Journal of Symbolic Logic 17 (1952) 1–31.

[4] M. Presburger, 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.

[5] A. Bès, A survey of arithmetical definability, Soc. Math. Belgique (2002).

[6] E. C. Freuder, Complexity of k-tree structured constraint satisfaction problems, in: Proceedings of the 8th National Conference on Artificial Intelligence, 1990, pp. 4–9.

[7] M. Grohe, The complexity of homomorphism and constraint satisfaction problems seen from the other side, J. ACM 54 (1).

[8] T. Feder, M. Vardi, 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.

[9] T. J. Schaefer, The complexity of satisfiability problems, in: Proceedings of STOC'78, 1978, pp. 216–226.

[10] A. Bulatov, A dichotomy theorem for constraint satisfaction problems on a 3-element set, J. ACM 53 (1) (2006) 66–120.

[11] P. Hell, J. Nešetřil, On the complexity of H-coloring, Journal of Combinatorial Theory, Series B 48 (1990) 92–110.

[12] L. Barto, M. Kozik, T. Niven, 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.

[13] A. Rafiey, J. Kinne, T. Feder, Dichotomy for digraph homomorphism problems, CoRR abs/1701.02409.
URL `http://arxiv.org/abs/1701.02409`

[14] A. A. Bulatov, A dichotomy theorem for nonuniform csps, CoRR abs/1703.03021, accepted at FOCS 2017.
URL `http://arxiv.org/abs/1703.03021`

[15] D. Zhuk, The proof of CSP dichotomy conjecture, CoRR abs/1704.01914, accepted at FOCS 2017.
URL `http://arxiv.org/abs/1704.01914`

[16] A. Bulatov, A. Krokhin, P. G. Jeavons, Classifying the complexity of constraints using finite algebras, SIAM Journal on Computing 34 (2005) 720–742.

[17] M. Bodirsky, M. Grohe, Non-dichotomies in constraint satisfaction complexity, in: Proceedings of ICALP'08, 2008, pp. 184–196.

[18] M. Bodirsky, J. Kára, The complexity of temporal constraint satisfaction problems, J. ACM 57 (2).

[19] M. Bodirsky, M. Pinsker, 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.

[20] M. Bodirsky, B. Martin, A. Mottet, Constraint satisfaction problems over the integers with successor, in: Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I, 2015, pp. 256–267.

[21] M. Bodirsky, P. Jonsson, T. von Oertzen, Essential convexity and complexity of semi-algebraic constraints, Logical Methods in Computer Science 8 (4), an extended abstract about a subset of the results has been published under the title *Semilinear Program Feasibility* at ICALP'10.

[22] L. J. Stockmeyer, A. R. Meyer, Word problems requiring exponential time: Preliminary report, in: Proceedings of the 5th Annual ACM Symposium on Theory of Computing, (STOC), 1973, pp. 1–9.

[23] K. Wagner, The complexity of problems concerning graphs with regularities (extended abstract), in: Proceedings of the Mathematical Foundations of Computer Science 1984, Springer-Verlag, London, UK, UK, 1984, pp. 544–552.

[24] K. Yang, Integer circuit evaluation is pspace-complete, J. Comput. Syst. Sci. 63 (2) (2001) 288–303, an extended abstract of appeared at CCC 2000.

[25] P. McKenzie, K. W. Wagner, 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.

[26] S. D. Travers, The complexity of membership problems for circuits over sets of integers, Theoretical Computer Science 369 (1-3) (2006) 211–229.

[27] H. Breunig, The complexity of membership problems for circuits over sets of positive numbers, in: International Symposium on Fundamentals of Computation Theory, Vol. 4639 of Lecture Notes in Computer Science, Springer, 2007, pp. 125–136.

[28] C. Glaßer, K. Herr, C. Reitwießner, S. D. Travers, M. Waldherr, Equivalence problems for circuits over sets of natural numbers, Theory of Computing Systems 46 (1) (2010) 80–103.

[29] I. Pratt-Hartmann, I. Düntsch, Functions definable by arithmetic circuits, in: Conference on Mathematical Theory and Computational Practice, Vol. 5635 of Lecture Notes in Computer Science, Springer, 2009, pp. 409–418.

[30] A. Jez, A. Okhotin, Complexity of equations over sets of natural numbers, Theoretical Computer Science 48 (2) (2011) 319–342.

[31] A. Jez, A. Okhotin, Computational completeness of equations over sets of natural numbers, Information and Computation 237 (2014) 56–94.

[32] S. Salehi, On the multiplicative theory of numbers, CoRR abs/1707.04732.
URL `https://arxiv.org/abs/1707.04732`

[33] M. Bodirsky, Cores of countably categorical structures, Logical Methods in Computer Science (LMCS)DOI: 10.2168/LMCS-3(1:2).

[34] T. Feder, F. R. Madelaine, I. A. Stewart, Dichotomies for classes of homomorphism problems involving unary functions, Theor. Comput. Sci. 314 (1-2) (2004) 1–43.

[35] P. G. Jeavons, On the algebraic structure of combinatorial problems, Theoretical Computer Science 200 (1998) 185–204.

[36] C. Smorynski, The incompleteness theorems, in: J. Barwise (Ed.), Handbook of Mathematical Logic, North-Holland, Amsterdam, 1977, pp. 821–865.

[37] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967.

[38] C. H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

[39] J. Ferrante, C. Rackoff, A decision procedure for the first order theory of real addition with order, SIAM J. Comput. 4 (1975) 69–76.

[40] J. Ferrante, C. W. Rackoff, The Computational Complexity of Logical Theories, Vol. 718 of Lecture Notes in Mathematics, Springer Verlag, 1979.

[41] Y. V. Matiyasevich, Enumerable sets are diophantine, Doklady Akad. Nauk SSSR 191 (1970) 279–282, translation in Soviet Math. Doklady, 11:354–357, 1970.

[42] M. Davis, H. Putnam, J. Robinson, The decision problem for exponential Diophantine equations, Annals of Mathematics 74 (2) (1961) 425–436.

[43] I. Borosh, L. B. Treybig, Bounds on positive integral solutions of linear Diophantine equations, Proceedings American Mathematical Society 55 (1976) 299–304.

[44] C. H. Papadimitriou, On the complexity of integer programming, Journal of the ACM 28 (4) (1981) 765–768.

[45] B. Scarpellini, Complexity of subcases of presburger arithmetic, Transactions of the American Mathematical Society 284 (1) (1984) 203–218.

[46] E. Grädel, Dominoes and the complexity of subclasses of logical theories, Ann. Pure Appl. Logic 43 (1) (1989) 1–30.

[47] D. Barth, M. Beck, T. Dose, C. Glaßer, L. Michler, M. Technau, Emptiness problems for integer circuits, Electronic Colloquium on Computational Complexity (ECCC) 24 (2017) 12, to appear at MFCS 2017. URL https://eccc.weizmann.ac.il/report/2017/012

[48] A. Schrijver, Theory of Linear and Integer Programming, John Wiley & Sons, Inc., New York, NY, USA, 1986.

[49] P. Jonsson, T. Lööw, Computational complexity of linear constraints over the integers, Artificial Intelligence 195 (2013) 44–62, an extended abstract appeared at IJCAI 2011.

[50] L. Barto, M. Kozik, Constraint satisfaction problems of bounded width, in: 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA, 2009, pp. 595–603.

[51] T. Dose, Complexity of constraint satisfaction problems over finite subsets of natural numbers, in: 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland, 2016, pp. 32:1–32:13. doi:10.4230/LIPIcs.MFCS.2016.32.
URL `https://doi.org/10.4230/LIPIcs.MFCS.2016.32`