Balance Problems for Integer Circuits

Titus Dose
Institute of Computer Science, Julius-Maximilians Universität Würzburg, Germany
titus.dose@uni-wuerzburg.de

Abstract

We investigate the computational complexity of *balance problems* for $\{-, \cdot\}$-circuits computing finite sets of natural numbers. These problems naturally build on problems for integer expressions and integer circuits studied by Stockmeyer and Meyer (1973), McKenzie and Wagner (2007), and Glaßer et al. (2010).

Our work shows that the balance problem for $\{-, \cdot\}$-circuits is undecidable which is the first natural problem for integer circuits or related constraint satisfaction problems that admits only one arithmetic operation and is proven to be undecidable.

Starting from this result we precisely characterize the complexity of balance problems for proper subsets of $\{-, \cdot\}$. These problems turn out to be complete for one of the classes L, NL, and NP.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness, Theory of computation → Computability

Keywords and phrases computational complexity, integer expressions, integer circuits

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.5

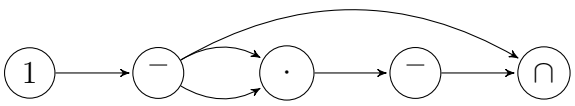
Related Version For a more comprehensive presentation of the results we refer to the technical report [6], `https://eccc.weizmann.ac.il/report/2018/055`.

1 Introduction

In 1973, Stockmeyer and Meyer [18] defined and studied membership and equivalence problems for *integer expressions*. They considered expressions built up from single natural numbers by using set operations ($\cup$, $\cap$, $\overline{\ }$), pairwise addition ($+$), and pairwise multiplication ($\cdot$). For example, $\overline{\overline{1} \cdot \overline{1}} \cap \overline{1}$ describes the set of primes $\mathbb{P}$.

The *membership problem for integer expressions* asks whether some given number is contained in the set described by a given integer expression, whereas the *equivalence problem for integer expressions* asks whether two given integer expression describe the same set. Restricting the set of allowed operations results in problems of different complexities.

Wagner [20] studied a more succinct way to represent such expressions, namely *circuits over sets of natural numbers*, also called integer circuits. Each input gate of such a circuit is labeled with a natural number, the inner gates compute set operations and arithmetic operations ($\cup$, $\cap$, $\overline{\ }$, $+$, $\cdot$). The following circuit with only 4 inner gates computes the set of primes.



Starting from this circuit, one can use integer circuits to express fundamental number theoretic questions: thus, a circuit describing the set of all twin primes or the set of all

Sophie Germain primes can be constructed. McKenzie and Wagner [15] constructed a circuit C computing a set that contains 0 if and only if the Goldbach conjecture holds.

Wagner [20], Yang [21], and McKenzie and Wagner [15] investigated the complexity of membership problems for circuits over natural numbers: here, for a given circuit C, one has to decide whether a given number n belongs to the set described by C. Travers [19] and Breunig [2] considered membership problems for circuits over integers and positive integers, respectively. Glaßer et al. [9] studied *equivalence problems for circuits over sets of natural numbers*, i.e., the problem of deciding whether two given circuits compute the same set.

Satisfiability problems for circuits over sets of natural numbers, investigated by Glaßer et al. [11], are a generalization of the membership problems investigated by McKenzie and Wagner [15]: the circuits can have *unassigned input gates* and the question is: on input of a circuit C with gate labels from $\mathcal{O} \subseteq \{\cup, \cap, \bar{}, +, \cdot\}$ and a natural number b, does there exist an assignment of the unassigned input gates with natural numbers such that b is contained in the set described by the circuit?

Barth et al. [1] investigated emptiness problems for integer circuits. Here, for both circuits with unassigned inputs and circuits without unassigned inputs, the question of whether an integer circuit computes the empty set (for some/all assignment(s) if the circuits allow unassigned inputs) is raised and investigated.

Apart from the mentioned research on circuit problems there has been work on related variants like functions computed by circuits [17] and constraint satisfaction problems (csp) over natural numbers [10, 5]. The constraint satisfaction problems by Glaßer, Jonsson, and Martin [10] can be considered as conjunctions of equations of integer expressions with variables standing for singleton sets of natural numbers. Here the question is whether there is an assignment of the variables such that all equations are satisfied. These constraint satisfaction problems have the peculiarity that expressions describe sets of integers whereas variables can only store singleton sets of natural numbers. Dose [5] addressed this and studied constraint satisfaction problems over finite subsets of $\mathbb{N}$, consequently replaced the set complement $\bar{}$ with the set difference $-$, and allowed the variables to describe arbitrary finite subsets of $\mathbb{N}$.

Our Model and Contributions

The definition of the circuits investigated in this paper follows the definition of previous papers such as [15, 9, 11, 1]. Yet there are some differences:

Our circuit problems are about *balanced sets* where a finite and non-empty set $S \subseteq \mathbb{N}$ is balanced if $|S| = |\{0, 1, \ldots, \max(S)\} - S|$. Analogously, S is unbalanced if $|S| \neq |\{0, 1, \ldots, \max(S)\} - S|$. That means, the maximum of a set marks the relevant area and then we ask whether there are as many elements inside the set as outside of it. As the notion of balanced sets only makes sense for finite sets, our circuits should solely compute finite sets. Due to that we replace the commonly used set complement $\bar{}$ with the set difference $-$. Now, as the circuits only work over the domain of finite subsets of $\mathbb{N}$, it suggests itself to also allow the input gates of a circuit to compute arbitrary finite subsets of $\mathbb{N}$ and not only singleton sets (cf. Dose [5] where the analogous step was made for constraint satisfaction problems).

For such circuits we ask: is there an assignment of the unassigned inputs with arbitrary finite subsets of $\mathbb{N}$ under which the circuit computes a balanced set? This problem is denoted by $\mathrm{BC}(\mathcal{O})$, where $\mathcal{O} \subseteq \{\cup, \cap, -, +, \cdot\}$ is the set of allowed operations.

The notion of balance is important in computational complexity. It occurs when considering counting classes [12] like $\mathrm{C_{=}L}$ or $\mathrm{C_{=}P}$ for instance. There, the question is whether for some problem A there is a non-deterministic logarithmic space or polynomial-time machine

M accepting A, where M accepts some input x if and only if the number of accepting paths equals the number of rejecting paths.

Balance problems for integer circuits are interesting for another reason. To our knowledge, there is no natural decision problem for integer circuits or constraint satisfaction problems over sets of natural numbers that allows only one arithmetic operation and is known to be undecidable. In this paper, however, it is shown that $\mathrm{BC}(-,\cdot)$ is undecidable.

Starting from this undecidable problem $\mathrm{BC}(-,\cdot)$, we also investigate $\mathrm{BC}(\mathcal{O})$ for arbitrary proper subsets of $\{-,\cdot\}$ and precisely characterize the complexity of each such problem. It turns out that all these problems are in NP. In detail, we show that $\mathrm{BC}(\cdot)$ is NL-complete, $\mathrm{BC}(-)$ is NP-complete, and $\mathrm{BC}(\emptyset) \in \mathrm{L}$.

2 Preliminaries

Basic Notions

Let $\mathbb{N}$ denote the set of natural numbers. $\mathbb{N}^+ = \mathbb{N} - \{0\}$ is the set of positive naturals. Moreover, the set of primes is denoted by $\mathbb{P}$.

We extend the arithmetical operations $+$ and $\cdot$ to sets of naturals: for $A, B \subseteq \mathbb{N}$ define $A + B = \{a + b \mid a \in A, b \in B\}$ and $A \cdot B = \{a \cdot b \mid a \in A, b \in B\}$. In contrast to previous papers, in this paper the multiplication of sets is not denoted by $\times$ but by $\cdot$. Instead, $\times$ denotes the cartesian product. Furthermore, for arbitrary sets, the operations $\cup$, $\cap$, and $-$ define the union, intersection, and set difference, respectively. The power set of a set M is denoted by $\mathcal{P}(M)$ whereas $\mathcal{P}_{\mathrm{fin}}(M) = \{A \in \mathcal{P}(M) \mid A \text{ finite}\}$. For a finite and non-empty set S let $\max(S)$ (resp., $\min(S)$) denote the maximum (resp., minimum) number of S. Finite intervals $\{x \mid a \leq x \leq b\}$ for $a, b \in \mathbb{Z}$ are denoted by $[a, b]$.

L, NL, and NP denote standard complexity classes [16] and RE is the set of computably enumerable problems.

For problems A and B we say that A is (logarithmic-space) many-one reducible to B if there is some (logarithmic-space) computable function f with $c_A(x) = c_B(f(x))$, where c_X for a set X is the characteristic function of X. We denote this by $A \leq_{\mathrm{m}} B$ (resp., $A \leq_{\mathrm{m}}^{\log} B$).

For pairs (A, B) and (C, D) with $A \cap B = C \cap D = \emptyset$ we say that (A, B) is many-one reducible to (C, D) (denoted as $(A, B) \leq_{\mathrm{m}} (C, D)$) if there is a computable function f with $x \in A \Rightarrow f(x) \in C$ and $x \in B \Rightarrow f(x) \in D$. Note that if $B = \overline{A}$ and $D = \overline{C}$ this coincides with the usual many-one reducibility, i.e., $(A, \overline{A}) \leq_{\mathrm{m}} (C, \overline{C}) \Leftrightarrow A \leq_{\mathrm{m}} C$.

CSAT is the circuit satisfiability problem, i.e., the problem of determining whether a given Boolean circuit has an assignment of the unassigned inputs that makes the output gate true. The problem is $\leq_{\mathrm{m}}^{\log}$-complete for NP via a trivial reduction from SAT which itself can be shown to be $\leq_{\mathrm{m}}^{\log}$-complete for NP via a construction by Cook [3].

Balanced Sets

A finite and non-empty set $S \subseteq \mathbb{N}$ is *balanced* (resp., *unbalanced*) if $|S| = |\{0, 1, \ldots, \max(S)\} - S|$ (resp., $|S| \neq |\{0, 1, \ldots, \max(S)\} - S|$). Intuitively spoken, $\max(S)$ defines the universe $\{0, 1, \ldots, \max(S)\}$ and then S is balanced if it contains the same number of elements as its complement. Note that the notion of balance/unbalance only makes sense if there is some maximum element defining the universe. Hence the empty set is neither balanced nor unbalanced.

The following lemma immediately follows from the definition.

▶ **Lemma 1.** *Let $S \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ be balanced. Then $S \neq \emptyset$ and $\max(S)$ is odd.*

Moreover, we say that S is *subbalanced* if $|S| < (\max(S)+1)/2$ which is equivalent to $|S| \leq \max(S)/2$. As we want to investigate the complexity of balance problems with respect to deterministic logarithmic-space reductions, it is important to see that the test of whether some input set is balanced can be done in deterministic logarithmic space. Define $\mathrm{Bal} = \{S \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N}) \mid S \text{ is balanced}\}$ and the slightly more general problem $\mathrm{Bal}_M = \{S \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N}) \mid M \cdot S \text{ is balanced}\}$ for a non-empty and finite set M. Standard arguments yield the following proposition.

▶ **Proposition 2.** $\mathrm{Bal}_M \in \mathrm{L}$. *In particular,* $\mathrm{Bal} \in \mathrm{L}$.

Circuits and Balance Problems for Circuits

In previous papers such as [1] it was differentiated between completely and partially assigned circuits. As we restrict on partially assigned circuits in this paper, we define circuits in general as partially assigned circuits.

A *circuit* C is a triple (V, E, g_C) where (V, E) is a finite, non-empty, directed, acyclic graph with a designated vertex $g_C \in V$ and a topologically ordered vertex set $V \subseteq \mathbb{N}$, i.e., if $u, v \in V$ are vertices with $u < v$, then there is no edge from v to u. Here, graphs may contain multi-edges and are not necessarily connected. But we require that C is topologically ordered. Note that the test of whether a graph is topologically ordered or not is possible in deterministic logarithmic space. Consequently, we are able to check in deterministic logarithmic space whether an input graph is acyclic. Hence there is a deterministic logarithmic-space algorithm that on input of a graph tests whether the input is a circuit. Therefore, when presenting algorithms for circuits we may always assume that the input is a valid circuit.

Let $\mathcal{O} \subseteq \{\cup, \cap, -, +, \cdot\}$. An *$\mathcal{O}$-circuit* (or *circuit* for short if $\mathcal{O}$ is apparent from the context) is a quintuple $C = (V, E, g_C, \alpha, \beta)$ where (V, E, g_C) is a circuit whose nodes are labeled by the *labeling function* $\alpha : V \to \mathcal{O} \cup \mathcal{P}_{\mathrm{fin}}(\mathbb{N}) \cup \{\square\}$ such that each node has indegree 0 or 2, nodes with indegree 0 have a label from $\mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ (encoded as a list of all the numbers in the set) or from $\{\square\}$, and nodes with indegree 2 have labels from $\mathcal{O}$. Moreover, β is a function $E \to \{\mathrm{l}, \mathrm{r}\}$ and we require that for each node u with predecessors u_1 and u_2 it holds $\{\beta(u_1), \beta(u_2)\} = \{\mathrm{l}, \mathrm{r}\}$. Thus, β marks whether an edge starts in the left or right predecessor of the node it points to.

In the context of circuits, nodes are also called gates. A gate with indegree 0 is called *input gate*, all other nodes are *inner gates*, the designated gate g_C is also called *output gate*. Input gates with a label from $\mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ are *assigned input gates* whereas input gates with label $\square$ are *unassigned input gates*.

$\mathcal{O}$-circuits are also called *integer circuits*. If g is some gate of C with $\alpha(g) = \otimes \in \mathcal{O}$ and with predecessors g' and g'' satisfying $\beta(g') = \mathrm{l}$ and $\beta(g'') = \mathrm{r}$, then we also write $g = g' \otimes g''$.

For an $\mathcal{O}$-circuit C with unassigned input gates $g_1 < \cdots < g_n$ and $X_1, \ldots, X_n \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$, let $C(X_1, \ldots, X_n)$ be the circuit that arises from C by modifying the labeling function α such that $\alpha(g_i) = X_i$ for every $1 \leq i \leq n$.

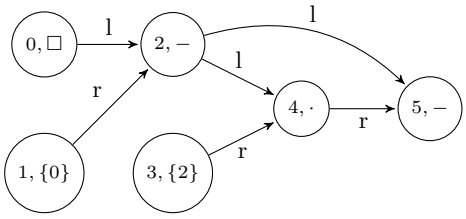
For a circuit $C = (V, E, g_C, \alpha)$ without unassigned input gates we inductively define the set $I(g; C)$ computed by a gate $g \in V$ for $g = 1, \ldots, |V|$ by

$$I(g; C) = \begin{cases} \alpha(g) \subseteq \mathbb{N} & \text{if } g \text{ has indegree 0,} \\ I(g', C) \otimes I(g'', C) & \text{if } g = g' \otimes g''. \end{cases}$$

The set computed by the circuit is denoted by $I(C)$ and defined to be the set computed by the output gate $I(g_C; C)$.

It is convenient to introduce notations for basic constructions of circuits. For $X \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ we use X as an abbreviation for the circuit $(\{1\}, \varnothing, \{1\}, 1 \mapsto X)$. For $\mathcal{O}$-circuits C, C' for some $\mathcal{O}$ and $\otimes \in \{\cup, \cap, , -+, \cdot\}$ let $C \otimes C'$ be the circuit obtained from C' and C'' by feeding their output gates to the new output gate $\otimes$. This is possible in logarithmic space.

As an example, for an unassigned input gate $g = 0$, consider the circuit $C = (g - \{0\}) - ((g - \{0\}) \cdot \{2\})$, which is the following circuit



where each node is given by its number and its label. The node 5 is the output gate and it computes the set $\{1\}$ if and only if $I(2; C)$ is a set of the form $\{2^0, 2^1, 2^2, \ldots, 2^r\}$ for $r \in \mathbb{N}$.

Now we define the problems this paper focuses on.

▶ **Definition 3.** Let $\mathcal{O} \subseteq \{-, \cup, \cap, +, \cdot\}$ and define

$$\text{BC}(\mathcal{O}) = \{C \mid C \text{ is an } \mathcal{O}\text{-circuit with } n \text{ unassigned inputs and there exist } X_1, \ldots, X_n \in \mathcal{P}_{\text{fin}}(\mathbb{N}) \text{ such that } I(C(X_1, \ldots, X_n)) \text{ is balanced}\}.$$

We use the following abbreviations if confusions are impossible: we write g or $I(g)$ for $I(g; C)$, where C is a circuit and g is a gate of C; we write C for $I(C)$, where C is a circuit; we write $\text{BC}(-, \cdot)$ for $\text{BC}(\{-, \cdot\})$ and the like.

3 Set Difference and Multiplication Lead to Undecidability

This section contains our main result: the undecidability of $\text{BC}(-, \cdot)$ which is achieved by a reduction from a famously known RE-complete problem. According to the Matiyasevich-Robinson-Davis-Putnam theorem [14, 4] the problem of determining whether there is a solution for a given Diophantine equation is RE-complete. It can be derived by standard arguments that also the following problem is RE-complete (with regard to $\leq_{\text{m}}$).

$$\text{DE} = \{(p(x_1, \ldots, x_n), q(x_1, \ldots, x_n)) \mid \exists a_1, \ldots, a_n \in \mathbb{N}^+,\ p(a_1, \ldots, a_n) = q(a_1, \ldots, a_n) \text{ for multivariate polynomials } p \text{ and } q \text{ with coefficients from } \mathbb{N}^+\}.$$

▶ **Theorem 4.** $\text{BC}(-, \cdot)$ *is* RE*-complete.*

Let for the remainder of this section $\mathcal{O} = \{-, \cdot\}$ unless stated differently. For the sake of brevity, we make use of intersection gates but note that $A \cap B$ is just an abbreviation for $A - (A - B)$. Further abbreviated notations are $A - \bigcup_{i=1}^{n} B_i$ for $(\ldots((A - B_1) - B_2) - \ldots) - B_n$ and $A - (\bigcup_{i=1}^{n} B_i - \{1\})$ for $(\ldots((A - (B_1 - \{1\})) - (B_2 - \{1\})) - \ldots - (B_n - \{1\})$.

In order to prove Theorem 4 we define a slightly different version of the problem $\text{BC}(-, \cdot)$ which can be reduced to the original version in logarithmic space.

▶ **Definition 5.** Define

$$\text{BC}'(\mathcal{O}) = \{(C, Q) \mid C \text{ is a partially assigned } \mathcal{O}\text{-circuit, } Q \text{ is a subset of the nodes of } C, \text{ and there exist } X_1, \ldots, X_n \in \mathcal{P}_{\text{fin}}(\mathbb{N}^+) \text{ such that } I(C(X_1, \ldots, X_n)) \text{ is balanced and } I(K; C(X_1, \ldots, X_n)) = \{1\} \text{ for all } K \in Q\}.$$

For the sake of simplicity, we call instances of $\text{BC}'(\mathcal{O})$ $\mathcal{O}$-circuits as well.

▶ **Lemma 6.** *1. For $K \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ with $\kappa := \max(K) \geq 3$ it holds $|K \cdot K \cdot K| < \kappa^3/2$.*
2. $\text{BC}'(\mathcal{O}) \leq_{\text{m}}^{\log} \text{BC}(\mathcal{O})$ for $\mathcal{O} = \{-, \cdot\}$.

Proof sketch. We only sketch the proof of statement 2. Let C be a partially assigned $\mathcal{O}$-circuit with output node g_C and let Q be a subset of the nodes of C. Starting with this circuit, we build a new circuit and denote this modified circuit by C':

For each assigned or unassigned input node g, add a node g' of type $-$ which computes the set $g - \{0\}$, replace all edges (g, h) with (g', h), and in case $g \in Q$, remove g from Q and add g'. Then add a new output node $g_{C'} = g_C \cdot \prod_{K \in Q}(K \cdot K \cdot K)$.

Making use of Statement 1, it can be proved that $\text{BC}'(\mathcal{O}) \leq_{\text{m}}^{\log} \text{BC}(\mathcal{O})$ via $C \mapsto C'$. ◀

Before proving Theorem 4 we introduce some $\mathcal{O}$-circuits which will be used extensively as components of circuits expressing Diophantine equations.

▶ **Lemma 7.** *For every finite $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{P}$ with $n = |P| \geq 1$ there is an $\mathcal{O}$-circuit (C_P, Q_P) containing gates $g_P^1, \ldots, g_P^n$ satisfying the following properties:*

1. *For an arbitrary assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ it holds:*
   if $K = \{1\}$ for all $K \in Q_P$, then $\exists_{m \in \mathbb{N}} \, \forall_{i=1,\ldots,n} \, g_P^i = \{1, p_i, \ldots, p_i^m\}$.
2. *For each $m \in \mathbb{N}$ there is an assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ under which $g_P^i = \{1, p_i, \ldots, p_i^m\}$ and $K = \{1\}$ for all $K \in Q_P$.*

Proof sketch. Construct the $\mathcal{O}$-circuit (C_P, Q_P) as follows.

- For each $p \in P$ insert an input gate X_p and gates $h_p = X_p - (X_p \cdot \{p\})$ and $h'_p = (\{1, p\} \cdot X_p) - (X_p - \{1\})$. Put all the nodes h_p into Q_P.
- Similarly, for $k \in \{p_1 \cdot p_2, p_2 \cdot p_3, \ldots, p_{n-1} \cdot p_n\}$ insert an input gate X_k and gates $h_k = X_k - (X_k \cdot \{k\})$ and $h'_k = (\{1, k\} \cdot X_k) - (X_k - \{1\})$. Insert all nodes h_k into Q_P.
- For each $k = p_i \cdot p_{i+1}$ with $i \in \{1, \ldots, n-1\}$ add a node $\gamma_k = h'_k - \big((h'_{p_i} \cdot h'_{p_{i+1}}) - \{1\}\big)$ and let Q_P contain all these nodes.
- Denote $g_P^i = X_{p_i}$.

It can be shown that (C_P, Q_P) satisfies the requirements of the lemma. ◀

By adding nodes of the form $\prod_{j=1}^{i} g_P^j$ for some i we receive the following.

▶ **Lemma 8.** *For every finite $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{P}$ with $n = |P| \geq 1$ there is an $\mathcal{O}$-circuit (D_P, Q_P) with gates $g_P^0, g_P^1, \ldots, g_P^n$ satisfying the following properties:*

1. *For an arbitrary assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ it holds*

$$\forall_{K \in Q_P} \, K = \{1\} \quad \Rightarrow \quad \exists_{m \in \mathbb{N}^+} \forall_{i=0,\ldots,n} \, |g_P^i| = m^i, \; 1 \in g_P^i, \text{ and the prime divisors of numbers in } g_P^i \text{ are all in } P.$$

2. *For each $m \in \mathbb{N}^+$ there is an assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ under which $|g_P^i| = m^i$ and $1 \in g_P^i$ for all i, the prime divisors of numbers in g_P^i are all in P, and $K = \{1\}$ for all $K \in Q_P$.*

Proof of Theorem 4. Due to Lemma 6 it suffices to show $\text{DE} \leq_{\text{m}} \text{BC}'(-, \cdot)$. Instead of showing this reduction directly we define an intermediate problem, the cardinality circuit problem CC given by

$\{(C, Q, s, t) \mid C = (V, E, g_C, \alpha, \beta)$ is a $\{-, \cdot\}$-circuit, $Q \subseteq V$, $s, t \in V$, and there exists an assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ under which

1. $|I(s)| = |I(t)|$
2. $1 \in I(s) \cap I(t)$
3. $I(K) = \{1\}$ for all $K \in Q$
4. $I(s)$ and $I(t)$ only contain numbers whose prime divisors are all $> 3\}$.

Moreover, define

$$\mathcal{C} = \{(C,Q,s,t) \mid C = (V,E,g_C,\alpha,\beta) \text{ is a } \{-,\cdot\}\text{-circuit, } Q \subseteq V,\ s,t \in V, \text{ for all assignments with values from } \mathcal{P}_{\text{fin}}(\mathbb{N}^+) \text{ satisfying } \forall_{K\in Q}\ K = \{1\} \text{ it holds that } s \geq t \text{ and } s \text{ and } t \text{ only contain numbers whose prime divisors are } > 3\},$$

i.e., for all circuits in $\mathcal{C}$ each relevant assignment maps s to a set with higher or equal cardinality than the set it maps t to and each relevant assignment maps s and t to sets that do not contain any numbers with prime divisors ≤ 3. For the sake of simplicity, we also call tuples (C,Q,s,t) $\{-,\cdot\}$-circuits.

The proof will be given in the two steps

1. $(\text{DE}, \overline{\text{DE}}) \leq_{\text{m}} (\text{CC}, \overline{\text{CC}} \cap \mathcal{C})$
2. $(\text{CC}, \overline{\text{CC}} \cap \mathcal{C}) \leq_{\text{m}} (\text{BC}'(-,\cdot), \overline{\text{BC}'(-,\cdot)})$.

Thus the function composition of the two reduction functions shows $\text{DE} \leq_{\text{m}} \text{BC}'(-,\cdot)$.

1. Roughly speaking, the first of the two reductions generates a circuit computing two sets whose cardinalities express the results of two multivariate polynomials.

Let q and q' be multivariate polynomials with variables $x_1,\dots,x_n$. Then for any assignment with positive natural numbers $a_1,\dots,a_n$ it holds $q(a_1,\dots,a_n) = q'(a_1,\dots,a_n)$ if and only if $q(a_1,\dots,a_n)^2 + q'(a_1,\dots,a_n)^2 = 2\cdot q(a_1,\dots,a_n)\cdot q'(a_1,\dots,a_n)$. Observe that here because of $(q(a_1,\dots,a_n) - q'(a_1,\dots,a_n))^2 \geq 0$ we have $q(a_1,\dots,a_n)^2 + q'(a_1,\dots,a_n)^2 \geq 2\cdot q(a_1,\dots,a_n)\cdot q'(a_1,\dots,a_n)$ for any assignment. Due to that we may assume that we are given multivariate polynomials q and q' with variables $x_1,\dots,x_n$ such that $q \geq q'$ for all assignments of the variables with values from $\mathbb{N}^+$. Let

$$q = \sum_{i=1}^{m} a_i \cdot \prod_{j=1}^{n} x_j^{d_{i,j}} \quad \text{and} \quad q' = \sum_{i=1}^{m'} a'_i \prod_{j=1}^{n} x_j^{d'_{i,j}}$$

for positive numbers m, m', a_i, and a'_i and natural numbers $d_{i,j}$ and $d'_{i,j}$. Moreover, for each variable x_j define $e_j = \max(\{d_{1,j},\dots,d_{m,j},d'_{1,j},\dots,d'_{m',j}\})$, i.e., e_j denotes the maximum exponent of the variable x_j occurring in a monomial of q or q'.

We now successively build the output circuit (C,Q,s,t). For the single steps we give intuition which is written italic.

1. For each variable x_j select a set $P_j = \{p_{j,1},\dots,p_{j,e_j}\}$ of primes greater than 3 such that $|P_j| = e_j$ and $P_j \cap P_{j'} = \emptyset$ for $j \neq j'$. Then insert a circuit (C_{P_j}, Q_{P_j}) according to Lemma 8 and for all P_j, insert the nodes of Q_{P_j} into Q.
   We will make use of the notation of Lemma 8, in particular of the nodes $g^0_{P_j},\dots,g^{e_j}_{P_j}$. That means, for any assignment which satisfies $K = \{1\}$ for all $K \in Q \supseteq Q_{P_j}$, it holds $|g^i_{P_j}| = m^i_j$ for $m_j \in \mathbb{N}^+$ and for all $i \leq e_j$. Moreover, in that case all primes dividing some number of $g^i_{P_j}$ are in P_j.
   For intuition, think of the node $g^i_{P_j}$ as a set whose cardinality describes x^i_j.
2. **a.** Choose a prime $p > 3$ not used before and insert gates $h_i = \{1,p,\dots,p^{a_i-1}\}\cdot\prod_{j=1}^{n} g^{d_{i,j}}_{P_j}$ for all $i = 1,\dots,m$.
   Loosely speaking, the cardinality of h_i describes the value of the i-th monomial of q.
   b. For each node h_i choose a prime $p_i > 3$ not used before and insert a node $h'_i = (\{1,p_i\}\cdot h_i) - \big(h_i - \{1\}\big)$.
   As addition is supposed to be simulated by union, we need to make sure that the sets standing for distinct monomials are disjoint. Still, for a technical reason we have to keep 1 in each set. So the idea is to let h'_i consist of 1 and a copy of h_i multiplied with an additional prime factor.

   c. For $i = 1, \dots, m$ add an unassigned input node z_q. Finally add nodes $z_q - \left(\bigcup_{i=1}^{m} h'_i - \{1\}\right)$ and $h'_i - (z_q - \{1\})$ (for $i = 1, \dots, m$) and insert these nodes into Q.
   Roughly speaking, z_q describes the value of $q + 1$ as it is the union of all the h'_i.
3. Do the same as in step 2 but for q'. In particular a node $z_{q'}$ is added.
4. Define $s = z_q$ and $t = z_{q'}$.

First, observe that the function $(q, q') \mapsto (C, Q, s, t)$ is computable. In order to show

$$(q, q') \in \mathrm{DE} \Rightarrow (C, Q, s, t) \in \mathrm{CC} \quad \text{and} \quad (q, q') \notin \mathrm{DE} \Rightarrow (C, Q, s, t) \in \overline{\mathrm{CC}} \cap \mathcal{C}$$

we make the following central observation.

▶ **Claim 9.**

1. *For each $y_1, \dots, y_n \in \mathbb{N}^+$ there is an assignment of the circuit (C, Q) with values from $\mathcal{P}_{\mathrm{fin}}(\mathbb{N}^+)$ such that s (resp., t) consists of $1 + q(y_1, \dots, y_n)$ (resp., $1 + q'(y_1 \dots, y_n)$) numbers whose prime divisors are greater than 3, $1 \in s \cap t$, and $K = \{1\}$ for all $K \in Q$.*
2. *If $K = \{1\}$ for all $K \in Q$ under some assignment with values from $\mathcal{P}_{\mathrm{fin}}(\mathbb{N}^+)$, then there are $y_1, \dots, y_n \in \mathbb{N}^+$ such that $|s| = 1 + q(y_1, \dots, y_n)$ and $|t| = 1 + q'(y_1, \dots, y_n)$ and s and t solely contain numbers whose prime divisors are all greater than 3.*

Proof of Claim 9.

1. Let $y_1, \dots, y_n \in \mathbb{N}^+$. Then according to Lemma 8 the inputs of the circuits (C_{P_j}, Q_{P_j}) can be chosen such that
   - $K = \{1\}$ for all $K \in Q_{P_j}$,
   - $|g^i_{P_j}| = y^i_j$ and $1 \in g^i_{P_j}$ for $i = 1, \dots, e_j$, and
   - all prime divisors of numbers in $g^i_{P_j}$ are in P_j and greater than 3.

   As the set of primes chosen for two different variables are disjoint and in step 2b we select primes not used before, the gate h_i associated with the monomial $a_i \cdot \prod_{j=1}^{n} x_j^{d_{i,j}}$ contains $a_i \cdot \prod_{j=1}^{n} y_j^{d_{i,j}}$ elements that only have prime divisors greater than 3. Furthermore, as $1 \in h_i$ for all i, we have $|h'_i| = 2 \cdot |h_i| - (|h_i| - 1) = |h_i| + 1$. Moreover, observe that $h'_i \cap h'_j = \{1\}$ for arbitrary $i \neq j$.
   For the node z_q choose the assignment $\bigcup_{i=1}^{m} h'_i$. Consequently, $1 \in z_q$ and

$$|z_q| = 1 + \sum_{i=1}^{m} \underbrace{(|h'_i| - 1)}_{=|h_i|} = 1 + \sum_{i=1}^{m} a_i \cdot \prod_{j=1}^{n} x_j^{d_{i,j}} = 1 + q(y_1, \dots, y_n).$$

   Since we do the same for the nodes associated with the polynomial q' we have $|z_{q'}| = 1 + q'(y_1, \dots, y_n)$ and $1 \in z_{q'}$. Observe that the prime divisors of numbers in z_q and $z_{q'}$ are greater than 3.
   It remains to observe that all nodes added into Q in step 2c compute the set $\{1\}$. This holds since z_q was chosen to be $\bigcup_{i=1}^{m} h'_i$.
2. Consider an assignment with $K = \{1\}$ for all $K \in Q$. Then according to Lemma 8 for each variable x_j we have $|g^i_{P_j}| = y^i_j$ for some $y_j \in \mathbb{N}^+$ and $i = 0, \dots, e_j$ and all numbers in these gates solely have prime divisors in P_j. As the P_j are pairwise disjoint and in step 2b we select primes not used before, we obtain $|h_i| = a_i \cdot \prod_{j=1}^{n} y_j^{d_{i,j}}$ and $|h'_i| = |h_i| + 1$. As $h'_i \cap h'_j = \{1\}$ for $i \neq j$ and each h'_i contains 1, it holds $|z_q| = 1 + \sum_{i=1}^{n} a_i \cdot \prod_{j=1}^{n} y_j^{d_{i,j}} = 1 + q(y_1, \dots, y_n)$. Similarly we obtain $|z_{q'}| = 1 + q'(y_1, \dots, y_n)$.
   It remains to argue that under the given assignment s and t do not contain any numbers with prime divisors ≤ 3. Obviously, the assigned inputs only compute sets whose elements solely have prime divisors greater than 3. By our construction and Lemma 8 the same

holds for all nodes $g^i_{P_j}$. As a consequence, all nodes h_i and h'_i have the same property and due to $z_q - \left(\bigcup_{i=1}^m h'_i - \{1\}\right) = \{1\}$ (cf. Step 2c) this also holds for $z_q = s$. An analogous argumentation shows that also t does not contain any numbers with prime divisors ≤ 3. ◀

As it has been argued above that $|q| \geq |q'|$, Claim 9 implies the following two statements: If $(q, q') \in \mathrm{DE}$, then $(C, Q, s, t) \in \mathrm{CC}$. If $(q, q') \notin \mathrm{DE}$, then $(C, Q, s, t) \in \overline{\mathrm{CC}} \cap \mathcal{C}$.

2. Now we show $(\mathrm{CC}, \overline{\mathrm{CC}} \cap \mathcal{C}) \leq_{\mathrm{m}} (\mathrm{BC}'(-,\cdot), \overline{\mathrm{BC}'(-,\cdot)})$. The following algorithm computes the reduction function. The italic comments are supposed to give some intuition.

1. Let a circuit (C, Q, s, t) be given. We construct a circuit (C', Q') by successively updating the given circuit.
2. Add new unassigned input gates X and X'. Insert the following nodes into Q':

$$\{1,2\} \cdot s - (X - \{1\}), \tag{1}$$
$$\{1,2\} \cdot t - (X - \{1\}), \tag{2}$$
$$\{1,2\} \cdot (X - s) - \big((X' \cup (X - s)) - \{1\}\big), \tag{3}$$
$$X' - \{2\} \cdot (X - s). \tag{4}$$

The basic idea is as follows: X is supposed to be an interval containing s and t and X' basically encodes the set $X - s$ where this set is made disjoint to t by multiplying it with $\{2\}$. As $|s| \geq |t|$, the set $X' \cup t$ is subbalanced. But if $|s| = |t|$, then $X' \cup t$ is almost balanced. Adding the element $\max(X')+1$ would make the set balanced. This element is generated in the next step.

3. Let $p_1 = 2$ and $p_2 = 3$. Add a circuit $(C_{\{p_1,p_2\}}, Q_{\{p_1,p_2\}})$ according to Lemma 7. Put all nodes of $Q_{\{p_1,p_2\}}$ into Q'. Add a node $g = \big(g^2_{\{p_1,p_2\}} \cdot \{1,3\}\big) - (g^2_{\{p_1,p_2\}} - \{1\})$.
4. Add a new unassigned input node O and the following nodes which are also added to Q':

$$O - \Big(\big(X' \cup t \cup g\big) - \{1\}\Big), \tag{5}$$
$$X' - (O - \{1\}), \tag{6}$$
$$t - (O - \{1\}), \tag{7}$$
$$g - (O - \{1\}). \tag{8}$$

Thus, roughly speaking, the output set O equals $X' \cup t \cup g$ and is only balanced if $|t| \geq |s|$.

5. Let O be the output node of the circuit (C', Q').

▶ **Claim 10.** *If $(C, Q, s, t) \in \mathrm{CC}$, then $(C', Q') \in \mathrm{BC}'(-,\cdot)$.*

Proof of Claim 10. Let $(C, Q, s, t) \in \mathrm{CC}$. Then there is some assignment with
- $|s| = |t|$,
- $1 \in s \cap t$,
- $K = \{1\}$ for all $K \in Q$, and
- s and t only contain numbers whose prime divisors are all greater than 3.

We now consider the circuit (C', Q') under an assignment satisfying the four conditions just mentioned. Moreover, we choose the input of $C_{\{p_1,p_2\}}$, X, X', and O such that
- $g = \{1, 3^m\}$ for m minimal with $4 \cdot (\max(s \cup t) + 1) < 3^m$ and $4 \mid 3^m - 1$ and all nodes in $Q_{\{p_1,p_2\}}$ compute $\{1\}$ (such an assignment exists by Lemma 7),
- $X = \{x \mid 1 \leq x \leq (3^m - 1)/2\}$,
- $X' = \{1\} \cup \{2\} \cdot (X - s)$, and
- $O = X' \cup t \cup g = \big(\big(\{2\} \cdot (X - s)\big)\big) \cup t \cup \{3^m\}$.

In order to see $K = \{1\}$ for all $K \in Q'$ it remains to consider the nodes added in the steps 2 and 4. Due to the choice of g and X it holds $\max(X) > 2{\cdot}\max(s \cup t)$ and thus the nodes defined in (1) and (2) compute $\{1\}$. The choice of X' immediately implies that the node defined in (4) computes $\{1\}$. Now we argue for the node defined in (3): As $X' = \{1\} \cup \{2\} \cdot (X - s)$ we have $\{1,2\} \cdot (X - s) - \big((X' \cup (X - s)) - \{1\}\big) = \{1,2\} \cdot (X - s) - \big((\{1,2\} \cdot (X - s)) - \{1\}\big) = \{1\}$. The nodes defined in (5), (6), (7), and (8) compute $\{1\}$ by the choice of g, X, X', and O.

As s and t only contain numbers whose prime divisors are > 3, the sets $\{2\} \cdot (X - s)$, t, and $\{3^m\}$ are disjoint. Hence, $|O| = \max(X) - |s| + |t| + 1 = \max(X) + 1 = \frac{\max(O)-1}{2} + 1 = \frac{\max(O)+1}{2}$ and thus O is balanced. ◀

▶ **Claim 11.** *If* $(C, Q, s, t) \in \overline{\mathrm{CC}} \cap \mathcal{C}$*, then* $(C', Q') \in \overline{\mathrm{BC}'(-,\cdot)}$.

Proof of Claim 11. For a contradiction, assume that $(C, Q, s, t) \in \overline{\mathrm{CC}} \cap \mathcal{C}$ and $(C', Q') \in \mathrm{BC}'(-,\cdot)$. As the second circuit is an extended version of the first circuit, both circuits can now be considered under the same assignment. Choose an assignment with values from $\mathcal{P}_{\mathrm{fin}}(\mathbb{N}^+)$ under which O is balanced and all $K \in Q'$ satisfy $K = \{1\}$. As by construction $Q \subseteq Q'$, we have $K = \{1\}$ for $K \in Q$.

As in particular the nodes defined in (1) and (2) compute $\{1\}$, we obtain $1 \in s \cap t$, $X \supseteq \{1,2\} \cdot s \cup \{1,2\} \cdot t$, and in particular $s \subseteq X$ and $\max(X) > \max(s) \geq 1$. As $\{1,2\} \cdot (X - s) - \big((X' \cup (X - s)) - \{1\}\big) = \{1\}$ (cf. (3)), it holds $2 \cdot \max(X) \in X'$. Since the node defined in (4) computes $\{1\}$, we obtain $X' \subseteq \{1\} \cup \{2\} \cdot (X - s)$. In particular, $\max(X') = 2 \cdot \max(X)$.

The fact that the nodes defined in (5), (6), (7), and (8) compute $\{1\}$ implies $1 \in O \cap X' \cap t \cap g$ and $O = X' \cup t \cup g$. Moreover, it follows from Lemma 7 that $g = \{1, 3^m\}$ for some $m \in \mathbb{N}^+$. Thus, as $1 \in t$,

$$O \subseteq \{1\} \cup \big(\{2\} \cdot (X - s)\big) \cup t \cup g = \big(\{2\} \cdot (X - s)\big) \cup t \cup \{3^m\}. \tag{9}$$

As O is balanced, $\max(O)$ is odd by Lemma 1. Since $X \supseteq t$ and $\max(X') = 2 \cdot \max(X)$ is even, $\max(O) = 3^m > \max(X')$. Due to $(C, Q, s, t) \in \mathcal{C}$, under the given assignment $|s| \geq |t|$ and s and t do not contain any numbers with prime divisors ≤ 3. Due to that, since we have seen $1 \in s \cap t$, and as by assumption $(C, Q, s, t) \notin \mathrm{CC}$, it even holds $|s| > |t|$.

Putting things together, as we have proven (9), $|s| > |t|$, $1 \in t$, $s \subseteq X$, $\max(X') = 2 \cdot \max(X)$, and $\max(O) > \max(X')$, we now obtain $|O| \leq \max(X) - |s| + |t| + 1 < \max(X) + 1 = \frac{\max(X')+2}{2} \leq \frac{\max(O)+1}{2}$, which contradicts the fact that O is balanced. ◀

This completes the proof of $(\mathrm{CC}, \overline{\mathrm{CC}} \cap \mathcal{C}) \leq_{\mathrm{m}} (\mathrm{BC}'(-,\cdot), \overline{\mathrm{BC}'(-,\cdot)})$ and thus $\mathrm{BC}'(-,\cdot)$ and $\mathrm{BC}(-,\cdot)$ are $\leq_{\mathrm{m}}$-complete for RE. ◀

4 Smaller Sets of Operations Lead to Problems in NP

In this section it is shown that all problems $\mathrm{BC}(\mathcal{O})$ for $\mathcal{O} \subsetneq \{-,\cdot\}$ are in NP. Each of these problems is proven to be $\leq_{\mathrm{m}}^{\log}$-complete for one of the classes L, NL, and NP.

4.1 The Complexity of the Problem Solely Admitting Multiplication

This section's purpose is to argue for the NL-completeness of $\mathrm{BC}(\cdot)$. Special cases of strong results from the literature [7, 8, 13] essentially yield the following theorem.

▶ **Theorem 12.** *There exists* $\mu \in \mathbb{N}$ *such that for all non-empty sets* $A, B \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ *with* $\max(A) \geq \mu$ *and* $\max(B) \geq \mu$ *the set* $A \cdot B$ *is subbalanced.*

▶ **Theorem 13.** BC(·) *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* NL.

Proof. In the following we present an NL-algorithm for BC(·). We make use of the fact that the graph accessibility problem for directed graphs and the modifications of this problem

$$\mathrm{GAP}_{\geq k} = \{(G,s,t) \mid G \text{ is an directed graph, there exist } k \text{ paths from } s \text{ to } t\}$$

and consequently

$$\mathrm{GAP}_{=k} = \{(G,s,t) \mid G \text{ is an directed graph, the number of paths from } s \text{ to } t \text{ is } k\}$$

for $k \in \mathbb{N}^+$ are in NL. We may assume the following for the input circuit C:

1. All gates in C are connected to the output gate g_C. Otherwise, delete all edges not connected to the output, which can be done by an NL-subroutine.
2. No assigned input computes the empty set or the set $\{0\}$. Otherwise, under the assumption of 1 we may reject immediately.
3. There is an assigned input gate a computing a set with maximum ≥ 2. Otherwise: under the assumption of 1 and 2,
   - we may accept if there is an unassigned input or no assigned input computes $\{0,1\}$
   - we may reject if there does not exist an unassigned input and there is an assigned input computing $\{0,1\}$.
4. No assigned input gate but possibly a computes a set containing 0. Otherwise, under the assumptions 1 and 3 we may delete 0 from all assigned inputs and insert 0 into a.
5. There is an assigned input node g_1 computing $\{1\}$.
6. For each set $M \subseteq \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ there is at most one assigned input computing M. Otherwise, select one of the nodes computing M, let all outgoing edges of nodes computing M start in this node, and delete all other nodes computing M and their incident edges.

Assume there is an NL-algorithm P that accepts the set of those circuits C which satisfy the mentioned properties and whose unassigned inputs can be assigned with sets of *positive* naturals such that the output set is balanced. Then the following NL-algorithm accepts BC(·) (on input of a circuit C satisfying the properties listed above).

- If P accepts on C, accept.
- If there is an unassigned input, then add 0 into the set computed by the aforementioned node a and accept if P accepts the modified circuit.
- Reject.

Now we sketch P and argue that it is an NL-algorithm. Let $\mu \geq 2$ be the number mentioned in Theorem 12, i.e., for $A, B \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ with $\max(A) \geq \mu \leq \max(B)$ the set $A \cdot B$ is subbalanced. The algorithm will query the following constant-size problem

$$\Theta = \{(B,k_1,k_2) \mid B \subseteq \{(h,i_h) \mid h \subseteq \{0,1,\ldots,\mu\}, 1 \leq i_h \leq 2\}, |B| \leq \mu, k_1 \leq \mu, k_2 \leq \mu,$$
$$\exists_{E_1,\ldots,E_{k_1},F_1,\ldots,F_{k_2} \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N}^+)} \prod_{(h,i_h)\in B} h^{i_h} \cdot \prod_{i=1}^{k_1} E_i \cdot \prod_{i=1}^{k_2} F_i^2 \text{ is balanced}\}.$$

1. If there are two assigned input gates each containing an element $\geq \mu$, reject.
   If there is an assigned input gate with two paths to g_C containing an element $\geq \mu$, reject.
2. If there are at least μ assigned input gates computing a set with maximum ≥ 2, reject.
3. In case there is an assigned input gate computing a set with maximum ≥ 2 with at least three paths to the output, reject.
4. Let $v_1,\ldots,v_n$ be the nodes of the circuit in topological order. For $i = 1,\ldots,n$, if one of the conditions

- v_i is an unassigned input with at least three paths to g_C.
- v_i is an unassigned input with precisely one path to g_C, such that there are at least μ unassigned inputs $< v_i$ with precisely one path to g_C.
- v_i is an unassigned input with precisely two paths to g_C, such that there are at least μ unassigned inputs $< v_i$ with precisely two paths to g_C.
- g_1 is the only input with a path to v_i.

is satisfied, then delete v_i and let all outgoing edges of v_i start in g_1.
This step can be implemented as a non-deterministic logarithmic-space subroutine.

5. Let n_1 (resp., n_2) be the number of unassigned inputs with 1 path (resp., 2 paths) to g_C. Due to Step 4 we have $\max(n_1, n_2) \leq \mu$. Moreover, let A be a set consisting of all pairs (h, i_h) where h is a set computed by an assigned input with $1 < \max(h) \leq \mu$ and $i_h \in \{1, 2\}$ is the number of paths from h to g_C. Due to Step 2 it holds $|A| \leq \mu$. We have the following cases.

a. In case there is no assigned input gate with an element $\geq \mu$:
If $(A, n_1, n_2) \in \Theta$, then accept. Otherwise reject.
Computing the triple (A, n_1, n_2) is possible in non-deterministic logarithmic space whereas the subsequent test only requires constant time.

b. In case there is one assigned input gate g with an element $\geq \mu$:
Due to Step 1 the node g only has one path to the output.

i. For all $E_1, \ldots, E_{n_1}, F_1, \ldots, F_{n_2} \in \mathcal{P}(\{1, \ldots, \mu\})$ do the following

- Compute the constant-size set

$$M = \prod_{i=1}^{n_1} E_i \cdot \prod_{i=1}^{n_2} F_i^2 \cdot \prod_{(h,i_h)\in A, h\neq g} h^{i_h}.$$

- Test whether $g \in \mathrm{Bal}_M$ and accept in case the answer is "yes".

ii. Reject.

By Proposition 2 this step can be executed in logarithmic space.

In the following we observe that each step of the algorithm P accepts (resp., rejects) if and only if the circuit at the beginning of the execution of the respective step has a (resp., no) balancing assignment with values from $\mathcal{P}_{\mathrm{fin}}(\mathbb{N}^+)$. It suffices to argue for the following steps.

1. If the algorithm rejects in this step, then there are sets A and B with $\max(A) \geq \mu \leq \max(B)$ and a set M such that $g_C = A \cdot B \cdot M$. Then according to Theorem 12 it holds $|(A \cup \{0\}) \cdot B| \leq \max(A) \cdot \max(B)/2$. Hence for each set $M \in \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ the set $M \cdot A \cdot B \subseteq (A \cup \{0\}) \cdot B \cdot (M - \{0\})$ contains at most $\max(A) \cdot \max(B) \cdot \max(M)/2$ elements and its greatest element is $\max(A) \cdot \max(B) \cdot \max(M)$. Thus, the set is subbalanced.

2. If there are $\geq \mu$ sets with maximum greater 2 connected to the output, then we can interpret these sets as two sets with maxima $\geq \mu$ and argue in the same way as in the step before.

3. If the algorithm rejects in this step, then there are sets A and M with $\max(A) \geq 2$ and $g_C = A \cdot A \cdot A \cdot M$. If $\max(A) = 2$, then Lemma 1 states that the output set is not balanced. Otherwise, $\max(A) \geq 3$ and according to Statement 2 of Lemma 6 the set $A \cdot A \cdot A$ contains less than $\max(A)^3/2$ elements. Hence g_C contains less than $\max(A)^3 \cdot \max(M)/2$ elements and the maximum of this set is $\max(A)^3 \cdot \max(M)$. Thus g_C is subbalanced.

5. At the beginning of the execution of this step we have the following situation: Due to the steps 1, 2, and 3 and because of the assumption we made on the input circuit there

- is at most one assigned input containing an element $> \mu$ and this has at most one path to the output gate.
- are at most μ assigned inputs with maximum ≥ 2 and all these inputs have at most two paths to the output gate.
- is one assigned input with maximum < 2, namely $g_1 = \{1\}$.

Moreover, as observed above, because of Step 4 it holds $\max(n_1, n_2) \leq \mu$ and there are no unassigned inputs with more than 2 paths to the output.

Thus we have to consider two cases. Either there is no assigned input with maximum $> \mu$ or there is one. In the first case the circuit has a balancing assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ if and only if there are $n_1 + n_2$ sets $E_1, \ldots, E_{n_1}, F_1, \ldots, F_{n_2} \in \mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ such that $\prod_{(h,i_h)\in B} h^{i_h} \cdot \prod_{i=1}^{k_1} E_i \cdot \prod_{i=1}^{k_2} F_i^2$ is balanced. This is what the algorithm tests. In the second case, assigning one of the unassigned inputs with a set with maximum $> \mu$ would lead to a subbalanced output with the same argument as was used for Step 1. Thus, only assignments with values from $\mathcal{P}(\{1, \ldots, \mu\})$ have to be considered. Hence, there is a balancing assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ if and only if there are sets $E_1, \ldots, E_{n_1}, F_1, \ldots, F_{n_2} \in \mathcal{P}(\{1, \ldots, \mu\})$ such that $\prod_{i=1}^{n_1} E_i \cdot \prod_{i=1}^{n_2} F_i^2 \cdot \left(\prod_{(h,i_h)\in A, h\neq g} h^{i_h}\right) \cdot g$ is balanced. This is what the algorithm tests.

It remains to observe that the circuit has a balancing assignment with values from $\mathcal{P}_{\text{fin}}(\mathbb{N}^+)$ before the execution of Step 4 if and only if it has afterwards:
In case there are more than μ unassigned inputs with one path (resp., two paths) to the output and more than μ of them are mapped to sets containing elements ≥ 2, then the same arguments as for Step 2 yield that the output is subbalanced. Therefore, all but μ of these nodes can be replaced with g_1.
Let g be an unassigned input with at least three paths to the output (if such a node exists). Assigning this node with a set with maximum ≥ 2 leads to a subbalanced output set with the same arguments as were used for Step 3. Therefore, g can be replaced with g_1.
For each node v_i there exists an input that has a path to v_i. Hence, if no input different from g_1 has a path to v_i, then v_i computes $\{1\}$ and can be replaced with g_1.

By a straightforward reduction from a problem investigated by McKenzie and Wagner [15] one receives the NL-hardness of BC(·). ◀

4.2 The Complexity of the Problems Not Admitting Multiplication

We consider the two remaining problems and prove that BC(−) is $\leq_{\text{m}}^{\log}$-complete for NP and BC(∅) is in L. The NP-hardness of BC(−) can be obtained by a straightforward reduction from CSAT. Hence, for the following theorem it suffices to argue for the membership in NP.

▶ **Theorem 14.** BC(−) *is* $\leq_{\text{m}}^{\log}$*-complete for* NP.

Proof. We sketch an NP-algorithm that accepts BC(−).

1. Input: a circuit C with output node g_C and labeling function α.
2. Go from g_C upwards always taking the left predecessor. Denote the input gate finally reached by g.
3. If g is assigned, then: guess an assignment with values from $\mathcal{P}(\alpha(g))$ and accept if the output set is balanced for this assignment, otherwise reject.
4. Here g is unassigned. Let M be the union of all sets computed by assigned inputs. Let $m = \max(M) + 1$. Guess an assignment such that $I(g) = \{m\}$ and each unassigned input either computes $\{m\}$ or $\emptyset$. If under this assignment g_C contains m, then accept.

5. Guess an assignment of the unassigned inputs such that each of them computes a subset of M. In case g_C is balanced, accept. Otherwise reject.

If the algorithm accepts, then $C \in \mathbf{BC}(-)$: It suffices to consider the 4-th step. If the algorithm accepts in this step, then there is an assignment that maps each unassigned input either to $\{m\}$ or to $\emptyset$ such that m is in the output set. Now change this assignment such that the sets mapped to $\{m\}$ are now mapped to $\{m+1, m+2, \ldots, 2m+1\}$. Then $I(C) = \{m+1, m+2, \ldots, 2m+1\}$ is balanced and $C \in \mathrm{BC}(-)$. Trivially, in case C is accepted in the 5-th step, $C \in \mathrm{BC}(-)$.

If the algorithm rejects, then $C \notin \mathbf{BC}(-)$: If the algorithm rejects, then this happens in step 3 or step 5. We argue for the first case. Here g is an assigned input gate. As the output set is a subset of $\alpha(g)$, it holds $g_c \subseteq \alpha(g)$ for any assignment and hence it suffices to consider assignments that map all unassigned inputs to subsets of $\alpha(g)$. As the algorithm rejects, g_C is not balanced under any of these assignments and thus $C \notin \mathrm{BC}(-)$.

It remains to argue for the case where the algorithm rejects in step 5. In this case, g is an unassigned input and as step 4 did not accept, there is no assignment putting elements outside of M into the circuit's output set. Hence, it is sufficient to consider assignments that solely map to subsets of M. As the algorithm rejects, none of these assignments yields a balanced output set and hence there is no assignment at all under which the output set is balanced. Therefore, $C \notin \mathrm{BC}(-)$. ◀

The following theorem basically follows from Proposition 2.

▶ **Theorem 15.** $\mathrm{BC}(\emptyset) \in \mathrm{L}$.

5 Conclusion and Open Questions

The following table summarizes our results, namely the lower and upper complexity bounds for the complexity of $\mathrm{BC}(\mathcal{O})$ with $\mathcal{O} \subseteq \{-, \cdot\}$.

$\mathrm{BC}(\mathcal{O})$ for $\mathcal{O} =$	$\leq_{\mathrm{m}}^{\log}$-hard for	contained in
$\emptyset$	L	L, Theorem 15
$\{-\}$	NP, Theorem 14	NP, Theorem 14
$\{\cdot\}$	NL, Theorem 13	NL, Theorem 13
$\{\cdot, -\}$	undecidable, Theorem 4	

To our knowledge, in contrast to all results from previous papers on complexty issues concerning decision problems for integer circuits (e.g., [15, 19, 2, 9, 11, 1]) or related constraint satisfaction problems ([10, 5]), a problem *admitting only one arithmetic operation* is shown to be undecidable. Beginning with this problem, namely $\mathrm{BC}(-, \cdot)$, the problems $\mathrm{BC}(\mathcal{O})$ for $\mathcal{O} \subseteq \{-, \cdot\}$ are systematically investigated and for each of these problems the complexity is precisely characterized. It turns out that decreasing the size of the set of allowed operations yields problems that are in NP. In particular, all these problems are $\leq_{\mathrm{m}}^{\log}$-complete for one of the classes L, NL, and NP.

Hence, in some sense the questions of this paper are completely answered. Nevertheless, there arise new questions from our results: Is there a set $\mathcal{O} \subseteq \{-, \cup, \cap\}$ such that $\mathrm{BC}(\mathcal{O} \cup \{+\})$ is undecidable? And if so, for which of the sets this is the case and for which it is not?

References

1 D. Barth, M. Beck, T. Dose, C. Glaßer, L. Michler, and M. Technau. Emptiness problems for integer circuits. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 33:1–33:14, 2017. doi:10.4230/LIPIcs.MFCS.2017.33.

2 H.-G. Breunig. The complexity of membership problems for circuits over sets of positive numbers. In *Fundamentals of Computation Theory, 16th International Symposium, FCT 2007, Budapest, Hungary, August 27-30, 2007, Proceedings*, pages 125–136, 2007. doi:10.1007/978-3-540-74240-1_12.

3 S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158, 1971. doi:10.1145/800157.805047.

4 M. Davis, H. Putnam, and J. Robinson. The decision problem for exponential Diophantine equations. *Annals of Mathematics*, 74(2):425–436, 1961.

5 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*, pages 32:1–32:13, 2016. doi:10.4230/LIPIcs.MFCS.2016.32.

6 T. Dose. Balance problems for integer circuits. Technical Report 18-055, Electronic Colloquium on Computational Complexity (ECCC), 2018. URL: https://eccc.weizmann.ac.il/report/2018/055.

7 K. Ford. integers with a divisor in $(y, 2y]$. In *Anatomy of integers*, volume 46 of *CRM Proc. and Lect. Notes*, pages 65–81. Amer. Math. Soc., Providence, RI, 2008.

8 K. Ford. the distribution of integers with a divisor in a given interval. *Annals of Math. (2)*, 168:367–433, 2008.

9 C. Glaßer, K. Herr, C. Reitwießner, S. D. Travers, and M. Waldherr. Equivalence problems for circuits over sets of natural numbers. *Theory Comput. Syst.*, 46(1):80–103, 2010. doi:10.1007/s00224-008-9144-8.

10 C. Glaßer, P. Jonsson, and B. Martin. Circuit satisfiability and constraint satisfaction around skolem arithmetic. *Theor. Comput. Sci.*, 703:18–36, 2017. doi:10.1016/j.tcs.2017.08.025.

11 C. Glaßer, C. Reitwießner, S. D. Travers, and M. Waldherr. Satisfiability of algebraic circuits over sets of natural numbers. *Discrete Applied Mathematics*, 158(13):1394–1403, 2010. doi:10.1016/j.dam.2010.04.001.

12 Thomas Gundermann, Nasser Ali Nasser, and Gerd Wechsung. A survey on counting classes. In *Proceedings: Fifth Annual Structure in Complexity Theory Conference, Universitat Politècnica de Catalunya, Barcelona, Spain, July 8-11, 1990*, pages 140–153, 1990. doi:10.1109/SCT.1990.113963.

13 D. Koukoulopoulos. On the number of integers in a generalized multiplication table. *Journal für die reine und angewandte Mathematik*, 689:33–99, 2014. doi:10.1515/crelle-2012-0064.

14 Y. V. Matiyasevich. Enumerable sets are Diophantine. *Doklady Akad. Nauk SSSR*, 191:279–282, 1970. Translation in Soviet Math. Doklady, 11:354–357, 1970.

15 P. McKenzie and K. W. Wagner. The complexity of membership problems for circuits over sets of natural numbers. *Computational Complexity*, 16(3):211–244, 2007. doi:10.1007/s00037-007-0229-6.

16 C. M. Papadimitriou. *Computational complexity.* Addison-Wesley, Reading, Massachusetts, 1994.

17 I. Pratt-Hartmann and I. Düntsch. Functions definable by arithmetic circuits. In *Mathematical Theory and Computational Practice, 5th Conference on Computability in Europe, CiE 2009, Heidelberg, Germany, July 19-24, 2009. Proceedings*, pages 409–418, 2009. doi:10.1007/978-3-642-03073-4_42.

18 L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: Preliminary report. In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA*, pages 1–9, 1973. doi:10.1145/800125.804029.

19 S. D. Travers. The complexity of membership problems for circuits over sets of integers. *Theor. Comput. Sci.*, 369(1-3):211–229, 2006. doi:10.1016/j.tcs.2006.08.017.

20 K. W. Wagner. The complexity of problems concerning graphs with regularities (extended abstract). In *Mathematical Foundations of Computer Science 1984, Praha, Czechoslovakia, September 3-7, 1984, Proceedings*, pages 544–552, 1984. doi:10.1007/BFb0030338.

21 K. Yang. Integer circuit evaluation is pspace-complete. *J. Comput. Syst. Sci.*, 63(2):288–303, 2001. doi:10.1006/jcss.2001.1768.