Equivalence Problems for Circuits over Sets of Natural Numbers

Christian Glaßer, Katrin Herr, Christian Reitwießner,
Stephen Travers, and Matthias Waldherr

Universität Würzburg, Theoretische Informatik, Germany.

Abstract. We investigate the complexity of *equivalence problems* for $\{\cup, \cap, ^-, +, \times\}$-circuits computing sets of natural numbers. These problems were first introduced by Stockmeyer and Meyer (1973). We continue this line of research and give a systematic characterization of the complexity of equivalence problems over sets of natural numbers. Our work shows that equivalence problems capture a wide range of complexity classes like NL, $\mathrm{C}_=\mathrm{L}$, P, Π_2^P, PSPACE, NEXP, and beyond. McKenzie and Wagner (2003) studied related *membership problems* for circuits over sets of natural numbers. Our results also have consequences for these membership problems: We provide improved upper bounds for the cases of $\{\cup, \cap, ^-, +, \times\}$- and $\{\cap, +, \times\}$-circuits.

Classification: Computational and structural complexity; Combinational Circuits; Algorithms

1 Introduction

In 1973, Stockmeyer and Meyer [SM73] defined and investigated equivalence problems for *integer expressions*. They considered expressions that can be built up from single natural numbers by using Boolean operations ($^-$, $\cup$, $\cap$), addition ($+$), and multiplication ($\times$).

The *equivalence problem for integer expressions* is the question of whether two given such expressions describe the same set of natural numbers. Restricting the set of allowed operations results in equivalence problems of different complexities. Stockmeyer and Meyer [SM73] showed that the equivalence test for expressions over $\{^-, \cup, \cap, +\}$ is PSPACE-complete, and that this problem becomes Π_2^P-complete if one restricts to operations from $\{\cup, +\}$. We continue these investigations and study equivalence problems over natural numbers in a systematic way.

Despite of their simple definition, integer expressions are powerful enough to describe highly non-trivial sets. For instance, the set of primes can be described as

$$\textsc{Primes} = \overline{\overline{0 \cup 1} \times \overline{0 \cup 1}} \;\cap\; \overline{0 \cup 1}.$$

This can easily be verified: The complement of $\{0, 1\}$ multiplied with itself yields all composite numbers. Taking its complement gives the set consisting of 0, 1, and all primes. The intersection with $\overline{0 \cup 1}$ yields the set of primes. Expressions like this illustrate that equivalence problems for integer expressions comprise some of the most famous, unsolved problems in mathematics.

In 1742, Christian Goldbach stated his famous conjecture as a footnote in a letter to Leonhard Euler: "*At least it seems that every number greater than* 1 *is a sum of three prime numbers.*"[1] Euler answered with an equivalent version of this conjecture which nowadays we call the Goldbach conjecture.

Goldbach Conjecture: Every even integer ≥ 4 is the sum of two primes.

The following integer expression describes exactly the set of integers that are counterexamples for the Goldbach conjecture.

$$\textsc{CounterExamples} = (2 \times \overline{0 \cup 1}) \cap \overline{\textsc{Primes} + \textsc{Primes}}$$

The left set of the intersection is the set of even integers greater than or equal to 4, while the right set consists of those integers that are not a sum of two primes. The Goldbach conjecture is true if and only if the set of counter examples is empty. Therefore,

$$\textit{Goldbach conjecture holds} \iff \textsc{CounterExamples} \textit{ is equivalent to } 0 \cap \overline{0}.$$

So we have seen that the Goldbach conjecture can be formulated as an equivalence problem for integer expressions. Beside the original conjecture, the following variants and weakenings are studied in the literature:

Odd Goldbach Conjecture: Every odd integer ≥ 9 is the sum of three odd primes.

Weak Goldbach Conjecture: Every odd integer ≥ 7 is the sum of three primes.

Levy's Conjecture: Every odd integer ≥ 7 is the sum of a prime plus twice a prime. [Lev63]

Chen's version of the Goldbach Conjecture: Every even integer is the sum of a prime plus a number with at most two prime factors. [Che66,Che73,Che78]

It is an easy exercise to verify that all these variants can be formulated as equivalence problem for integer expressions. For this, note that the set of numbers having at most two prime factors can be described by the expression $1 \cup \textsc{Primes} \cup (\textsc{Primes} \times \textsc{Primes})$.

Regarding the Goldbach conjecture and their variants, several partial results are known: Ramaré [Ram95] proved that every even integer is the sum of at most six primes. Vinogradov [Vin37] based on work of Hardy and Littlewood [HL23] showed that the weak Goldbach conjecture holds for sufficiently large integers. Moreover, Chen [Che66,Che73,Che78] showed the same for his version of the Goldbach conjecture. Despite of these partial results, all conjectures defined so far are still open in the general case. So already at this point, the expressiveness of integer expressions makes us aware of the possibility that the general equivalence problem might be undecidable. Indeed, the decidability of the general equivalence problem will be one of our open questions.

Stockmeyer and Meyer's [SM73] motivation for the study of equivalence problems for integer expressions originated from equivalence problems for Kleene's regular expressions

[1] Note that at that time, 1 was considered to be a prime.

[MS72]. Since then, several variants and generalizations of integer expressions have been studied. Beside integer expressions (which we call integer formulas) researchers were also interested in *integer circuits* which were introduced by Wagner [Wag84]. The latter represent expressions in a succinct way and so yield problems of higher complexity.

Wagner [Wag84], Yang [Yan00], and McKenzie and Wagner [MW03] studied the complexity of *membership problems* for formulas and circuits over natural numbers: Here, for a given circuit C and a number n, one has to decide whether n belongs to the set that is described by C. Breunig [Bre03] studied membership problems for formulas and circuits over $\mathbb{N}^+$, the positive integers, while Travers [Tra04] studied the variant for $\mathbb{Z}$, the integers.

In this paper, we study equivalence problems for formulas and circuits over natural numbers. In particular, this contains the equivalence problems for formulas that Stockmeyer and Meyer [SM73] were interested in. For most of these equivalence problems we can precisely characterize their complexity.

It turns out that our results also have consequences for the known results about membership problems. In fact, our upper bounds for equivalence problems for $\{\cap, +, \times\}$-circuits and $\{\cup, \cap, ^-, +, \times\}$-circuits yield improved upper bounds for the membership problems for $\{\cap, +, \times\}$- and $\{\cup, \cap, ^-, +, \times\}$-circuits. In the latter case, this is the first nontrivial upper bound for $\mathrm{MC}_{\mathbb{N}}(\cup, \cap, ^-, +, \times)$, the most general membership problem.

Our main open question is whether the unrestricted version of the equivalence problem, $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, ^-, +, \times)$, is decidable or not. While we can show that this problem is equivalent to the corresponding membership problem, the upper bound we provide is not a decidable upper bound. So if one proves that $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, ^-, +, \times)$ is undecidable, then it follows that $\mathrm{MC}_{\mathbb{N}}(\cup, \cap, ^-, +, \times)$ also is undecidable.

A summary of the obtained results and a discussion of open problems can be found in the conclusions section.

2 Preliminaries

We fix the alphabet $\Sigma = \{0, 1\}$. Σ^* is the set of words, and $|w|$ is the length of a word $w \in \Sigma^*$. $\mathbb{N}$ denotes the set of the natural numbers, which include zero, whereas $\mathbb{N}^+$ denotes $\mathbb{N} - \{0\}$. For $a, b \in \mathbb{N}$ we define $[a, b] \overset{df}{=} \{a, a+1, \ldots, b-1, b\}$ if $a \leq b$ and $[a, b] \overset{df}{=} \emptyset$ otherwise.

The binary representation of a natural number n is denoted by $\mathrm{bin}(n)$ and this word is identified with the number itself.

We extend the arithmetical operations $+$ and $\cdot$ to subsets of $\mathbb{N}$: Let $M, N \subseteq \mathbb{N}$. We define the sum of M and N as $M + N \overset{df}{=} \{m + n : m \in M \text{ and } n \in N\}$. We define the product of M and N as $M \times N \overset{df}{=} \{m \cdot n : m \in M \text{ and } n \in N\}$.

In some cases we will identify the singleton $\{a\}$ with a. Unless otherwise stated, the domain of a variable is $\mathbb{N}$.

Furthermore, we need the function class #L and the complexity class $\mathrm{C_{=}L}$. For a nondeterministic logarithmic space machine M, define $\mathrm{acc}_M(x)$ as the number of accepting paths of M on input x. The class #L consists of precisely these functions. A set A is in $\mathrm{C_{=}L}$ if there exist $f, g \in \#\mathrm{L}$ such that $x \in A \Leftrightarrow f(x) = g(x)$ for every $x \in \Sigma^*$. See [All97] for a survey on these counting classes.

For a complexity class $\mathcal{C}$, let $\exists^{\mathrm{p}}\cdot\mathcal{C}$ denote the class of languages L such that there exists a polynomial p and a $B \in \mathcal{C}$ such that for all x,

$$x \in L \iff \exists y \text{ such that } |y| \leq p(|x|) \text{ and } (x, y) \in B.$$

Let $\mathcal{C}$ and $\mathcal{D}$ be complexity classes. We define

$$\mathcal{C} \vee \mathcal{D} \overset{\mathit{df}}{=} \{A \cup B \,\big|\, A \in \mathcal{C}, B \in \mathcal{D}\}.$$

The symmetric difference of sets A and B is defined as $A \triangle B = (A - B) \cup (B - A)$. The complex version is defined as

$$\mathcal{C} \oplus \mathcal{D} = \{A \triangle B : A \in \mathcal{C}, B \in \mathcal{D}\}.$$

For a class of languages $\mathcal{C}$ which is closed under union and intersection, the Boolean hierarchy over $\mathcal{C}$ [WW85] is the family of classes $\mathcal{C}(k)$ and $\mathrm{co}\mathcal{C}(k)$ where $k \geq 1$,

$$\begin{aligned} \mathcal{C}(k) &\overset{\mathit{df}}{=} \overbrace{\mathcal{C} \oplus \mathcal{C} \oplus \cdots \oplus \mathcal{C}}^{k \text{ times}}, \text{ and} \\ \mathrm{co}\mathcal{C}(k) &\overset{\mathit{df}}{=} \left\{\overline{L} : L \in \mathcal{C}(k)\right\}. \end{aligned}$$

Unless stated otherwise, all hardness- and completeness-results are in terms of logspace many-one reducibility.

The class DLOGCFL was introduced by McKenzie and Wagner [MW03]. It is the deterministic restriction of the class LOGCFL [Sud78]. A language L belongs to DLOGCFL if it can be accepted by a deterministic, logarithmic space-bounded Turing machine M that has also access to a pushdown store that is not subject to the logarithmic space-bound. Furthermore, M must have polynomial running time.

3 Decision Problems for Circuits over Sets of Natural Numbers

We define *circuits over sets of natural numbers* and related decision problems.

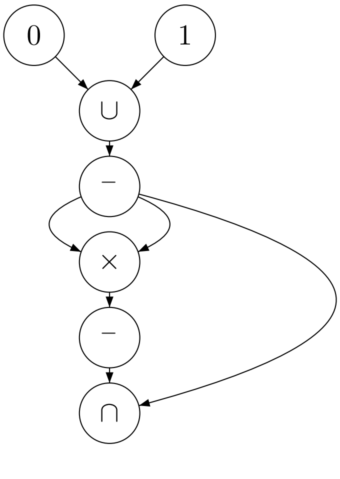
A *circuit* $C = (V, E, g_C)$ is a finite, non-empty, directed, acyclic graph (V, E) with a specified node $g_C \in V$. We remark that the graph can contain multi-edges, that it does not have to be connected, and that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. Moreover, the nodes in the graph (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. The nodes in V are also called *gates*. Nodes with indegree 0 are called *input gates* and g_C is called *output gate*. If in a circuit there is an edge going from gate u to gate v, then we say that u is a *direct predecessor* of v and v is the *direct successor* of u. If there is a path from u to v but u is not a direct predecessor of v, then u is an *indirect predecessor* of v and v is an *indirect successor* of u.

Let $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, \times\}$. An *$\mathcal{O}$-circuit* $C = (V, E, g_c, \alpha)$ is a circuit (V, E, g_c) with an attached labeling function $\alpha : V \to \mathcal{O} \cup \mathbb{N}$ such that the following holds: Each gate has an indegree in $\{0, 1, 2\}$, gates with indegree 0 have labels from $\mathbb{N}$, gates with indegree 1 have labels $^-$, and gates with indegree 2 have labels from $\{\cup, \cap, +, \times\}$. An *$\mathcal{O}$-formula* is an $\mathcal{O}$-circuit that only contains nodes with outdegree ≤ 1. For each of its gates g, the $\mathcal{O}$-circuit $C = (V, E, g_c, \alpha)$ computes a set $I(g) \subseteq \mathbb{N}$ as follows:

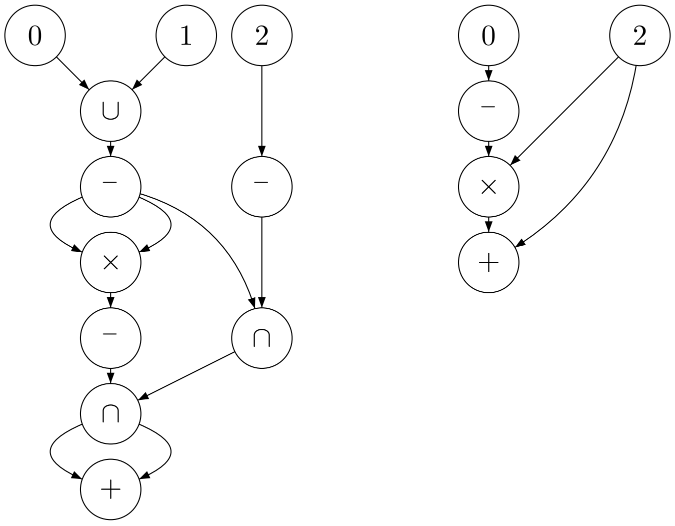
If g is an input gate, then $I(g) \overset{df}{=} \alpha(g)$. If g has label $^-$ and direct predecessor g_1, then $I(g) \overset{df}{=} \mathbb{N} - I(g_1)$. If g has label $\circ \in \{\cup, \cap, +, \times\}$ and direct predecessors g_1 and g_2, then $I(g) \overset{df}{=} I(g_1) \circ I(g_2)$.

The *set computed by* C is $I(C) = I(g_C)$; for simplification we will identify C and $I(C)$. For an $\mathcal{O}$-circuit $C = (V, E, g_c, \alpha)$ that has exactly n input gates $v_1, \ldots, v_n$ where $v_1 < v_2 < \cdots < v_n$ we define $C(x_1, \ldots, x_n)$ to be the circuit that is obtained from C when we assign the label x_i to the input gate v_i (i.e., $C(x_1, \ldots, x_n) = (V, E, g_c, \alpha')$ where $\alpha'(v_i) \overset{df}{=} x_i$ and $\alpha'(g) \overset{df}{=} \alpha(g)$ if g is not an input gate).

Example 1. We present circuits for the expressions given in the introduction.



This circuit produces the set of primes: The complement of $\{0, 1\}$ multiplied with itself gives all composite numbers. So the complement of this set yields the set of primes and additionally 0 and 1. We can remove $0, 1$ with a $\cap$-gate and obtain the set of all primes.



The left circuit in this second example produces all sums of two odd primes. The right circuit produces all even numbers greater than two. Hence, a terminating algorithm that decides whether these two circuits are equivalent would answer the Goldbach conjecture.

Definition 1. *Let* $\mathcal{O} \subseteq \{\cup, \cap, ^{-}, +, \times\}$*. We define* membership problems *and* equivalence problems *for circuits and formulas.*

$$\begin{aligned}
\mathrm{MC}_{\mathbb{N}}(\mathcal{O}) &\overset{df}{=} \{(C,b) \,|\, C \textit{ is an } \mathcal{O}\textit{-circuit, } b \in \mathbb{N}\textit{, and } b \in I(C)\}\\
\mathrm{MF}_{\mathbb{N}}(\mathcal{O}) &\overset{df}{=} \{(C,b) \,|\, C \textit{ is an } \mathcal{O}\textit{-formula, } b \in \mathbb{N}\textit{, and } b \in I(C)\}\\
\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) &\overset{df}{=} \{(C_1,C_2) \,|\, C_1 \textit{ and } C_2 \textit{ are } \mathcal{O}\textit{-circuits such that } I(C_1) = I(C_2)\}\\
\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) &\overset{df}{=} \{(C_1,C_2) \,|\, C_1 \textit{ and } C_2 \textit{ are } \mathcal{O}\textit{-formulas such that } I(C_1) = I(C_2)\}
\end{aligned}$$

When an $\mathcal{O}$-circuit $C = (V, E, g_c, \alpha)$ is used as input for an algorithm, then we use the following encoding: First, V is encoded as a list of increasing natural numbers (which are encoded in binary). Then follows the encoding of E as a list of pairs (v_1, v_2) where $v_1, v_2 \in V$. Multi-edges are encoded by a repeated occurrence of the same pair. Thereafter, the natural number g_C is encoded. Finally, α is encoded as a list of pairs $(v, \alpha(v))$ where $v \in V$. By this encoding of $\mathcal{O}$-circuits, it follows that a circuit with n gates can be encoded by $O(n^2 \log n)$ bits.

Note that it is possible to verify in deterministic logarithmic space whether a given string encodes a valid circuit. In the following, we will therefore assume that all algorithms start with such a validation of their input strings.

We summarize known results about the complexity of equivalence problems.

Theorem 1. *It holds that*

1. *[MW03]* $\mathrm{EC}_{\mathbb{N}}(+)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* $\mathrm{C_{=}L}$.
2. *[MW03]* $\mathrm{EC}_{\mathbb{N}}(+, \times)$ *is in* coNP.

3. *[SM73]* $\mathrm{EF}_{\mathbb{N}}(\cup,+)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* Π_2^{P}.
4. *[SM73]* $\mathrm{EF}_{\mathbb{N}}(^{-},\cup,\cap,+)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* PSPACE.

As demonstrated in the introduction, we can generate interesting sets like the set of all primes using only multiplication and the set operations. Contrary to that, circuits whose only arithmetic operation is addition can only compute finite or cofinite sets.

Proposition 1. *If C is a given circuit over $\mathcal{O} \subseteq \{\cap,\cup,^{-},+\}$, then there exists an $n \leq 2^{|C|}+1$ such that for all $z \geq n$,*

$$z \in I(C) \iff n \in I(C).$$

Proof. We will show this by an induction over the nodes of the circuit, starting with the singleton input sets and for now ignoring our size restriction put on n. During this proof we shall call a part of a set *monotone* if no alternation between contained and not contained numbers exists.

Let $C = (V, E, g_c, \alpha)$ be a given circuit. Recall that all nodes represent subsets of the natural numbers and we have to show the stated property for the set represented by the designated output node g_c. Clearly, the property of getting monotone above a certain value n holds for the input nodes if we chose n to be $c+1$ for c being the single element of the input set. This is because together with $c+1$, all numbers above c are not included in the set.

Let us now assume that S_1 and S_2 are nodes which fulfill our property and let n_1 and n_2 be the numbers above which the sets get monotone. So either all numbers above n_1 are contained in S_1 or all are not. The same applies to S_2 for n_2. It is easy to see that applying any of our three set operations to these two sets will result in a new set that fulfills our property if we chose n to be the bigger of the two numbers n_1 and n_2.

We investigate the case of applying an addition to S_1 and S_2. If both input sets are finite, thus n_1 and n_2 together with all numbers above them are not contained in their respective sets, it is clear, that the resulting set is finite and can't contain $n_1 + n_2$ or any greater number. So we can safely choose $n_1 + n_2$ to be our new n for the property.

We consider the case of one set being finite, let it be S_1 and the other set being infinite. If $S_1 = \emptyset$ then we are done by letting $n = n_1 + n_2$. Otherwise, let us look at the largest member of S_1 and call it m_1, which is at most $n_1 - 1$. Now add m_1 to every monotone member of S_2 starting with n_2. Obviously we obtain all natural numbers above and including $m_1 + n_2$ which is less than $n_1 + n_2$. So again letting our new n be $n_1 + n_2$ suffices and the property is again fulfilled because all greater numbers together with $n_1 + n_2$ are contained in our new set. The case of both sets being infinite is similar in that S_1 now contains infinitely many numbers but with the guaranteed member n_1 which we can again add to the monotone part of S_2.

Having now completed our induction we can be assured that there exists a number n, above which the output set of C contains either every or no further number. All that is left to check is the size of n which was claimed to be less than $2^{|C|}+1$. Obviously this is true for the input nodes, since their single elements are represented in binary and they start to get monotone right above them. The set operations do not increase n and every addition increases its length in binary at most by one. Each $+$ node uses at least one bit in the representation of C. So from the proof above we can see that the binary representation of the final n will not be longer than the biggest input value plus one plus the number of addition nodes and thus the proposition holds. □

Note that by Proposition 1, we gain knowledge about the generated set of a $\{\cap,\cup,^-,+\}$-circuit C once we know the membership for all numbers of length $\leq |C|+1$. Clearly, multiplication gates break this property: For instance, the set of all primes neither is finite nor cofinite, but can be generated by a $\{\cap,\cup,^-,\times\}$-circuit.

It is obvious that $\{\cap,\cup,+,\times\}$-circuits can only compute finite sets. However, the upper bound given in Proposition 1 is too small because of the multiplication gates.

Proposition 2. *It holds that*

1. *If C is a formula over $\mathcal{O}\subseteq\{\cap,\cup,+,\times\}$, then $C\subseteq[0,2^{|C|}]$.*
2. *If C is a $\{\cup,\cap,+,\times\}$-circuit, then $C\subseteq[0,2^{2^{2|C|}}]$.*

Proof. 1. The existence of an n such that $C\subseteq[0,n]$ is clear, because C can only produce finite sets. Now we argue for $n\leq 2^{|C|}$. We prove this by induction over the structure of C. If we ask how big the numbers computed by C are, then it is clear that set operations do not affect this question. So we can assume without loss of generality that C consists only of $+$ and $\times$ gates.

Note that $n\leq 2^{|C|}$ is trivially true for the input gates. Let $C=C_1+C_2$ and suppose $\max I(C_1)\leq 2^{|C_1|}$ and $\max I(C_2)\leq 2^{|C_2|}$. Then we obtain

$$\max I(C)=\max I(C_1)+\max I(C_2)\leq 2^{|C_1|}+2^{|C_2|}\leq 2^{|C_1|+|C_2|}=2^{|C|}.$$

Similarly, for $C=C_1\times C_2$ we obtain

$$\max I(C)=\max I(C_1)\times\max I(C_2)\leq 2^{|C_1|}\cdot 2^{|C_2|}=2^{|C_1|+|C_2|}=2^{|C|}.$$

2. Unfold the circuit C to a formula F. Clearly, $|F|\leq 2^{2|C|}$ and the maximum value computed by F is the same as that in C. By the first statement, this value is bounded by $2^{|F|}$. □

3.1 Relations to Membership Problems

In this section we show that in many cases (i.e., for several $\mathcal{O}\subseteq\{\cap\cup{}^-,+,\times\}$), the complexity of the equivalence problem $\mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ is related to the complexity of $\mathrm{MC}_{\mathbb{N}}(\mathcal{O})$.

For circuits without ×-gates, Lemma 1 gives a general upper bound. In the following sections, we will prove the upper bounds obtained in this way to be optimal in some cases, but we will also present significantly better upper bounds for several other cases.

Lemma 1. *It holds that*

1. *If* $\mathcal{O} \subseteq \{\cap, \cup, ^-, +\}$*, then* $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \in \mathrm{coNP}^{\mathrm{MC}_{\mathbb{N}}(\mathcal{O})}$ *and* $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \in \mathrm{coNP}^{\mathrm{MF}_{\mathbb{N}}(\mathcal{O})}$.
2. *If* $\mathcal{O} \subseteq \{\cap, \cup, +, \times\}$*, then* $\mathrm{EF}_{\mathbb{N}}(\mathcal{O})$ *is in* $\mathrm{coNP}^{\mathrm{MF}_{\mathbb{N}}(\mathcal{O})}$.

Proof. 1. Fix $\mathcal{O} \subseteq \{\cap, \cup, ^-, +\}$. By Proposition 1, in order to gain complete knowledge about the set generated by an $\mathcal{O}$-circuit C, we only have to look at the first $2^{|C|} + 1$ elements in the set. Thus when comparing the outputs of two circuits C_1 and C_2 over $\mathcal{O}$, it suffices to compare all numbers below and including $\max(2^{|C_1|} + 1, 2^{|C_2|} + 1)$. So the following holds for the polynomial $p(x) = x + 1$.

$$\begin{aligned}(C_1, C_2) \in \mathrm{EC}_{\mathbb{N}}(\mathcal{O}) &\iff \forall n\big(n \in I(C_1) \leftrightarrow n \in I(C_2)\big)\\ &\iff \forall n\Big(n \le \max(2^{|C_1|} + 1, 2^{|C_2|} + 1) \to \big(n \in I(C_1) \leftrightarrow n \in I(C_2)\big)\Big)\\ &\iff \forall n\Big(|n| \le p(|(C_1, C_2)|) \to \big(n \in I(C_1) \leftrightarrow n \in I(C_2)\big)\Big)\end{aligned}$$

As we can see this is exactly the representation of a coNP problem if

$$n \in I(C_1) \leftrightarrow n \in I(C_2)$$

can be resolved in polynomial time. This is done by querying our oracle $\mathrm{MC}_{\mathbb{N}}(\mathcal{O})$ twice to check $n \in I(C_1)$ and $n \in I(C_2)$ and compare the equality of the results. This proof also holds for formulas.

2. Utilizing Proposition 2, we can use the same proof as for the first statement.

□

Applying Lemma 1 to the results by McKenzie and Wagner, we obtain:

Corollary 1. *It holds that*

1. $\mathrm{EC}_{\mathbb{N}}(\cup, +)$, $\mathrm{EF}_{\mathbb{N}}(\cup, +)$, $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$, $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, +)$, $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, \times)$, $\mathrm{EF}_{\mathbb{N}}(\cup, +, \times)$*, and* $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, +, \times)$ *are in* Π_2^{P}.
2. $\mathrm{EC}_{\mathbb{N}}(^-, \cap, \cup, +)$ *and* $\mathrm{EF}_{\mathbb{N}}(^-, \cap, \cup, +)$ *are in* PSPACE.

Proof. McKenzie and Wagner [MW03] showed that $\mathrm{MC}_{\mathbb{N}}(\cup,+)$, $\mathrm{MF}_{\mathbb{N}}(\cup,+)$, $\mathrm{MF}_{\mathbb{N}}(\cup,\times)$, $\mathrm{MF}_{\mathbb{N}}(\cup,\cap,+)$, $\mathrm{MF}_{\mathbb{N}}(\cup,\cap,\times)$, $\mathrm{MF}_{\mathbb{N}}(\cup,+,\times)$, and $\mathrm{MF}_{\mathbb{N}}(\cup,\cap,+,\times)$ are in NP. Furthermore, they showed that $\mathrm{MC}_{\mathbb{N}}(^-,\cap,\cup,+)$ and $\mathrm{EF}_{\mathbb{N}}(^-,\cap,\cup,+)$ are in PSPACE. So the statements in the corollary follow from Lemma 1. □

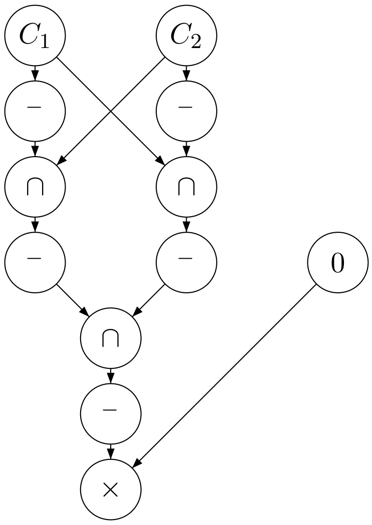
It turns out that in some cases, the equivalence problem is not harder than the membership problem. More precisely, we can reduce the equivalence problem $\mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ to $\mathrm{MC}_{\mathbb{N}}(\mathcal{O})$ if $\{\cap,^-,\times\} \subseteq \mathcal{O}$ or $\{\cup,^-,\times\} \subseteq \mathcal{O}$.

Proposition 3. *If* $\{\cap,^-,\times\} \subseteq \mathcal{O}$ *or* $\{\cup,^-,\times\} \subseteq \mathcal{O}$, *then* $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\mathrm{log}} \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$.

Proof. Assume $\{\cap,^-,\times\} \subseteq \mathcal{O}$. The following function witnesses $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\mathrm{log}} \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$.

$$f(C_1,C_2) \overset{df}{=} (\overline{C},0)$$

where C denotes the following circuit.



Clearly, if C_1 and C_2 are circuits over $\mathcal{O}$, then C is a circuit over $\mathcal{O}$. Also, observe that f is computable in deterministic logarithmic space.

If $(C_1,C_2) \in \mathrm{EC}_{\mathbb{N}}(\mathcal{O})$, then $(\overline{C_1}\cap C_2) = \emptyset$ and $(C_1\cap\overline{C_2}) = \emptyset$. Hence $\overline{(\overline{C_1} \cap C_2)}\cap\overline{(C_1 \cap \overline{C_2})} = \mathbb{N}$ and so $C = \emptyset$. This shows $(\overline{C},0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$. Otherwise, if $(C_1,C_2) \notin \mathrm{EC}_{\mathbb{N}}(\mathcal{O})$, then $C_1 \neq C_2$ and hence, $(\overline{C_1}\cap C_2) \neq \emptyset$ or $(\overline{C_1}\cap C_2) \neq \emptyset$. It follows that $\overline{(\overline{C_1} \cap C_2)}\cap\overline{(C_1 \cap \overline{C_2})} \neq \mathbb{N}$ and so $C = \{0\}$. Therefore, $(\overline{C},0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$. This shows $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\mathrm{log}} \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$ via reduction f.

By replacing $\cap$ by $\cup$ according to De Morgan's law, we obtain the same result under the assumption $\{\cup,^-,\times\} \subseteq \mathcal{O}$. □

Corollary 2. $\mathrm{EC}_{\mathbb{N}}(^-, \cap, \cup, \times), \mathrm{EF}_{\mathbb{N}}(^-, \cap, \cup, \times), \mathrm{EC}_{\mathbb{N}}(\cap, \cup, \times) \in \mathrm{PSPACE}$

Proof. The first assertion follows immediately from Proposition 3 and [MW03]. The other two follow from $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ and $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}') \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ if $\mathcal{O}' \subseteq \mathcal{O}$. □

The following proposition shows that in many cases, the intuition that equivalence problems are a generalization of membership problems is correct.

Proposition 4. *If* $\{\cap\} \subseteq \mathcal{O}$ *or* $\{\cup\} \subseteq \mathcal{O}$*, then* $\mathrm{MC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ *and* $\mathrm{MF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{EF}_{\mathbb{N}}(\mathcal{O})$.

Proof. Let $\{\cap\} \subseteq \mathcal{O}$. We define our reduction function f by

$$f(C, b) = (C \cap \{b\}, \{b\})$$

for a circuit C and a $b \in \mathbb{N}$. Now b is in the output of C if and only if $C \cap \{b\} = \{b\}$. So f is the desired reduction.

For $\{\cup\} \subseteq \mathcal{O}$ we obtain a similar reduction by defining

$$f(C, b) = (C \cup \{b\}, C).$$

In both cases, if we start with a formula C, then $f(C, b)$ is a formula as well. □

There exist similar reductions if we allow only arithmetic operations.

Proposition 5. *If* $\mathcal{O} \subseteq \{+, \times\}$*, then* $\mathrm{MC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ *and* $\mathrm{MF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{EF}_{\mathbb{N}}(\mathcal{O})$.

Proof. Observe that in these cases, each circuit computes a singleton. So a membership test reduces to an equivalence test via the straightforward reduction $f(C, b) = (C, \{b\})$. □

We summarize lower bounds resulting from the propositions above.

Corollary 3. *It holds that*

1. $\mathrm{EC}_{\mathbb{N}}(^-, \cup, \cap, +, \times)$ *and* $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, +, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* NEXP.
2. $\mathrm{EF}_{\mathbb{N}}(^-, \cup, \cap, +, \times)$, $\mathrm{EC}_{\mathbb{N}}(^-, \cup, \cap, +)$, $\mathrm{EC}_{\mathbb{N}}(^-, \cup, \cap, \times)$, $\mathrm{EF}_{\mathbb{N}}(^-, \cup, \cap, +)$, $\mathrm{EF}_{\mathbb{N}}(^-, \cup, \cap, \times)$, $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, +)$, $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, \times)$, *and* $\mathrm{EC}_{\mathbb{N}}(\cup, +, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* PSPACE.
3. $\mathrm{EC}_{\mathbb{N}}(^-, \cup, \cap)$, $\mathrm{EC}_{\mathbb{N}}(\cup, \cap)$, $\mathrm{EC}_{\mathbb{N}}(\cap, +, \times)$, *and* $\mathrm{EC}_{\mathbb{N}}(+, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* P.
4. $\mathrm{EC}_{\mathbb{N}}(\cap, +)$, $\mathrm{EC}_{\mathbb{N}}(\cap, \times)$, *and* $\mathrm{EC}_{\mathbb{N}}(+)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* $\mathrm{C}_=\mathrm{L}$.
5. $\mathrm{EC}_{\mathbb{N}}(\cup)$, $\mathrm{EC}_{\mathbb{N}}(\cap)$, *and* $\mathrm{EC}_{\mathbb{N}}(\times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* NL.
6. $\mathrm{EF}_{\mathbb{N}}(^-, \cup, \cap)$, $\mathrm{EF}_{\mathbb{N}}(\cup, \cap)$, $\mathrm{EF}_{\mathbb{N}}(\cup)$, $\mathrm{EF}_{\mathbb{N}}(\cap)$, $\mathrm{EF}_{\mathbb{N}}(\cap, +)$, $\mathrm{EF}_{\mathbb{N}}(\cap, \times)$, $\mathrm{EF}_{\mathbb{N}}(\cap, +, \times)$, $\mathrm{EF}_{\mathbb{N}}(+)$, $\mathrm{EF}_{\mathbb{N}}(\times)$, *and* $\mathrm{EF}_{\mathbb{N}}(+, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-hard for* L.

Proof. Follows directly from Propositions 4 and 5 using the results from McKenzie and Wagner [MW03]. □

4 Feasible Equivalence Problems

In this section, we analyze several equivalence problems for which we can show that efficient evaluation algorithms exist. While the algorithms presented in the first part all require deterministic polynomial time or less, we need randomization for the problems in the second part of the section.

4.1 Equivalence Problems Solvable in Polynomial Time

Lemma 2. *Let C be a circuit over $\mathcal{O} \subseteq \{\cup, \cap, ^{-}\}$ with inputs from $I \subseteq \mathbb{N}$. For any $x, y \in \mathbb{N} - I$,*

$$x \in C \iff y \in C.$$

Proof. We show this by structural induction over the operations. Observe that the assertion is true for a circuit with no operations (i.e., just input gates). Now consider two subcircuits C_1, C_2 of C for which we assume $(x \in I(C_1) \iff y \in I(C_1))$ and $(x \in I(C_2) \iff y \in I(C_2))$ for all $x, y \in \mathbb{N} - I$. For $C_1 \cup C_2$ the following equivalency holds:

$$\begin{aligned} x \in I(C_1 \cup C_2) &\iff x \in I(C_1) \text{ or } x \in I(C_2) \iff y \in I(C_1) \text{ or } y \in I(C_2) \iff \\ &\iff y \in I(C_1) \cup I(C_2) = I(C_1 \cup C_2) \end{aligned}$$

The proof for $\cap$ works analogously. Now consider $\overline{C_1}$, there we have

$$x \in I(\overline{C_1}) \iff x \notin I(C_1) \iff y \notin I(C_1) \iff y \in I(\overline{C_1}).$$

□

Proposition 6. *If $\mathcal{O} \subseteq \{\cup, \cap, ^{-}\}$, then $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{T}}^{\log} \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$ and $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{T}}^{\log} \mathrm{MF}_{\mathbb{N}}(\mathcal{O})$*

Proof. Let C_1 and C_2 be circuits over $\mathcal{O} \subseteq \{\cup, \cap, ^{-}\}$ and let I_1 and I_2 be the sets of inputs. We describe the reduction $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{T}}^{\log} \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$ which will also work for formulas.

For every $i \in I_1 \cup I_2 \cup \{\max(I_1 \cup I_2) + 1\}$, we check $(C_1, i) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O}) \iff (C_2, i) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O})$. If this holds for all i, then we accept, otherwise we reject.

Clearly, the algorithm describes a $\leq_{\mathrm{T}}^{\log}$-reduction to $\mathrm{MC}_{\mathbb{N}}(\mathcal{O})$. We now prove its correctness.

C_1 and C_2 are equal if and only if for all $x \in \mathbb{N}$ it holds that $x \in C_1 \iff x \in C_2$. Let $x \in \mathbb{N}$. If $x \in I_1 \cup I_2$, then the algorithm tests if C_1 and C_2 are equal at x. Otherwise, let $n \overset{df}{=} \max(I_1 \cup I_2) + 1$. By Lemma 2, $(x \in C_1 \iff n \in C_1)$ and $(x \in C_2 \iff n \in C_2)$. So the equality at x is also tested. □

Corollary 4. $\mathrm{EF}_{\mathbb{N}}(\cup), \mathrm{EF}_{\mathbb{N}}(\cap), \mathrm{EF}_{\mathbb{N}}(\cup, \cap), \mathrm{EF}_{\mathbb{N}}(\cup, \cap, ^{-}) \in \mathrm{L}$, $\mathrm{EC}_{\mathbb{N}}(\cup), \mathrm{EC}_{\mathbb{N}}(\cap) \in \mathrm{NL}$, *and* $\mathrm{EC}_{\mathbb{N}}(\cup, \cap), \mathrm{EC}_{\mathbb{N}}(\cup, \cap, ^{-}) \in \mathrm{P}$.

Proof. Follows from Proposition 6 and the results by Wagner and McKenzie [MW03], since L, NL, and P are closed under $\leq_{\mathrm{T}}^{\log}$. □

Proposition 7. *If* $\mathcal{O} \subseteq \{+, \times\}$*, then* $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$ *and* $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{m}}^{\log} \mathrm{MF}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$.

Proof. Let C_1, C_2 be two $\mathcal{O}$-circuits. Now consider the $(\mathcal{O} \cup \{\cap, \times\})$-circuit $C \overset{df}{=} (C_1 \cap C_2) \times 0$. Since $\mathcal{O}$ contains no set operations, C_1 and C_2 produce a single number as output and therefore, C_1 and C_2 are equivalent if and only if $C_1 \cap C_2$ is not empty. So we obtain

$$(C_1, C_2) \in \mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \iff (C, 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}).$$

This reduction is computable in logarithmic space and thus shows the assertion. The same reduction works for formulas, since the constructed circuit C is a formula if C_1 and C_2 are formulas. □

Corollary 5. *It holds that* $\mathrm{EF}_{\mathbb{N}}(\times) \in \mathrm{L}$ *and* $\mathrm{EF}_{\mathbb{N}}(+) \in \mathrm{L}$.

Proof. The first statement is a direct consequence of $\mathrm{MF}_{\mathbb{N}}(\times, \cap) \in \mathrm{L}$ [MW03].

To see that $\mathrm{EF}_{\mathbb{N}}(+) \in \mathrm{L}$, observe that for a given formula one can compute in deterministic logarithmic space the list of all inputs that are connected to the output gate. Computing the sum of such a list is also possible in deterministic logarithmic space. This shows $\mathrm{EF}_{\mathbb{N}}(+) \leq_{\mathrm{m}}^{\log} \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a = b\} \in \mathrm{L}$. □

Proposition 8. *If* $\mathcal{O} \subseteq \{\cap, +, \times\}$*, then* $\mathrm{EC}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{T}}^{\log} \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$ *and* $\mathrm{EF}_{\mathbb{N}}(\mathcal{O}) \leq_{\mathrm{T}}^{\log} \mathrm{MF}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$.

Proof. Let C_1, C_2 be two $\mathcal{O}$-circuits. Now consider the circuits $C_1' \overset{df}{=} C_1 \times 0, C_2' = C_2 \times 0$ and $C \overset{df}{=} (C_1 \cap C_2) \times 0$. These are circuits over $\mathcal{O} \cup \{\cap, \times\}$. Since $\mathcal{O}$ contains at most $\cap$ as set operation, C_1 and C_2 produce a single number or the empty set as output.

If C_1 and C_2 are equivalent, then either both circuits produce the empty set or both circuits produce the same value. In the first case, $(C_1', 0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C_2', 0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C, 0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$ and in the second case, $(C_1', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C_2', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$ and $(C, 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$.

If C_1 and C_2 are not equivalent, then either one of them produces the empty set (without loss of generality let C_1 be that circuit) while the other one produces a number or both produce numbers but different ones. In the first case, $(C_1', 0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C_2', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C, 0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$ and in the second case, $(C_1', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C_2', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\}), (C, 0) \notin \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$.
So we can test for equivalence just by verifying that the tests $(C_1', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$, $(C_2', 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$, and $(C, 0) \in \mathrm{MC}_{\mathbb{N}}(\mathcal{O} \cup \{\cap, \times\})$ yield the same result. □

We now show that $\mathrm{EC}_{\mathbb{N}}(\cap,+)$ is complete for the class $\mathrm{C_=L}\vee\mathrm{coC_=L}$. As a useful tool, we introduce *non-emptiness problems* for circuits.

Definition 2. *We define non-emptiness problems for $\mathcal{O}$-circuits and $\mathcal{O}$-formulas.*

$$\mathrm{NEC}_{\mathbb{N}}(\mathcal{O}) \overset{\mathit{df}}{=} \{C \mid C \textit{ is an } \mathcal{O}\textit{-circuit such that } C \neq \emptyset\}.$$
$$\mathrm{NEF}_{\mathbb{N}}(\mathcal{O}) \overset{\mathit{df}}{=} \{C \mid C \textit{ is an } \mathcal{O}\textit{-formula such that } C \neq \emptyset\}.$$

Lemma 3. $\mathrm{NEC}_{\mathbb{N}}(\cap,+)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* $\mathrm{C_=L}$ *and* $\mathrm{NEF}_{\mathbb{N}}(\cap,+) \in \mathrm{L}$.

Proof. The auxiliary construction in this proof uses a new type of gates: An id-gate is a gate that has one input and that computes the identity.

First observe that an $\{\mathrm{id},+\}$-circuit (resp., $\{\mathrm{id},+\}$-formula) C can be translated in logarithmic space into a $\{+\}$-circuit (resp., $\{+\}$-formula) C' such that $C = C'$. On input of an $\{\mathrm{id},+\}$-circuit (resp., $\{\mathrm{id},+\}$-formula) $C = (V, E, g_C)$, the $\{+\}$-circuit (resp., $\{+\}$-formula) C' is obtained as follows: For every gate $g \in V$ whose label is id, change g's label to $+$, add a new input gate with label 0, and use this new input gate as g's second input. This can be done in logarithmic space and from the construction it immediately follows that $C = C'$. As a consequence, $\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)\leq_{\mathrm{m}}^{\log}\mathrm{EC}_{\mathbb{N}}(+)$ and $\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+)\leq_{\mathrm{m}}^{\log}\mathrm{EF}_{\mathbb{N}}(+)$. So by Theorem 1, $\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+) \in \mathrm{C_=L}$ and by Corollary 5, $\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+) \in \mathrm{L}$.

We describe a function f that on input of a $\{\cap,+\}$-circuit (resp., $\{\cap,+\}$-formula) C and a number k outputs an $\{\mathrm{id},+\}$-circuit (resp., $\{\mathrm{id},+\}$-formula). More precisely, for a $\{\cap,+\}$-circuit (resp., $\{\cap,+\}$-formula) $C = (V, E, g_C)$ and a number k, the $\{\mathrm{id},+\}$-circuit (resp., $\{\mathrm{id},+\}$-formula) $f(C,k)$ is obtained as follows: First, for every gate $g \in V$ whose label is $\cap$, change g's label to id and delete the connection to g's second input. Secondly, k will be the new output gate of the circuit $f(C,k)$. Observe that f is computable in logarithmic space.

Claim 1. Let C be a $\{\cap,+\}$-circuit. If g is a gate in C such that $I(g) \neq \emptyset$, then for all k, the gate g in C computes the same set as the gate g in $f(C,k)$.

Proof. Assume the claim does not hold. Choose the first gate g in C and some k such that $I(g) \neq \emptyset$, but the gate g in C computes a different set than the gate g in $f(C,k)$. From the choice of g it follows that if g' is a direct or indirect predecessor of g, then the gate g' in C computes the same set as the gate g' in $f(C,k)$. The function f modifies only gates with label $\cap$. Therefore, g must have label $\cap$. Let g_1 and g_2 be the direct predecessors of g. We have already observed that the gate g_1 (resp., g_2) in C computes the same set as the gate g_1 (resp., g_2) in $f(C,k)$. Note that $\{\cap,+\}$-circuits can only produce sets of cardinality at most 1. So in C, the gates g, g_1, and g_2 compute singletons and it follows that these singletons must be equal. Hence, in $f(C,k)$, the gates g_1 and g_2 compute the same singleton. By the definition of f, the gate g in $f(C,k)$ computes the set $I(g_1)$. This shows that the gate g in $f(C,k)$ computes the same set as the gate g in C. This contradicts our assumption and proves Claim 1. □

Now we define a function h that on input of a $\{\cap,+\}$-circuit (resp., $\{\cap,+\}$-formula) C outputs a particular list of pairs $(C_{1,1},C_{1,2}),(C_{2,1},C_{2,2}),\dots,(C_{r,1},C_{r,2})$ where the $C_{i,j}$ are $\{\mathrm{id},+\}$-circuits (resp., $\{\mathrm{id},+\}$-formulas). Later we will show that h is a conjunctive truth-table reduction from $\mathrm{NEC}_{\mathbb{N}}(\cap,+)$ to $\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)$ and also from $\mathrm{NEF}_{\mathbb{N}}(\cap,+)$ to $\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+)$. The function h is defined by the following algorithm where the input is a $\{\cap,+\}$-circuit (resp., $\{\cap,+\}$-formula) $C=(V,E,g_C)$ with n gates.

```
1. for i = 1 to n
2.    if gate i is connected to $g_C$ and has label $\cap$, then
3.       let $g_1$ and $g_2$ be the direct predecessors of gate i
4.       let $C_1 = f(C, g_1)$ and $C_2 = f(C, g_2)$
5.       output $(C_1, C_2)$
6.    endif
7. next i
```

Observe that for circuits, the test whether gate i is connected to g_C (line 2) can be carried by one query to an NL-oracle. Hence h is computable in deterministic logarithmic space with the help of an NL-oracle. If we restrict to formulas, then the test in line 2 can be carried out in deterministic logarithmic space and h is computable without oracle access. FL denotes the class of functions computable in deterministic logarithmic space. FNL [ÀBJ95] denotes the class of functions computable in deterministic logarithmic space with access to an NL-oracle. So $h \in \mathrm{FNL}$ and restricted to formulas, $h \in \mathrm{FL}$.

Claim 2. $\mathrm{NEC}_{\mathbb{N}}(\cap,+)$ conjunctively truth-table reduces to $\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)$ via function h. Formally, for every $\{\cap,+\}$-circuit C, if $h(C)=(C_{1,1},C_{1,2}),(C_{2,1},C_{2,2}),\dots,(C_{r,1},C_{r,2})$, then

$$C \in \mathrm{NEC}_{\mathbb{N}}(\cap,+) \iff \bigwedge_{j\in[1,r]} (C_{j,1},C_{j,2}) \in \mathrm{EC}_{\mathbb{N}}(\mathrm{id},+). \tag{1}$$

Proof. "$\Longleftarrow$" Assume $C \notin \mathrm{NEC}_{\mathbb{N}}(\cap,+)$, i.e., $C=\emptyset$. So C must contain a gate k with label $\cap$ such that k is connected to the output gate and $I(k)=\emptyset$. Choose the smallest such k and let g_1 and g_2 be the direct predecessors of gate k. So $I(g_1)\neq\emptyset$, $I(g_2)\neq\emptyset$, and $I(g_1)\neq I(g_2)$. By Claim 1, for all k, the gate g_1 (resp., g_2) in C computes the same set as the gate g_1 (resp. g_2) in $f(C,k)$. So for all k, the gates g_1 and g_2 in $f(C,k)$ compute different sets. Therefore, $C_1=f(C,g_1)$ and $C_2=f(C,g_2)$ compute different sets and hence $(C_1,C_2)\notin \mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)$. The pair (C_1,C_2) appears on the list $h(C)$, since k is connected to the output gate g_C and has label $\cap$. Therefore, the right-hand side of (1) is false.

"$\Longrightarrow$" Assume $C \in \mathrm{NEC}_{\mathbb{N}}(\cap,+)$, i.e., $C\neq\emptyset$. Fix any $j\in[1,r]$. The pair $(C_{j,1},C_{j,2})$ appears on the list $h(C)$, because at a certain time, the algorithm made the output $(C_{j,1},C_{j,2})$ in line 5. Assume that at this time the variable i had the value k. So gate k is connected to the output gate g_C and k has the label $\cap$. Let g_1 and g_2 be the direct predecessors of k. So $C_{j,1}=f(C,g_1)$ and $C_{j,2}=f(C,g_2)$. The gates k, g_1, and g_2 are connected to g_C and therefore none of them computes the empty set. It follows that all three gates must compute the same singleton. So by Claim 1, $f(C,g_1)$ and $f(C,g_2)$ compute the same singleton. Therefore, $(C_{j,1},C_{j,2})=(f(C,g_1),f(C,g_2))\in \mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)$. This shows that the right-hand side of (1) is true. □

With the same proof we can show Claim 2 for formulas instead of circuits.

Claim 3. $\mathrm{NEF}_{\mathbb{N}}(\cap,+)$ conjunctively truth-table reduces to $\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+)$ via function h. Formally, for every $\{\cap,+\}$-formula C, if $h(C)=(C_{1,1},C_{1,2}),(C_{2,1},C_{2,2}),\dots,(C_{r,1},C_{r,2})$, then

$$C\in\mathrm{NEF}_{\mathbb{N}}(\cap,+)\iff\bigwedge_{j\in[1,r]}(C_{j,1},C_{j,2})\in\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+).$$

By Claim 2, $\mathrm{NEC}_{\mathbb{N}}(\cap,+)$ conjunctively truth-table reduces to $\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)$ via a function from FNL, i.e., $\mathrm{NEC}_{\mathbb{N}}(\cap,+)\leq_{\mathrm{ctt}}^{\mathrm{FNL}}\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)$. We have seen that $\mathrm{EC}_{\mathbb{N}}(\mathrm{id},+)\in\mathrm{C_{=}L}$. Allender and Ogihara [AO96] show that $\mathrm{C_{=}L}$ is closed under $\leq_{\mathrm{ctt}}^{\mathrm{FNL}}$. Therefore, $\mathrm{NEC}_{\mathbb{N}}(\cap,+)\in\mathrm{C_{=}L}$.

By Claim 3, $\mathrm{NEF}_{\mathbb{N}}(\cap,+)$ conjunctively truth-table reduces to $\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+)$ via a function from FL, i.e., $\mathrm{NEF}_{\mathbb{N}}(\cap,+)\leq_{\mathrm{ctt}}^{\log}\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+)$. This implies $\mathrm{NEF}_{\mathbb{N}}(\cap,+)\in\mathrm{L}$, since $\mathrm{EF}_{\mathbb{N}}(\mathrm{id},+)\in\mathrm{L}$ and L is closed under $\leq_{\mathrm{ctt}}^{\log}$.

McKenzie and Wagner [MW03] show that $\mathrm{MC}_{\mathbb{N}}(+)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\mathrm{C_{=}L}$. Observe that for an arbitrary $\{+\}$-circuit C and a number k it holds that $(C,k)\in\mathrm{MC}_{\mathbb{N}}(+)$ if and only if $C\cap k\in\mathrm{NEC}_{\mathbb{N}}(\cap,+)$. So $\mathrm{MC}_{\mathbb{N}}(+)\leq_{\mathrm{m}}^{\log}\mathrm{NEC}_{\mathbb{N}}(\cap,+)$. This shows that $\mathrm{NEC}_{\mathbb{N}}(\cap,+)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\mathrm{C_{=}L}$. □

Theorem 2. $\mathrm{EC}_{\mathbb{N}}(\cap,+)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* $\mathrm{C_{=}L}\vee\mathrm{coC_{=}L}$.

Proof. We start the proof by showing that $\mathrm{EC}_{\mathbb{N}}(\cap,+)$ belongs to $\mathrm{C_{=}L}\vee\mathrm{coC_{=}L}$.

$$A_1\overset{df}{=}\{(C_1,C_2)\,|\,C_1,C_2\text{ are }\{\cap,+\}\text{-circuits and }C_1\cap C_2\in\mathrm{NEC}_{\mathbb{N}}(\cap,+)\}$$
$$A_2\overset{df}{=}\{(C_1,C_2)\,|\,C_1,C_2\text{ are }\{\cap,+\}\text{-circuits and }C_1,C_2\notin\mathrm{NEC}_{\mathbb{N}}(\cap,+)\}$$

By Lemma 3, $A_1\in\mathrm{C_{=}L}$. From Lemma 3 and the fact that $\mathrm{C_{=}L}$ is closed under union [AO96] it follows that $A_2\in\mathrm{coC_{=}L}$. Observe that $\mathrm{EC}_{\mathbb{N}}(\cap,+)=A_1\cup A_2$. Therefore, $\mathrm{EC}_{\mathbb{N}}(\cap,+)\in\mathrm{C_{=}L}\vee\mathrm{coC_{=}L}$.

Now we show that $\mathrm{EC}_{\mathbb{N}}(\cap,+)$ is $\leq_{\mathrm{m}}^{\log}$-hard for $\mathrm{C_{=}L}\vee\mathrm{coC_{=}L}$. Let

$$B\overset{df}{=}\big\{(C_1,k_1,C_2,k_2)\,\big|\;\begin{array}{l}C_1,C_2\text{ are }\{+\}\text{-circuits}\\ \text{and }\big((C_1,k_1)\in\mathrm{MC}_{\mathbb{N}}(+)\text{ or }(C_2,k_2)\notin\mathrm{MC}_{\mathbb{N}}(+)\big)\big\}.\end{array}$$

The set B is $\leq_{\mathrm{m}}^{\log}$-complete for $\mathrm{C_{=}L}\vee\mathrm{coC_{=}L}$, since $\mathrm{MC}_{\mathbb{N}}(+)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\mathrm{C_{=}L}$ [MW03]. We show $B\leq_{\mathrm{m}}^{\log}\mathrm{EC}_{\mathbb{N}}(\cap,+)$ via the following reduction f.

$$f(C_1,k_1,C_2,k_2)\overset{df}{=}(C,C'),\quad\text{where }C=(C_2\cap k_2)+k_1\text{ and }C'=(C_2\cap k_2)+C_1.$$

Assume $(C_1,k_1,C_2,k_2)\in B$. If $(C_2,k_2)\notin\mathrm{MC}_{\mathbb{N}}(+)$, then $C=C'=\emptyset$ and hence $(C,C')\in\mathrm{EC}_{\mathbb{N}}(\cap,+)$. So assume now $(C_1,k_1)\in\mathrm{MC}_{\mathbb{N}}(+)$. If $C\neq\emptyset$, then $C'\neq\emptyset$ and $(C,C')\in\mathrm{EC}_{\mathbb{N}}(\cap,+)$. Otherwise, $C=\emptyset$ and hence, $C'=\emptyset$ and $(C,C')\in\mathrm{EC}_{\mathbb{N}}(\cap,+)$.

Assume $(C_1, k_1, C_2, k_2) \notin B$. So $(C_1, k_1) \notin \mathrm{MC}_{\mathbb{N}}(+)$ and $(C_2, k_2) \in \mathrm{MC}_{\mathbb{N}}(+)$. It follows that $C = k_1 + k_2$ and $C' = C_1 + k_2$ where $k_1 \notin C_1$. Therefore, $(C, C') \notin \mathrm{EC}_{\mathbb{N}}(\cap, +)$. This shows $B \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\cap, +)$ via f. □

Theorem 3. $\mathrm{EF}_{\mathbb{N}}(\cap, +) \in \mathrm{L}$.

Proof. We define the following sets.

$$A_1 \overset{df}{=} \{(C_1, C_2) \,|\, C_1, C_2 \text{ are } \{\cap, +\}\text{-formulas and } C_1 \cap C_2 \in \mathrm{NEF}_{\mathbb{N}}(\cap, +)\}$$
$$A_2 \overset{df}{=} \{(C_1, C_2) \,|\, C_1, C_2 \text{ are } \{\cap, +\}\text{-formulas and } C_1, C_2 \notin \mathrm{NEF}_{\mathbb{N}}(\cap, +)\}$$

By Lemma 3, $A_1, A_2 \in \mathrm{L}$. Observe that $\mathrm{EF}_{\mathbb{N}}(\cap, +) = A_1 \cup A_2$. So $\mathrm{EF}_{\mathbb{N}}(\cap, +) \in \mathrm{L}$. □

Proposition 9. $\mathrm{EC}_{\mathbb{N}}(\cap, +) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\cap, \times)$ *and* $\mathrm{EC}_{\mathbb{N}}(+) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\times)$.

Proof. First we observe that for a given number e we can construct in deterministic logarithmic space a $\{+\}$-circuit C_e such that $C_e = \{e\}$. If $e = 0$, then this is done by the single-node circuit with input 0. Assume now that $e \geq 1$ and let $\mathrm{bin}(e) = e_n \cdots e_0$. We construct a circuit that consists of gates $v_0, \ldots, v_n$ such that v_0 is an input gate with input 1 and all other gates have label $+$. Moreover, for $i \in [0, n-1]$, there are two edges from gate v_i to gate v_{i+1}. Observe that $I(v_i) = 2^i$ and therefore,

$$\sum_{i, e_i = 1} I(v_i) = e.$$

This sum can be produced by adding at most $n - 1$ additional gates with label $+$ to our circuit and by suitably connecting these new gates to the gates v_i where $e_i = 1$. This results in a circuit C_e such that $C_e = \{e\}$.

As a consequence, there exists a function f computable in deterministic logarithmic space that for a given $\{\cap, +\}$-circuit C computes a $\{\cap, +\}$-circuit $f(C)$ such that $C = f(C)$ and the inputs of $f(C)$ are from $\{0, 1\}$. Moreover, if C is a $\{+\}$-circuit, then $f(C)$ is a $\{+\}$-circuit.

Let g be the function that translates a given $\{\cap, +\}$-circuit C with inputs from $\{0, 1\}$ into the following $\{\cap, \times\}$-circuit $g(C)$: All $+$-gates become $\times$-gates, all inputs 0 become 1, and all inputs 1 become 2. Note that g is computable in deterministic logarithmic space. Observe that $C = \{e\}$ if and only if $g(C) = \{2^e\}$. Moreover, if C is a $\{+\}$-circuit, then $g(C)$ is a $\{+\}$-circuit.

Now it is easy to see that the function $(C_1, C_2) \mapsto (g(f(C_1)), g(f(C_2)))$ performs both reductions, $\mathrm{EC}_{\mathbb{N}}(+) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\times)$ and $\mathrm{EC}_{\mathbb{N}}(\cap, +) \leq_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(\cap, \times)$. □

Corollary 6. $\mathrm{EC}_{\mathbb{N}}(\cap, \times)$ *is* $\leq_{\mathrm{m}}^{\log}$*-hard for* $\mathrm{C_{=}L} \vee \mathrm{coC_{=}L}$ *and* $\mathrm{EC}_{\mathbb{N}}(\times)$ *is* $\leq_{\mathrm{m}}^{\log}$*-hard for* $\mathrm{C_{=}L}$.

Proof. Follows from Theorem 2 and Corollary 3. □

4.2 Equivalence Problems Solvable by Randomized Algorithms

In this section we show that $\mathrm{EC}_{\mathbb{N}}(+,\times)$ and $\mathrm{MC}_{\mathbb{N}}(\cap,+,\times)$ are in RP, i.e., they are decidable in randomized polynomial time. McKenzie and Wagner [MW03] already report $\mathrm{MC}_{\mathbb{N}}(\cap,+,\times) \in \mathrm{RP}$ and attribute it to unpublished work by Glaßer. Here we give the first proof of this result.

Definition 3 ([Gil77]). RP *is the class of languages L for which there exists a probabilistic polynomial-time-bounded Turing machine M such that for all x,*

$$x \in L \implies M(x) \textit{ accepts with probability} \geq \tfrac{1}{2}$$
$$x \notin L \implies M(x) \textit{ accepts with probability } 0$$

The class does not change if $\frac{1}{2}$ is replaced by $\frac{1}{q(|x|)}$ where q is a polynomial.

Theorem 4. $\mathrm{EC}_{\mathbb{N}}(+,\times) \in \mathrm{coRP}$.

Proof. It suffices to show that the non-equivalence problem for $\{+,\times\}$-circuits belongs to RP. This is done by the following nondeterministic algorithm which works on input of two $\{+,\times\}$-circuits C_1 and C_2 where n denotes the input length.

```
1. guess some m $\in [1, 2^{3n}]$
2. if m is not prime then reject
3. $r_1 := (C_1 \bmod m)$ and $r_2 := (C_2 \bmod m)$
4. if $r_1 = r_2$ then reject else accept
```

We observe that the algorithm works in polynomial time in n. Line 2 is realized in polynomial time by the algorithm by Agrawal, Kayal, and Saxena [AKS04]. Line 3 can be carried out in polynomial time, since m is of polynomial size in n and hence, we can determine r_1 and r_2 by performing each operation in the circuits modulo m.

Clearly, if $C_1 = C_2$, then the algorithm rejects for all guesses of m, i.e., it accepts with probability 0. Assume now that $C_1 \neq C_2$. Suppose there exist more than 2^{2n} pairwise different primes $p \in [1, 2^{3n}]$ such that $(C_1 \bmod p) = (C_2 \bmod p)$, i.e., $C_1 \equiv C_2 (\bmod\ p)$. Observe that the product of these primes is greater than or equal to $2^{2^{2n}}$. From the Chinese remainder theorem and the fact that $C_1, C_2 \in [0, 2^{2^{2n}}]$ (Corollary 2) it follows that $C_1 = C_2$. This contradicts our assumption and therefore,

$$(C_1 \bmod p) = (C_2 \bmod p) \text{ for at most } 2^{2n} \text{ primes } p \in [1, 2^{3n}]. \tag{2}$$

Let $\pi(k)$ denote the number of primes $\leq k$. Rosser and Schoenfeld [RS62] prove that for $k \geq 17$,

$$\frac{k}{\ln k} < \pi(k) < 1.25506 \frac{k}{\ln k},$$

where $\ln k$ denotes the natural logarithm of k. So for $n \geq 5$, the number of primes in $[1, 2^{3n}]$ is at least

$$\frac{2^{3n}}{\ln(2^{3n})} \geq \frac{2^{3n}}{3n} \geq 2 \cdot 2^{2n}.$$

Together with (2) this implies that for $n \geq 5$,

$$\text{there exist at least } \frac{2^{3n}}{6n} \text{ primes } p \in [1, 2^{3n}] \text{ such that } (C_1 \bmod p) \neq (C_2 \bmod p). \tag{3}$$

Therefore, if $C_1 \neq C_2$, then the algorithm nondeterministically produces 2^{3n} paths and at least $\frac{2^{3n}}{6n}$ of these paths accept. Thus if $C_1 \neq C_2$, then the algorithm accepts with probability $\geq \frac{1}{6n}$. This shows that the non-equivalence of $\{+, \times\}$-circuits belongs to RP.

□

Corollary 7. *$\mathrm{MC}_{\mathbb{N}}(\cap, +, \times) \in \mathrm{coRP}$.*

Proof. McKenzie and Wagner [MW03] show $\mathrm{MC}_{\mathbb{N}}(\cap, +, \times) \equiv_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(+, \times)$. □

Corollary 8. *$\mathrm{EC}_{\mathbb{N}}(\cap, +, \times) \in \mathrm{BPP}$, $\mathrm{EC}_{\mathbb{N}}(\cap, \times) \in \mathrm{P}$, $\mathrm{EF}_{\mathbb{N}}(\cap, +, \times) \in \mathrm{DLOGCFL}$, and $\mathrm{EF}_{\mathbb{N}}(\cap, \times) \in \mathrm{L}$.*

Proof. By Corollary 7, $\mathrm{MC}_{\mathbb{N}}(\cap, +, \times) \in \mathrm{coRP}$ and hence, by Proposition 8, $\mathrm{EC}_{\mathbb{N}}(\cap, +, \times) \in \mathrm{P}^{\mathrm{coRP}} \subseteq \mathrm{BPP}$. McKenzie and Wagner [MW03] showed that $\mathrm{MC}_{\mathbb{N}}(\cap, \times) \in \mathrm{P}$, $\mathrm{MF}_{\mathbb{N}}(\cap, +, \times) \in \mathrm{DLOGCFL}$, and $\mathrm{MF}_{\mathbb{N}}(\cap, \times) \in \mathrm{L}$. The assertion follows from Proposition 8, since P, DLOGCFL, and L are closed under $\leq_{\mathrm{T}}^{\log}$. □

5 Intractable Equivalence Problems

In this section we analyze equivalence problems which are more difficult to decide than the problems presented in the former section. The scope ranges from Π_2^{P}-complete for the more restricted problems like $\mathrm{EF}_{\mathbb{N}}(\cup, +)$ and $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$ up to NEXP-hard for $\mathrm{EC}_{\mathbb{N}}(^{-}, \cup, \cap, +, \times)$, the most general membership problem we consider. The best upper bound we can give for the complexity of this problem is the Turing-degree of the halting problem.

5.1 Π_2^{P}-Complete Problems

We show Π_2^{P}-completeness for several equivalence problems. Corollary 1 already shows that some of these problems belong to Π_2^{P}. Hence, it suffices to prove Π_2^{P}-hardness for $\mathrm{EF}_{\mathbb{N}}(\cup, +)$ and $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$.

Stockmeyer and Meyer [SM73] showed that the following problem of evaluating quantified Boolean formulas (QBF) is PSPACE-complete.

$$\mathrm{QBF} \overset{df}{=} \big\{H \,\big|\, H \text{ is a Boolean formula in 3-CNF with variables } x_1,\dots,x_n \text{ such that } \underbrace{\exists x_1 \forall x_2 \cdots \exists x_n}_{\text{strict alternation}} \big(H(x_1,\dots,x_n)=1\big)\big\}$$

Furthermore, the following restricted version QBF_2 is complete for Π_2^{P} [SM73].

$$\mathrm{QBF}_2 \overset{df}{=} \big\{H \,\big|\, H \text{ is a Boolean formula in 3-CNF with variables } x_1,\dots,x_{2n} \text{ such that } \forall x_1 \forall x_2 \cdots \forall x_n \exists x_{n+1} \exists x_{n+2} \dots \exists x_{2n} \big(H(x_1,\dots,x_{2n})=1\big)\big\}$$

Travers [Tra04] showed that QSOS, a quantified version of the NP-complete problem *sum of subset (SOS)*, also is PSPACE complete. We here follow this track and show that QSOS_2, a restricted version of QSOS, is $\leq_{\mathrm{m}}^{\log}$-complete for Π_2^{P}.

$$\mathrm{QSOS}_2 \overset{df}{=} \big\{(x_1,\dots,x_{2n},b) \,\big|\, x_1,\dots,x_{2n},b \in \mathbb{N} \text{ and } \forall I \subseteq \{1,\dots,n\} \; \exists J \subseteq \{n+1,\dots,2n\} \; \big(\textstyle\sum_{i\in I} x_i + \sum_{j \in J} x_j = b\big)\big\}$$

Lemma 4. $\mathrm{QBF}_2 \leq_{\mathrm{m}}^{\log} \mathrm{QSOS}_2$.

Proof. We define a logspace computable function f such that $H \in \mathrm{QBF}_2 \Leftrightarrow f(H) \in \mathrm{QSOS}_2$. Let $H = \bigwedge_{i=1}^{m}(z_{i_1} \vee z_{i_2} \vee z_{i_3})$ be a Boolean formula where $z_{i_j} \in \{x_1,\dots,x_{2n},\neg x_1,\dots,\neg x_{2n}\}$.

We then define f as

$$f(H) \overset{df}{=} \big(v_1, v_2, \dots, v_{2n}, \underbrace{0,\dots,0}_{2m}, v_1', v_2', \dots, v_{2n}', c_1, \dots, c_m, d_1, \dots, d_m, b\big),$$

where $b, c_1, \dots, c_m, d_1, \dots, d_m, v_1, \dots, v_{2n}, v_1', \dots, v_{2n}'$ are natural numbers which have the following decimal representations:

$$v_i \stackrel{df}{=} \underbrace{k_1 k_2 \ldots k_m}_{m}\underbrace{0\ldots0\overset{(i)}{1}0\ldots0}_{2n},\ \text{where}$$

k_j is the number of occurrences of literal x_i in the j-th clause of H,

$$v'_i \stackrel{df}{=} \underbrace{k'_1 k'_2 \ldots k'_m}_{m}\underbrace{0\ldots0\overset{(i)}{1}0\ldots0}_{2n},\ \text{where}$$

k'_j is the number of occurrences of literal $\neg x_i$ in the j-th clause of H,

$$c_i \stackrel{df}{=} \underbrace{0\ldots0\overset{(i)}{1}0\ldots0}_{m}\underbrace{0\ldots00\ldots0}_{2n},\quad \text{(balancing vectors)}$$

$$d_i \stackrel{df}{=} \underbrace{0\ldots0\overset{(i)}{2}0\ldots0}_{m}\underbrace{0\ldots00\ldots0}_{2n},\quad \text{(balancing vectors)}$$

$$b \stackrel{df}{=} \underbrace{4\ldots444\ldots4}_{m}\underbrace{1\ldots11\ldots1}_{2n}\quad \text{(target vector).}$$

By defining f in this way, we achieve that all vectors are appropriately quantified in the QSOS-instance: $v_1, \ldots, v_{2n}$ are all quantified universally and all other vectors (except from the target vector which obviously is not quantified at all) are quantified existentially.

In the following, let $I_{a_1,\ldots,a_{2n}}$ be the interpretation which, for $1 \le i \le 2n$, assigns truth-value $a_i \in \{0,1\}$ to variable x_i in H. The following equivalences now hold:

$$\begin{aligned}
H \in \mathrm{QBF}_2 &\Leftrightarrow \forall a_1 \cdots \forall a_n \exists a_{n+1} \cdots \exists a_{2n} (I_{a_1,\ldots,a_{2n}} \text{ satisfies } H)\\
&\Leftrightarrow \forall a_1 \cdots \forall a_n \exists a_{n+1} \cdots \exists a_{2n} (I_{a_1,\ldots,a_{2n}} \text{ satisfies each clause of } H)\\
&\Leftrightarrow \forall_{e_1\in\{0,1\}} \forall_{e_2\in\{0,1\}} \cdots \forall_{e_n\in\{0,1\}} \exists_{e_{n+1}\in\{0,1\}} \exists_{e_{n+2}\in\{0,1\}} \cdots \exists_{e_{2n}\in\{0,1\}}\\
&\qquad \exists_{k_1,\ldots,k_m\in\{1,2,3\}} \Big((\sum_{i=1}^{2n} e_i v_i + (1-e_i) v'_i) = k_1 k_2 \ldots k_m \underbrace{1\ldots1}_{2n} \Big)\\
&\Leftrightarrow \forall_{e_1\in\{0,1\}} \forall_{e_2\in\{0,1\}} \cdots \forall_{e_n\in\{0,1\}} \exists_{e_{n+1}\in\{0,1\}} \exists_{e_{n+2}\in\{0,1\}} \cdots \exists_{e_{2n}\in\{0,1\}}\\
&\qquad \exists_{l_1\in\{0,1\}} \exists_{l_2\in\{0,1\}} \cdots \exists_{l_{2n}\in\{0,1\}}\\
&\qquad \exists_{k_1,\ldots,k_m\in\{1,2,3\}} \Big(\Big((\sum_{i=1}^{2n} e_i v_i + l_i v'_i) \Big) = k_1 k_2 \ldots k_m \underbrace{1\ldots1}_{2n} \Big)\\
&\overset{(\star)}{\Leftrightarrow} \forall_{e_1\in\{0,1\}} \forall_{e_2\in\{0,1\}} \cdots \forall_{e_n\in\{0,1\}} \exists_{e_{n+1}\in\{0,1\}} \exists_{e_{n+2}\in\{0,1\}} \cdots \exists_{e_{2n}\in\{0,1\}}\\
&\qquad \exists_{l_1,\ldots,l_{2n}\in\{0,1\}} \exists_{r_1,\ldots,r_m,r'_1,\ldots,r'_m\in\{0,1\}}\\
&\qquad \Big(\Big((\sum_{i=1}^{2n} e_i v_i + l_i v'_i) + (\sum_{i=1}^{m} r_i c_i + r'_i d_i) \Big) = \underbrace{44\ldots4}_{m}\underbrace{1\ldots1}_{2n} \Big)\\
&\Leftrightarrow \forall I \subseteq \{1,\ldots,n\} \exists J_1 \subseteq \{n+1,\ldots,2n\} \exists J_2, J_3 \subseteq \{1,\ldots,m\}\\
&\qquad \Big(\sum_{i\in I} v_i + \sum_{i\in J_1} v'_i + \sum_{i\in J_2} c_i + \sum_{i\in J_3} d_i = \underbrace{44\ldots4}_{m}\underbrace{1\ldots1}_{2n} \Big)\\
&\Leftrightarrow f(H) \in \mathrm{QSOS}_2.
\end{aligned}$$

To see $(\star)$, observe that the target sum $\overbrace{44\dots4}^{m}\overbrace{1\dots1}^{2n}$ can be obtained by adding appropriate balancing vectors to a number which has the decimal representation

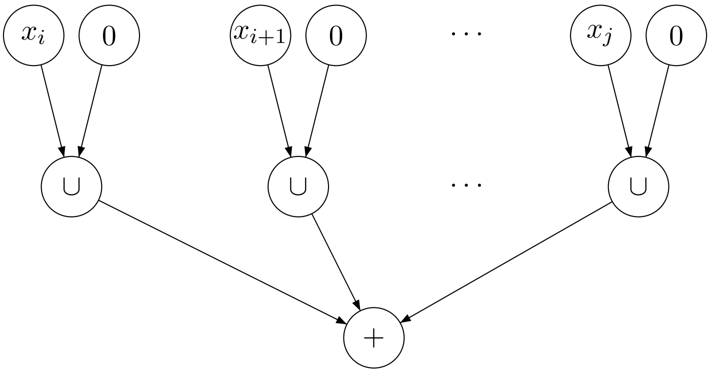
$$k_1k_2\dots k_m\underbrace{1\dots1}_{2n}$$

where $k_i \in \{1,2,3\}$ for $1 \le i \le n$. This can be done since all balancing vectors are quantified existentially. Since f is computable in logarithmic space, our reduction is complete. □

Corollary 9. QSOS_2 *is* $\le_{\mathrm{m}}^{\log}$*-complete for* Π_2^{P}*.*

Theorem 5. $\mathrm{EF}_{\mathbb{N}}(\cup,+)$ *is* $\le_{\mathrm{m}}^{\log}$*-complete for* Π_2^{P}*.*

Proof. By Corollary 9, the problem QSOS_2 is Π_2^{P}-complete. We here describe a reduction from QSOS_2 to $\mathrm{EF}_{\mathbb{N}}(\cup,+)$. To model the sums in the definition of QSOS_2 we define the formula $C_{x_{i,j}}$ for $1 \le i \le j \le 2n$ as follows.



Note that the final addition must be expanded appropriately. For $i > j$ we define $C_{x_{i,j}} \overset{df}{=} 0$. It is obvious that $C_{x_{i,j}} = \{\sum\limits_{k\in I} x_k \mid I \subseteq \{i, i+1, \dots, j\}\}$ for $i \ge 1$, $j \in \mathbb{N}$. Let us also define $C_y \overset{df}{=} \sum\limits_{j=n+1}^{2n} x_j$. Using these formulas, the following equivalency holds:

$$\begin{aligned}
&(x_1,\dots,x_n,x_{n+1},\dots,x_{2n},b) \in \mathrm{QSOS}_2\\
&\iff \forall I \subseteq \{1,\dots,n\} \exists J \subseteq \{n+1,\dots,2n\}, \sum_{i\in I} x_i + \sum_{j\in J} x_j = b\\
&\iff \forall I \subseteq \{1,\dots,n\} \exists J \subseteq \{n+1,\dots,2n\}, \sum_{i\in I} x_i + \sum_{j=n+1}^{2n} x_j = b + \sum_{j\notin J} x_j\\
&\iff \forall I \subseteq \{1,\dots,n\} \exists J \subseteq \{n+1,\dots,2n\}, \sum_{i\in I} x_i + \sum_{j=n+1}^{2n} x_j = b + \sum_{j\in J} x_j\\
&\iff \forall l \in \left(C_{x_{1,n}} + C_y\right) \exists r \in \left(b + C_{x_{n+1,2n}}\right), l = r\\
&\iff \left(C_{x_{1,n}} + C_y\right) \subseteq \left(b + C_{x_{n+1,2n}}\right)\\
&\iff \left(C_{x_{1,n}} + C_y\right) \cup \left(b + C_{x_{n+1,2n}}\right) = \left(b + C_{x_{n+1,2n}}\right)\\
&\iff \left(\left(C_{x_{1,n}} + C_y\right) \cup \left(b + C_{x_{n+1,2n}}\right), \left(b + C_{x_{n+1,2n}}\right)\right) \in \mathrm{EC}_{\mathbb{N}}(\cup,+)
\end{aligned}$$

This reduction can be carried out in logarithmic space. Thus, since QSOS_2 is Π_2^{P}-complete, $\mathrm{EC}_{\mathbb{N}}(\cup,+)$ is Π_2^{P}-hard. □

To show Π_2^{P}-hardness for $\mathrm{EF}_{\mathbb{N}}(\cup,\times)$, we define an analogue of QSOS_2 for products.

$$\mathrm{QPOS}_2 \overset{\mathit{df}}{=} \big\{(x_1,\ldots,x_{2n},b) \,\big|\, x_1,\ldots,x_{2n},b \geq 1 \text{ and } \forall I \subseteq \{1,\ldots,n\}\exists J \subseteq \{n+1,\ldots,2n\} \\ \big(\textstyle\prod_{i\in I} x_i \prod_{j\in J} x_j = b\big)\big\}$$

Lemma 5. $\mathrm{QBF}_2 \leq_{\mathrm{m}}^{\log} \mathrm{QPOS}_2$.

Proof. The proof is similar to the one for Lemma 4. However, the obvious idea to use the same proof and to just encode the weights of a QSOS_2-instance into exponents of numbers does not work, since these numbers become too large. Instead, we will represent weights by several but small exponents. We define a logspace computable function f such that $H \in \mathrm{QBF}_2 \Leftrightarrow f(H) \in \mathrm{QPOS}_2$. Let $H = \bigwedge_{i=1}^{m}(z_{i_1} \vee z_{i_2} \vee z_{i_3})$ with $z_{i_j} \in \{x_1,\ldots,x_{2n},\neg x_1,\ldots,\neg x_{2n}\}$ be a boolean formula.

We then define f as

$$f(H) \overset{\mathit{df}}{=} \big(v_1,\ldots,v_n,\underbrace{1,\ldots,1}_{3n+2m},v_{n+1},\ldots,v_{2n},v'_1,v'_2,\ldots,v'_{2n},p_1,\ldots,p_m,p_1^2,\ldots,p_m^2,b\big),$$

where p_i is the i-th prime and $v_1,\ldots,v_{2n},v'_1,\ldots,v'_{2n}$ and b are the following natural numbers:

$$v_i \overset{\mathit{df}}{=} p_{m+i}\prod_{r=1}^{m} p_r^{k_r}, \text{ where}$$

k_s is the number of occurrences of literal x_i in the r-th clause of H,

$$v'_i \overset{\mathit{df}}{=} p_{m+i}\prod_{r=1}^{m} p_r^{k'_r}, \text{ where}$$

k'_r is the number of occurrences of literal $\neg x_i$ in the r-th clause of H,

$$b \overset{\mathit{df}}{=} \prod_{i=1}^{2n} p_{m+i}\prod_{r=1}^{m} p_r^4 \quad \text{(target vector)}.$$

By defining f in this way, we achieve that all vectors are appropriately quantified in the QPOS-instance: $v_1,\ldots,v_n$ are all quantified universally and all other vectors (except for the target vector which obviously is not quantified at all) are quantified existentially.

In the following, let $I_{a_1,\ldots,a_{2n}}$ be the interpretation that, for $1 \leq i \leq 2n$, assigns the truth-value $a_i \in \{0,1\}$ to variable x_i in H.

The following equivalences now hold:

$$\begin{aligned}
H \in \mathrm{QBF}_2 &\Leftrightarrow \forall a_1 \cdots \forall a_n \exists a_{n+1} \cdots \exists a_{2n} (I_{a_1,\ldots,a_{2n}} \text{ satisfies } H) \\
&\Leftrightarrow \forall a_1 \cdots \forall a_n \exists a_{n+1} \cdots \exists a_{2n} (I_{a_1,\ldots,a_{2n}} \text{ satisfies each clause of } H) \\
&\overset{(\star)}{\Leftrightarrow} \forall_{I\subseteq\{1,\ldots,n\}} \exists_{J\subseteq\{n+1,\ldots,2n\}} \exists_{k_1,\ldots,k_m\in\{1,2,3\}} \text{ such that} \\
&\quad \prod_{i\in I} v_i \prod_{i\notin I} v_i' \prod_{j\in J} v_j \prod_{j\notin J} v_j' = \prod_{i=1}^{2n} p_{m+i} \prod_{r=1}^{m} p_r^{k_r} \\
&\Leftrightarrow \forall_{I\subseteq\{1,\ldots,n\}} \exists_{J\subseteq\{n+1,\ldots,2n\}} \exists_{R_1\subseteq\{1,\ldots,m\}} \exists_{R_2\subseteq\{1,\ldots,m\}} \text{ such that} \\
&\quad \prod_{i\in I} v_i \prod_{i\notin I} v_i' \prod_{j\in J} v_j \prod_{j\notin J} v_j' \prod_{r\in R_1} p_r \prod_{r\in R_2} p_r^2 = \prod_{i=1}^{2n} p_{m+i} \prod_{r=1}^{m} p_r^4 = b \\
&\Leftrightarrow \forall_{I\subseteq\{1,\ldots,n\}} \exists_{J\subseteq\{n+1,\ldots,2n\}} \exists_{I'\subseteq\{1,\ldots,2n\}} \exists_{R_1\subseteq\{1,\ldots,m\}} \exists_{R_2\subseteq\{1,\ldots,m\}} \text{ such that} \\
&\quad \prod_{i\in I} v_i \prod_{j\in J} v_j \prod_{i\in I'} v_i' \prod_{r\in R_1} p_r \prod_{r\in R_2} p_r^2 = b \\
&\Leftrightarrow f(H) \in \mathrm{QPOS}_2.
\end{aligned}$$

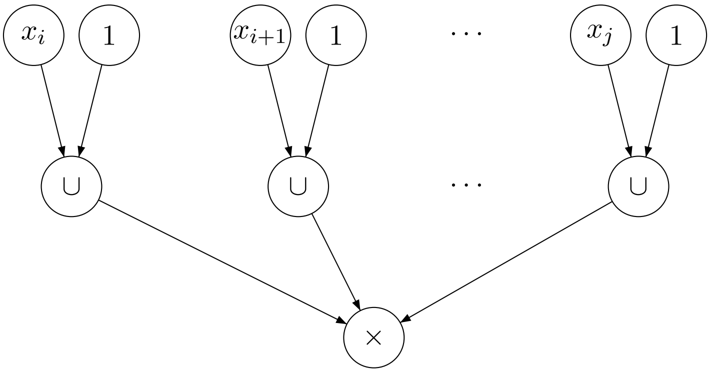
To see ($\star$), observe that the first product on the right hand side enforces that for each variable exactly one value out of $\{0,1\}$ is chosen (this is actually only needed later on), and the second product makes sure that in each clause at least one literal is true (the exponents k_j are not zero).

Since the primes are logarithmic in the input length they can be computed in logarithmic space. Thus, f is computable in logarithmic space and our reduction is complete. □

Corollary 10. QPOS_2 *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* Π_2^{P}*.*

Theorem 6. $\mathrm{EF}_{\mathbb{N}}(\cup,\times)$ *is* $\leq_{\mathrm{m}}^{\log}$*-hard for* Π_2^{P}*.*

Proof. We describe a reduction from QPOS_2 to $\mathrm{EF}_{\mathbb{N}}(\cup,\times)$. To model the products in the definition of QPOS_2 we define the formula $C_{x_{i,j}}$ for $1 \leq i \leq j \leq 2n$ as shown in the following diagram.



Note that the final product must be expanded appropriately. For $i > j$ we define additionally $C_{x_{i,j}} \overset{df}{=} 1$. It is obvious that $C_{x_{i,j}} = \{\prod\limits_{k \in I} x_k \mid I \subseteq \{i, i+1, \ldots, j\}\}$ for $i \geq 1$,$j \in \mathbb{N}$. Let us also define $C_y \overset{df}{=} \prod\limits_{j=n+1}^{2n} x_j$.

Our reduction function f is now defined as follows: If the input to the function is not a tuple, has an even number of elements, or one of the elements is zero, then the function returns $(0, 1)$. Otherwise we define:

$$f((x_1, \ldots, x_{2n}, b)) \overset{df}{=} \left(\left(C_{x_{1,n}} \times C_y\right) \cup \left(b \times C_{x_{n+1,2n}}\right), \left(b \times C_{x_{n+1,2n}}\right)\right)$$

The correctness is observed as follows.

$$\begin{aligned}
&(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n}, b) \in \mathrm{QPOS}_2 \\
\iff\ & \forall_{I \subseteq \{1,\ldots,n\}} \exists_{J \subseteq \{n+1,\ldots,2n\}}, \prod_{i \in I} x_i \prod_{j \in J} x_j = b \\
\iff\ & \forall_{I \subseteq \{1,\ldots,n\}} \exists_{J \subseteq \{n+1,\ldots,2n\}}, \prod_{i \in I} x_i \prod_{j=n+1}^{2n} x_j = b \prod_{j \notin J} x_j \\
\iff\ & \forall_{I \subseteq \{1,\ldots,n\}} \exists_{J \subseteq \{n+1,\ldots,2n\}}, \prod_{i \in I} x_i \prod_{j=n+1}^{2n} x_j = b \prod_{j \in J} x_j \\
\iff\ & \forall \left(l \in \left(C_{x_{1,n}} \times C_y\right)\right) \exists \left(r \in \left(b \times C_{x_{n+1,2n}}\right)\right), l = r \\
\iff\ & \left(C_{x_{1,n}} \times C_y\right) \subseteq \left(b \times C_{x_{n+1,2n}}\right) \\
\iff\ & \left(C_{x_{1,n}} \times C_y\right) \cup \left(b \times C_{x_{n+1,2n}}\right) = \left(b \times C_{x_{n+1,2n}}\right) \\
\iff\ & \left(\left(C_{x_{1,n}} \times C_y\right) \cup \left(b \times C_{x_{n+1,2n}}\right), \left(b \times C_{x_{n+1,2n}}\right)\right) \in \mathrm{EC}_{\mathbb{N}}(\cup, \times)
\end{aligned}$$

Again, f can be computed in logarithmic space. Thus, by Corollary 10, $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$ is $\leq_{\mathrm{m}}^{\log}$-complete for Π_2^{P}. □

In order to show that $\mathrm{EC}_{\mathbb{N}}(\cup, \times)$ belongs to Π_2^{P} we reduce $\mathrm{EC}_{\mathbb{N}}(\cup, \times)$ to $\mathrm{EC}_{\mathbb{N}}(\cup, +)$. The idea is as follows: We represent numbers in the $\{\cup, \times\}$-circuits as products of the form $q_1^{e_1} q_2^{e_2} \cdots q_m^{e_m}$ and we build corresponding $\{\cup, +\}$-circuits that generate exactly the vectors $(e_1, e_2, \ldots, e_m)$ which we will appropriately encode as numbers. Note that a factorization into prime factors q_i would be welcome, but is not possible in polynomial time. However, as demonstrated by McKenzie and Wagner, for our purpose, a factorization into factors that are relatively prime suffices. For this end we need the following problem.

Definition 4 ([BS96]). GCD-Free Basis (GFB) *is the following problem:*

Given: *Numbers* $a_1, \ldots, a_n \geq 1$

Compute: *Numbers* $m \geq 1, q_1, \ldots, q_m \geq 2$ *and* $e_{11}, \ldots, e_{nm} \geq 0$ *such that*

$$\gcd(q_i, q_j) = 1 \textit{ for } i \neq j \textit{ and } a_i = \prod_{j=1}^{m} q_j^{e_{ij}} \textit{ for } i = 1, \ldots, n.$$

Proposition 10 ([BS96]). *GCD-Free Basis can be computed in polynomial time.*

First we establish the reduction for $\{\cup,\times\}$-circuits that do not generate 0 in their outputs.

Lemma 6. *There exists a polynomial-time computable function f that maps pairs of $\{\cup,\times\}$-circuits to pairs of $\{\cup,+\}$-circuits such that for any two $\{\cup,\times\}$-circuits C_1 and C_2 with positive inputs,*

$$(C_1, C_2) \in \mathrm{EC}_{\mathbb{N}}(\cup,\times) \iff f(C_1,C_2) \in \mathrm{EC}_{\mathbb{N}}(\cup,+).$$

Proof. We transform the $\{\cup,\times\}$-circuit into a $\{\cup,+\}$-circuit with the same structure such that the transformed circuit operates on the exponents of the GCD-free basis of the inputs. To this end, we replace each $\times$-node by a $+$-node and each input by a number whose binary representation consists of blocks of fixed length. Each such block contains the exponent of one component of the base. Also, the blocks are long enough to ensure that they do not interfere with each other.

Let C_1 and C_2 be two $\{\cup,\times\}$-circuits. If their numbers of input gates do not match, introduce additional input gates with assignment one that are multiplied to the output gate. We denote the input gates of C_1 and C_2 by $x_1,\dots,x_n$ and $\tilde{x}_1,\dots,\tilde{x}_n$, respectively. Now compute a GCD-free basis for $x_1,\dots,x_n,\tilde{x}_1,\dots,\tilde{x}_n$. Let this basis be $q_1,\dots,q_m$. By Proposition 10, this is possible in polynomial time (note that all inputs are positive). For $x=\prod_{j=1}^m q_j^{e_j}$, let $\varepsilon(x,j)=e_j$ be the unique j-th exponent in the gcd-representation of x.

For $N \overset{df}{=} 2^{2|C_1|+2|C_2|}$ we now define a mapping $\sigma : (\mathbb{N}^+,\times) \to (\mathbb{N},+)$ by $x \mapsto \sum_{j=1}^m \varepsilon(x,j)N^{j-1}$. Note that σ is a monoid-homomorphism and that it is one-one on $X \overset{df}{=} \{x \mid \varepsilon(x,j) < N \text{ for } j=1,\dots,m\}$.

Now let $f(C_1,C_2)=(C_1',C_2')$ where C_1' results from C_1 by replacing all $\times$-gates by $+$-gates and replacing the inputs x_i by $\sigma(x_i)$ (C_2' is obtained analogously). Since σ is a homomorphism, we obtain $\sigma(C_1)=C_1'$ and $\sigma(C_2)=C_2'$.

By Proposition 2, for all numbers x produced by gates of the circuits C_1 and C_2 it holds that $x < 2^{2^{2|C_1|+2|C_2|}}$, and thus $\varepsilon(x,j) < 2^{2|C_1|+2|C_2|} = N$. Therefore, σ is one-one in C_1 and C_2 and we obtain $C_1 = C_2 \iff \sigma(C_1)=\sigma(C_2) \iff C_1' = C_2'$. This shows $(C_1,C_2)\in \mathrm{EC}_{\mathbb{N}}(\cup,\times)$ if and only if $f(C_1,C_2)\in\mathrm{EC}_{\mathbb{N}}(\cup,+)$. □

Proposition 11. $\mathrm{EC}_{\mathbb{N}}(\cup,\times) \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{EC}_{\mathbb{N}}(\cup,+)$

Proof. By Lemma 6, it suffices to construct a polynomial-time computable function that transforms a pair of $\{\cup,\times\}$-circuits (C_1,C_2) into a pair of $\{\cup,\times\}$-circuits (C_1',C_2') such that all inputs of C_1' and C_2' are positive and $(C_1 = C_2 \iff C_1' = C_2')$.

Let C_1 and C_2 be two $\{\cup,\times\}$-circuits and assume without loss of generality that each of these circuits is connected. If neither C_1 nor C_2 has zero as input, then we are done by returning (C_1,C_2). If exactly one of the circuits has zero as input, then we know that this circuit has zero in its output, but the other circuits has not. So we are done by returning two fixed non-equivalent circuits. From now on assume that both circuits have inputs that are zero and hence they have zero in their output.

Claim 4. There exists a polynomial-time computable function that on input of a $\{\cup, \times\}$-circuit C returns the following: If $C = \{0\}$ then the algorithm outputs "C computes the singleton $\{0\}$". Otherwise, the output is a $\{\cup, \times\}$-circuit C' such that $C' = C - \{0\}$.

Observe that with an easy recursive algorithm, we can determine in polynomial time whether the set generated by a gate in C contains 0 and whether this set contains a positive number. If the output gate of C does not contain a positive number, then $C = \{0\}$ and we are done by returning "C computes the singleton $\{0\}$". Otherwise, the output gate of C contains at least one positive number. Let $\tilde{C}$ be the circuit that is obtained from C if each gate that does not generate a positive number is replaced by a new input gate with label 0. Clearly, $\tilde{C} = C$. So in $\tilde{C}$, each gate that is not an input gate and that is connected with the output gate generates a set that contains positive numbers. Therefore, in $\tilde{C}$, the direct successors of input gates that are zero and that are connected to the output gate must be $\cup$-gates. Let C' be the circuit that is obtained from $\tilde{C}$ by deleting all gates not connected to the output gate and all input gates that are zero (by doing the latter, the adjacent $\cup$-gates become edges). Observe that apart from 0, the gates in C' generate the same sets as the gates in $\tilde{C}$. This proves Claim 4.

Applying Claim 4 to C_1 and C_2. If both, C_1 and C_2, compute the singleton $\{0\}$, then we are done by returning two fixed equivalent circuits. If exactly one of the circuits computes the singleton $\{0\}$, then we are done by returning two fixed non-equivalent circuits. Otherwise, the algorithm in Claim 4 returns two circuits C_1' and C_2' such that $C_1' = C_1 - \{0\}$ and $C_2' = C_2 - \{0\}$. So we are done by returning (C_1', C_2'). □

Corollary 11. *The problems* $\mathrm{EC}_{\mathbb{N}}(\cup, +)$*,* $\mathrm{EC}_{\mathbb{N}}(\cup, \times)$*,* $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, +, \times)$*,* $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, +)$*,* $\mathrm{EF}_{\mathbb{N}}(\cup, \cap, \times)$*,* $\mathrm{EF}_{\mathbb{N}}(\cup, +, \times)$*,* $\mathrm{EF}_{\mathbb{N}}(\cup, +)$*,* $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$ *are* $\leq_{\mathrm{m}}^{\log}$*-complete for* Π_2^{P}*.*

Proof. $\mathrm{EF}_{\mathbb{N}}(\cup, +)$ and $\mathrm{EF}_{\mathbb{N}}(\cup, \times)$ are $\leq_{\mathrm{m}}^{\log}$-hard for Π_2^{P} by the Theorems 5 and 6. So all mentioned problems are $\leq_{\mathrm{m}}^{\log}$-hard for Π_2^{P}. The problems belong to Π_2^{P} by Corollary 1 and Proposition 11. □

Proposition 12. $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, +, \times)$ *and* $\mathrm{EC}_{\mathbb{N}}(\cup, +, \times)$ *are in* $\mathrm{coNEXP}^{\mathrm{NP}}$*.*

Proof. It suffices to consider $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, +, \times)$. By Proposition 2, the maximum value in the output of a $\{\cup, \cap, +, \times\}$-circuit C is bounded by $2^{2^{2|C|}}$. We describe a nondeterministic exponential time oracle Turing machine that accepts $\mathrm{EC}_{\mathbb{N}}(\cup, \cap, +, \times)$. For given circuits C_1 and C_2 choose nondeterministically a number n between 0 and $\max\{2^{2^{2|C_1|}}, 2^{2^{2|C_2|}}\}$. Unfold the circuits C_1 and C_2 to formulas F_1 and F_2. Query the NP oracle for $(F_1, n) \in \mathrm{MF}_{\mathbb{N}}(\cap, \cup, +, \times)$ and $(F_2, n) \in \mathrm{MF}_{\mathbb{N}}(\cap, \cup, +, \times)$. (This is possible with help of an NP-oracle, since $\mathrm{MF}_{\mathbb{N}}(\cap, \cup, +, \times) \in \mathrm{NP}$ [MW03].) Accept if the answers are equal, otherwise reject.

We observe that $(C_1, C_2) \in \mathrm{EC}_{\mathbb{N}}(\cup, \cap, +, \times)$ if and only if all paths accept: By Proposition 2, the outputs of both circuits are bounded by $\max\{2^{2^{2|C_1|}}, 2^{2^{2|C_2|}}\}$. So the circuits are equivalent if and only if for every number n between 0 and $\max\{2^{2^{2|C_1|}}, 2^{2^{2|C_2|}}\}$ it holds that $n \in I(C_1) \iff n \in I(C_2)$. This is what the algorithm checks. □

5.2 Upper Bounds for $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$

In this section we analyze the complexity of the most general equivalence problem, $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$. From the Propositions 3 and 4 it follows that $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times) \equiv_{\mathrm{m}}^{\mathrm{log}} \mathrm{MC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$.

Every decision algorithm for $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$ would enable us to automatically verify Goldbach's conjecture. This means that we run the algorithm on input of the circuit that formulates Goldbach's conjecture (this circuits is shown in the introduction of this paper) and the algorithm definitely tells us whether or not the conjecture is true.

It is possible that $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$ is undecidable, but at the moment, we cannot prove this. Observe that the problem $\mathrm{EF}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$ shares the same fate: Obviously, a terminating decision procedure for $\mathrm{EF}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$ can also be used to decide $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$ by simply unfolding the circuit into a (possibly exponentially larger) formula before feeding it to the decision algorithm.

The best we can show is that the Turing degree of the halting problem is an upper bound for $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$. To see this, we consider circuits over positive natural numbers. Breunig [Bre03] showed that if we restrict the range to $\mathbb{N}^+$ (instead of $\mathbb{N}$), then the general membership problem for circuits is decidable in PSPACE. We will utilize this result to show that we can solve $\mathrm{EC}_{\mathbb{N}}(^-,\cup,\cap,+,\times)$ if the evaluation algorithm has oracle access to the halting problem.

Theorem 7 ([Bre03]). $\mathrm{MC}_{\mathbb{N}^+}(^-,\cup,\cap,+,\times)$ *is* $\leq_{\mathrm{m}}^{\mathrm{log}}$*-complete for* PSPACE*.*

Lemma 7. *There exists an oracle Turing machine M with oracle* K *(the halting problem) that on input of a $\{^-,\cup,\cap,+,\times\}$-circuit C over $\mathbb{N}$ outputs a $\{^-,\cup,\cap,+,\times\}$-circuit D over $\mathbb{N}^+$ and a set $Z \subseteq \{0\}$ such that $C = D \cup Z$.*

Proof. M is defined by the following algorithm. Variables denoted by C represent circuits over $\mathbb{N}$, variables denoted by D represent circuits over $\mathbb{N}^+$, and variables denoted by Z represent subsets of $\{0\}$. On input of a circuit C over $\mathbb{N}$, the algorithm simulates C by a circuit D over $\mathbb{N}^+$ where a possible element 0 is stored in the separate set Z. More precisely, with help of recursive calls, the algorithm first determines $\mathbb{N}^+$-circuits D and sets Z that correspond to all direct predecessors of C's output gate, and then it joins the obtained circuits and sets in an appropriate way.

```
function M(C)
if C's output gate $g_C$ is an input gate then
   if $g_C$ has label $l > 0$ then return $(\{l\},\emptyset)$ else return $(\emptyset,\{0\})$
endif
// here $g_C$ is not an input gate
if $g_C$ has label $^-$ then
```

```
 7.    let $C'$ be the circuit obtained from $C$ by defining $g_C$'s direct
       predecessor to be the new output gate and by deleting $g_C$
 8.    let $(D, Z) = M(C')$
 9.    return $(\overline{D}, \{0\} - Z)$
10. endif
11. // here $g_C$ has a label from $\{\cap, \cup, +, \times\}$
12. let $C_1$ ($C_2$) be the circuit obtained from $C$ by defining $g_C$'s left
    (right) direct predecessor to be the new output gate and by deleting $g_C$
13. let $(D_1, Z_1) = M(C_1)$ and $(D_2, Z_2) = M(C_2)$
14. if $g_C$ has label $\cap$ then return $(D_1 \cap D_2, Z_1 \cap Z_2)$
15. if $g_C$ has label $\cup$ then return $(D_1 \cup D_2, Z_1 \cup Z_2)$
16. if $g_C$ has label $+$ then
17.    if $Z_1 = \{0\}$ then $D'_2 = D_2$ else $D'_2 = \emptyset$
18.    if $Z_2 = \{0\}$ then $D'_1 = D_1$ else $D'_1 = \emptyset$
19.    return $((D_1 + D_2) \cup D'_1 \cup D'_2, Z_1 \cap Z_2)$
20. endif
21. // here $g_C$ has label $\times$
22. if $Z_1 = \{0\}$ and $D_2 \neq \emptyset$ (det. via oracle access) then $Z'_1 = \{0\}$ else $Z'_1 = \emptyset$
23. if $Z_2 = \{0\}$ and $D_1 \neq \emptyset$ (det. via oracle access) then $Z'_2 = \{0\}$ else $Z'_2 = \emptyset$
24. return $(D_1 \times D_2, Z'_1 \cup Z'_2 \cup (Z_1 \cap Z_2))$
```

First observe that whenever the algorithm makes a recursive call (lines 8 and 13), then it calls an instance smaller than C. So the algorithm terminates. Also, it is easy to observe that the algorithm outputs a $\{^-, \cup, \cap, +, \times\}$-circuit over $\mathbb{N}^+$ and a subset of $\{0\}$.

A straightforward induction over the circuit size shows that if the algorithm outputs (D, Z) then $C = D \cup Z$. For treating the $\times$-gates the algorithm needs oracle access at lines 22 and 23. By Theorem 7, the membership problem for $\{^-, \cup, \cap, +, \times\}$-circuits over $\mathbb{N}^+$ is decidable. Therefore, the non-emptiness problem is computably enumerable and hence can be answered with one query to our oracle, the halting problem. □

Theorem 8. $\mathrm{MC}_{\mathbb{N}}(^-, \cup, \cap, +, \times) \in \deg_{\mathrm{T}}(\mathrm{K})$ *where* K *denotes the halting problem.*

Proof. We describe a Turing reduction to the halting problem. Let (C, n) be the input. With help of Lemma 7 we transform the $\{^-, \cup, \cap, +, \times\}$-circuit C over $\mathbb{N}$ into a $\{^-, \cup, \cap, +, \times\}$-circuit D over $\mathbb{N}^+$ and a set $Z \subseteq \{0\}$ such that $C = D \cup Z$. (By doing so we make queries to the halting problem.) If $n = 0$, then we accept if and only if $Z = \{0\}$. Otherwise, we accept if and only if $(D, n) \in \mathrm{MC}_{\mathbb{N}^+}(^-, \cup, \cap, +, \times)$. By Theorem 7, the latter is decidable in polynomial space. □

Corollary 12. $\mathrm{EC}_{\mathbb{N}}(^-, \cup, \cap, +, \times) \in \deg_{\mathrm{T}}(\mathrm{K})$ *where* K *denotes the halting problem.*

Proof. By the Propositions 3 and 4, $\mathrm{EC}_{\mathbb{N}}(^-, \cup, \cap, +, \times) \equiv_{\mathrm{m}}^{\log} \mathrm{MC}_{\mathbb{N}}(^-, \cup, \cap, +, \times)$. □

Proposition 13. *The problem* $\mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$ *is recursively-enumerable if and only if* $\mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$ *is decidable.*

Proof. A set A is decidable if and only if A and $\overline{A}$ are recursively enumerable. From Propositions 3 and 4 it follows that $\mathrm{MC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times) \equiv_{\mathrm{m}}^{\log} \mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$. Making use of complementation gates, it is easy to show that $\mathrm{MC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times) \equiv_{\mathrm{m}}^{\log} \overline{\mathrm{MC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)}$. Hence, $\mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times) \equiv_{\mathrm{m}}^{\log} \overline{\mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)}$. □

6 Conclusions

In Table 1 we summarize upper and lower bounds for $\mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ and $\mathrm{EF}_{\mathbb{N}}(\mathcal{O})$ for different sets of operations. In general, equivalence problems are more difficult to solve than their corresponding membership problems. Two circuits are equivalent if for all natural numbers x, they coincide with respect to membership of x. The difference between equivalence and membership becomes even more apparent if one realizes that in general, this universal quantifier is not polynomially bounded in the length of the circuits. For example, a circuit that contains ×-gates can produce numbers of exponential length in its output. So an equivalence test has to make sure that two given circuits of size n agree with respect to membership of all x such that $|x| \leq 2^{2n}$. In some cases (e.g., $\mathrm{EC}_{\mathbb{N}}(\cup,\times)$) it is possible to condense the search space such that one ends at a polynomially-bounded universal quantifier. In other cases (e.g., $\mathrm{EC}_{\mathbb{N}}(\cup,+,\times)$) we were not able to establish such a polynomial bound. We leave open whether the bounds for $\mathrm{EC}_{\mathbb{N}}(\cup,+,\times)$ and $\mathrm{EC}_{\mathbb{N}}(\cup,\cap,+,\times)$ can be improved.

The most general case we consider is the equivalence problem for $\{^{-},\cup,\cap,+,\times\}$-circuits. As discussed in the introduction, Goldbach's conjecture can be formulated as such an equivalence problem. This explains why we were not able to find decidable upper bounds for $\mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$ and $\mathrm{EF}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$. However, with the Turing-degree of the halting problem we identify at least one non-trivial upper bound. This yields the first non-trivial upper bound for $\mathrm{MC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$ and $\mathrm{MF}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$. It is possible that $\mathrm{EC}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$ and $\mathrm{EF}_{\mathbb{N}}(^{-},\cup,\cap,+,\times)$ are undecidable, but so far the best provable lower bound is NEXP. We leave as our most challenging open question whether these problems are decidable. Here we only know that they are either decidable or not recursively enumerable.

We prove that coRP is an upper bound for $\mathrm{EC}_{\mathbb{N}}(+,\times)$. From this it follows that $\mathrm{EC}_{\mathbb{N}}(\cap,+,\times)$ belongs to BPP. Moreover, since $\mathrm{MC}_{\mathbb{N}}(\cap,+,\times)$ and $\mathrm{EC}_{\mathbb{N}}(+,\times)$ are equivalent [MW03], we obtain coRP as an improved upper bound for $\mathrm{MC}_{\mathbb{N}}(\cap,+,\times)$. Still these upper bounds for $\mathrm{EC}_{\mathbb{N}}(+,\times)$, $\mathrm{EC}_{\mathbb{N}}(\cap,+,\times)$, and $\mathrm{MC}_{\mathbb{N}}(\cap,+,\times)$ do not match the lower bound P. We would like to know whether the upper bounds can be improved to P.

When comparing the complexities of membership problems with their corresponding equivalence problems, we notice that usually the complexity either stays the same or increases significantly because of the earlier discussed universal quantifier. When looking at the equivalence problem for $\{\cap, +\}$-circuits we observe a completely different behavior. While the complexity of $\mathrm{MC}_{\mathbb{N}}(\cap, +)$ is $\mathrm{C}_{=}\mathrm{L}$, the complexity of $\mathrm{EC}_{\mathbb{N}}(\cap, +)$ is $\mathrm{coC}_{=}\mathrm{L}(2)$, which is the complement of the second level of the Boolean hierarchy over $\mathrm{C}_{=}\mathrm{L}$. So here we observe a moderate jump of the complexity. For the related problem $\mathrm{EC}_{\mathbb{N}}(\cap, \times)$ we obtain $\mathrm{coC}_{=}\mathrm{L}(2)$ as lower bound, but we leave open whether this is also an upper bound. Similarly, we would like to know matching bounds for $\mathrm{EC}_{\mathbb{N}}(\times)$.

	$\mathrm{EC}_{\mathbb{N}}$		$\mathrm{EF}_{\mathbb{N}}$	
$\mathcal{O}$	Lower Bound	Upper Bound	Lower Bound	Upper Bound
$^{-}\ \cup\ \cap\ +\ \times$	NEXP (C3)	$\deg_{\mathrm{T}}(\mathrm{K})$ (C12)	PSPACE (C3)	$\deg_{\mathrm{T}}(\mathrm{K})$ (C12)
$^{-}\ \cup\ \cap\ +$	PSPACE (C3)	PSPACE (L1)	PSPACE (T1)	PSPACE (T1)
$^{-}\ \cup\ \cap\ \times$	PSPACE (C3)	PSPACE (C2)	PSPACE (C3)	PSPACE (C2)
$^{-}\ \cup\ \cap$	P (C3)	P (C4)	L (C3)	L (C4)
$\cup\ \cap\ +\ \times$	NEXP (C3)	$\mathrm{coNEXP}^{\mathrm{NP}}$ (L12)	Π_2^{P} (C11)	Π_2^{P} (C11)
$\cup\ \cap\ +$	PSPACE (C3)	PSPACE (L1)	Π_2^{P} (C11)	Π_2^{P} (C11)
$\cup\ \cap\ \times$	PSPACE (C3)	PSPACE (C2)	Π_2^{P} (C11)	Π_2^{P} (C11)
$\cup\ \cap$	P (C3)	P (C4)	L (C3)	L (C4)
$\cup\ +\ \times$	PSPACE (C3)	$\mathrm{coNEXP}^{\mathrm{NP}}$ (L12)	Π_2^{P} (C11)	Π_2^{P} (C11)
$\cup\ +$	Π_2^{P} (C11)	Π_2^{P} (C11)	Π_2^{P} (T1)	Π_2^{P} (T1)
$\cup\ \times$	Π_2^{P} (C11)	Π_2^{P} (C11)	Π_2^{P} (C11)	Π_2^{P} (C11)
$\cup$	NL (C3)	NL (C4)	L (C3)	L (C4)
$\cap\ +\ \times$	P (C3)	BPP (C8)	L (C3)	DLOGCFL (C8)
$\cap\ +$	$\mathrm{coC}_{=}\mathrm{L}(2)$ (T2)	$\mathrm{coC}_{=}\mathrm{L}(2)$ (T2)	L (C3)	L (T3)
$\cap\ \times$	$\mathrm{coC}_{=}\mathrm{L}(2)$ (C6)	P (C8)	L (C3)	L (C8)
$\cap$	NL (C3)	NL (C4)	L (C3)	L (C4)
$+\ \times$	P (C3)	coRP (T4)	L (C3)	DLOGCFL (C8)
$+$	$\mathrm{C}_{=}\mathrm{L}$ (T1)	$\mathrm{C}_{=}\mathrm{L}$ (T1)	L (C3)	L (C5)
$\times$	$\mathrm{C}_{=}\mathrm{L}$ (C6)	P (C8)	L (C3)	L (C5)

Table 1. Upper and lower bounds for $\mathrm{EC}_{\mathbb{N}}(\mathcal{O})$ and $\mathrm{EF}_{\mathbb{N}}(\mathcal{O})$. All lower bounds are with respect to $\leq_{\mathrm{m}}^{\log}$-reductions and the numbers in parentheses refer to the corresponding theorems (T), corollaries (C), or lemmas (L). The PSPACE-completeness of $\mathrm{EF}_{\mathbb{N}}(^{-}, \cup, \cap, +)$ and the Π_2^{P}-completeness of $\mathrm{EF}_{\mathbb{N}}(\cup, +)$ were shown by Stockmeyer and Meyer [SM73]. The $\mathrm{C}_{=}\mathrm{L}$-completeness of $\mathrm{EC}_{\mathbb{N}}(+)$ was shown by McKenzie and Wagner [MW03]. For $\mathrm{EC}_{\mathbb{N}}(^{-}, \cup, \cap, +, \times)$ and $\mathrm{EF}_{\mathbb{N}}(^{-}, \cup, \cap, +, \times)$ we only have $\deg_{\mathrm{T}}(\mathrm{K})$, the Turing degree of the halting problem, as upper bound. It is possible that these problems are undecidable.

Acknowledgements

The authors are grateful to Thomas Geier, Daniel Meister, and Marco Nehmeier for interesting discussions and various valuable hints.

References

[ÀBJ95] C. Àlvarez, J. L. Balcázar, and B. Jenner. Adaptive logspace reducibility and parallel time. *Mathematical Systems Theory*, 28(2):117–140, 1995.

[AKS04] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160:781–793, 2004.

[All97] E. Allender. Making computation count: Arithmetic circuits in the nineties. *SIGACT NEWS*, 28(4):2–15, 1997.

[AO96] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *RAIRO – Theoretical Informatics and Applications*, 30:1–21, 1996.

[Bre03] H. Breunig. Die Komplexität von Auswertungsproblemen für Schaltkreise über Mengen positiver natürlicher Zahlen. Diploma Thesis, University of Würzburg, 2003.

[BS96] E. Bach and J. Shallit. *Algorithmic Number Theory*, volume I: Efficient Algorithms of *Foundations of Computing*. The MIT Press, Cambridge, MA, 1996.

[Che66] J. R. Chen. On the representation of a large even integer as the sum of a prime and the product of at most two primes. *Kexue Tongbao*, 17:385–386, 1966.

[Che73] J. R. Chen. On the representation of a large even integer as the sum of a prime and the product of at most two primes I. *Scientia Sinica*, 16:157–176, 1973.

[Che78] J. R. Chen. On the representation of a large even integer as the sum of a prime and the product of at most two primes II. *Scientia Sinica*, 16:421–430, 1978.

[Gil77] J. Gill. Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, 6:675–695, 1977.

[HL23] G. H. Hardy and J. E. Littlewood. Some problems of 'partitio numerorum' III: On the expression of a number as a sum of primes. *Acta Mathematica*, 44:1–70, 1923. Reprinted in *Collected Papers of G. H. Hardy*, Vol. I, pp. 561-630, Clarendon Press, Oxford, 1966.

[Lev63] H. Levy. On Goldbach's theorem. *Mathematical Gazette*, 47:274, October 1963.

[MS72] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential time. In *Proceedings 13th Symposium on Switching and Automata Theory*, pages 125–129. IEEE Computer Society Press, 1972.

[MW03] P. McKenzie and K. W. Wagner. The complexity of membership problems for circuits over sets of natural numbers. In *Proceedings 20th Symposium on Theoretical Aspects of Computer Science*, volume 2607 of *Lecture Notes in Computer Science*, pages 571–582. Springer Verlag, 2003.

[Ram95] O. Ramaré. On Schnirelmann's constant. *Ann. Sc. Norm. Super. Pisa*, 22(4):645–706, 1995.

[RS62] J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6:64–97, 1962.

[SM73] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *Proceedings 5th ACM Symposium on the Theory of Computing*, pages 1–9. ACM Press, 1973.

[Sud78] I. H. Sudborough. On the tape complexity of deterministic context-free languages. *Journal of the ACM*, 25(3):405–414, 1978.

[Tra04] S. Travers. The complexity of membership problems for circuits over sets of integers. In *Proceedings 29th International Symposium on Mathematical Foundations of Computer Science*, volume 3153 of *Lecture Notes in Computer Science*, pages 322–333. Springer Verlag, 2004.

[Vin37] I. M. Vinogradov. Representation of an odd number as a sum of three primes. *Dokl. Akad. Nauk SSSR*, 15:169–172, 1937. English translation in *Goldbach Conjecture*, ed. Wang Yuan, World Scientific, 1984.

[Wag84] K. Wagner. The complexity of problems concerning graphs with regularities. In *Proceedings Mathematical Foundations of Computer Science*, volume 176 of *Lecture Notes in Computer Science*, pages 544–552. Springer-Verlag, 1984.

[WW85] K. W. Wagner and G. Wechsung. On the boolean closure of NP. In *Proceedings International Conference on Fundamentals of Computation Theory*, volume 199 of *Lecture Notes in Computer Science*, pages 485–493. Springer-Verlag, 1985.

[Yan00] K. Yang. Integer circuit evaluation is PSPACE-complete. In *IEEE Conference on Computational Complexity*, pages 204–213, 2000.