The Complexity of Membership Problems for Circuits over Sets of Natural Numbers

Pierre McKenzie
Informatique et recherche opérationnelle
Université de Montréal
C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H3C 3J7 Canada
mckenzie@iro.umontreal.ca

Klaus W. Wagner
Theoretische Informatik
Bayerische Julius-Maximilians-Universität Würzburg
Am Hubland, D-97074 Würzburg, Germany
wagner@informatik.uni-wuerzburg.de

Classification: Computational complexity

Abstract. The problem of testing membership in the subset of the natural numbers produced at the output gate of a $\{\cup, \cap, ^-, +, \times\}$ combinational circuit is shown to capture a wide range of complexity classes. Although the general problem remains open, the case $\{\cup, \cap, +, \times\}$ is shown NEXPTIME-complete, the cases $\{\cup, \cap, ^-, \times\}$, $\{\cup, \cap, \times\}$, $\{\cup, \cap, +\}$ are shown PSPACE-complete, the case $\{\cup, +\}$ is shown NP-complete, the case $\{\cap, +\}$ is shown $C_=L$-complete, and several other cases are resolved. Interesting auxiliary problems are used, such as testing nonemptyness for union-intersection-concatenation circuits, and expressing each integer, drawn from a set given as input, as powers of relatively prime integers of one's choosing. Our results extend in nontrivial ways past work by Stockmeyer and Meyer (1973), Wagner (1984) and Yang (2000).

1 Introduction

Combinational circuits permeate complexity theory. Countless lower bounds, complexity class characterizations, and completeness results involve circuits over the boolean semiring (see [Vo00] among many others). Circuits and formulas over more general structures have been studied as well (see for a few examples [BCGR92, BM95, CMTV98, BMPT97, AJMV98, AAD00]).

In this paper, we study circuits operating on sets of natural numbers. Our universe of natural numbers includes 0 and is simply called the set $\mathbb{N}$ of *numbers* from now on. Next to the boolean semiring, the semiring of numbers is certainly the most fundamental, and two results involving number arithmetic have appeared recently: iterated number multiplication was finally shown to belong to uniform TC^0 [He01, ABH01, CDL] and $(\cup, +, \times)$-circuit evaluation was shown PSPACE-hard [Ya00].

In the boolean setting, the AND and the OR operations combine to capture alternation (in, say, simulations of alternating Turing machines by circuits). In the setting of numbers, perhaps the closest analogs to the AND and the OR become the $\cap$ and the $\cup$. To make sense, this requires an adjustment: since gates will now compute sets of numbers, a $+$-gate and a $\times$-gate having input gates computing $S_1 \subseteq \mathbb{N}$ and $S_2 \subseteq \mathbb{N}$ then compute $\{a + b : a \in S_1, b \in S_2\}$ and $\{a \times b : a \in S_1, b \in S_2\}$ respectively.

Another reason to study $\{\cup, \cap, +, \times\}$-circuits over $\mathbb{N}$ is that they obey some form of monotonicity condition: if the set $S \subset \mathbb{N}$ carried by an input gate is replaced by a larger set $S' \supset S$, then the set computed at the output gate of the circuit can only become larger (never smaller). This is reminescent of monotone boolean functions (a boolean function is monotone if flipping an input from 0 to 1 can never change the output from 1 to 0), for which significant complexity bounds are known. Are the circuit and formula evaluation problems over subsets of $\{\cup, \cap, +, \times\}$ related to monotone boolean function complexity?

The problem studied in this paper is the following, which we think of as combining number arithmetic with some form of alternation: given a number b and a $\{\cup, \cap, ^-, +, \times\}$-circuit C with number inputs, does b belong to the set computed by the output gate of C? We call this problem MC$(\cup, \cap, ^-, +, \times)$, and we call MF$(\cup, \cap, ^-, +, \times)$ the same problem restricted to formulas. Note that a complement gate $^-$ applied to a finite set $S \subset \mathbb{N}$ computes the infinite set $\mathbb{N} \setminus S$. Hence, in the presence of complement gates, the brute force strategy which would exhaustively compute all the sets encountered in the circuit fails. The notation for restricted versions of these problems, for instance MC$(\cup, +)$, is self-explanatory (see section 2).

Beyond the results mentioned above, it was known prior to this work that the problem MF$(\cup, +)$ is NP-complete [SM73], that MF$(\cup, \cap, ^-, +)$ and MF$(\cup, ^-, +)$ are PSPACE-complete [SM73], and that MC$(\cup, +)$ is in PSPACE [Wa84].

Adding results from the present paper, we obtain Table I. We highlight here some of the interesting results or techniques:

- The problem MC$(\cup, \cap, +, \times)$ is NEXPTIME-complete. As an intermediate step, we prove that determining whether a union-intersection-concatenation circuit over a finite alphabet produces a nonempty set is also NEXPTIME-complete.
- The problem MC$(\cup, \cap, \times)$ is PSPACE-complete.
- The problem MC$(\cup, \cap, \times)$ reduces in polynomial time to MC$(\cup, \cap, +)$, which is thus also PSPACE-complete. This reduction is possible because the following problem is solvable in polynomial time [BS96]: given as input a set S of numbers excluding 0, compute a set T of pairwise relatively prime numbers and express each $m \in S$ as a product of powers of the numbers in T.
- The problem MC$(\cup, +)$ is NP-complete. This is a nontrivial improvement over the former PSPACE upper bound.
- The problem MC$(\cap, +)$ is $\mathrm{C_=L}$-complete, and so is the problem of testing whether two MC$(+)$-circuits are equivalent.

The hardest problem we consider is MC$(\cup, \cap, ^-, +, \times)$. As will be seen, we do not have an upper bound for this problem and it may well be undecidable. In fact, we will see that Goldbach's conjecture is easily expressible as an instance of MC$(\cup, \cap, ^-, +, \times)$. On the other hand, Table I shows that its various restrictions hit upon a wealth of complexity classes.

2 Definitions and Known Results

A *circuit* $C = (G, E, g_C)$ is a finite directed acyclic graph (G, E) with a specified node g_C, the *output gate*. The gates with indegree 0 are called the *input gates*.

We consider different types of arithmetic circuits. Let $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, \times\}$. An *$\mathcal{O}$-circuit* $C = (G, E, g_C, \alpha)$ is a circuit (G, E, g_C) whose gates have indegree 0, 1, or 2 and are labelled by the function $\alpha : G \mapsto \mathcal{O} \cup \mathbb{N}$ in the following way: Every input gate g has a label $\alpha(g) \in \mathbb{N}$, every gate g with indegree 1 has the label $\alpha(g) = ^-$, and every gate g with indegree 2 has a label $\alpha(g) \in \{\cup, \cap, +, \times\}$. For each of its gates g the arithmetic circuit C computes a set $I(g) \subseteq \mathbb{N}$ inductively defined as follows:

- If g is an input gate with label a then $I(g) =_{\text{def}} \{a\}$.
- If g is $+$-gate with predecessors g_1, g_2 then $I(g) =_{\text{def}} \{k+m : k \in I(g_1) \wedge m \in I(g_2)\}$.
- If g is $\times$-gate with predecessors g_1, g_2 then $I(g) =_{\text{def}} \{k \cdot m : k \in I(g_1) \wedge m \in I(g_2)\}$.
- If g is a $\cup$-gate with predecessors g_1, g_2 then $I(g) =_{\text{def}} I(g_1) \cup I(g_2)$.
- If g is a $\cap$-gate with predecessors g_1, g_2 then $I(g) =_{\text{def}} I(g_1) \cap I(g_2)$.
- If g is a $^-$-gate with predecessor g_1 then $I(g) =_{\text{def}} \mathbb{N} \setminus I(g_1)$.

The set computed by C is $I(C) =_{\text{def}} I(g_C)$. If $I(g) = \{a\}$ then we also write $I(g) = a$. An *$\mathcal{O}$-formula* is an $\mathcal{O}$-circuit with maximal outdegree 1. For $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, \times\}$ the *membership problems* for $\mathcal{O}$-circuits and $\mathcal{O}$-formulae are

$$\text{MC}(\mathcal{O}) =_{\text{def}} \{(C, b) : C \text{ is an } \mathcal{O}\text{-circuit and } b \in \mathbb{N} \text{ such that } b \in I(C)\}$$

and

$$\text{MF}(\mathcal{O}) =_{\text{def}} \{(F, b) : F \text{ is an } \mathcal{O}\text{-formula and } b \in \mathbb{N} \text{ such that } b \in I(F)\}.$$

For simplicity we write $\text{MC}(o_1, \ldots o_r)$ instead of $\text{MC}(\{o_1, \ldots o_r\})$, and we write $\text{MF}(o_1, \ldots o_r)$ instead of $\text{MF}(\{o_1, \ldots o_r\})$.

Example 2.1 *(Prime numbers) A circuit* PRIMES *such that* I(PRIMES) *is the set of prime numbers is obtained by defining the subcircuit* GE2 *as* $\overline{0 \cup 1}$ *and defining* PRIMES *as* $\text{GE2} \cap (\overline{\text{GE2} \times \text{GE2}})$. *Hence the problem of primality testing easily reduces to* $\text{MF}(\cup, \cap, ^-, \times)$.

Example 2.2 *(Goldbach's conjecture) Consider the circuit* GOLDBACH *defined as* $(\text{GE2} \times 2) \cap (\overline{\text{PRIMES} + \text{PRIMES}})$. *Then* I(GOLDBACH) *is empty iff every even number greater than 2 is expressible as a sum of two primes. Hence Goldbach's conjecture holds iff "*$0 \in \overline{0 \times \text{GOLDBACH}}$*" is a positive instance of* $\text{MC}(\cup, \cap, ^-, +, \times)$.

If not otherwise stated, the hardness results in this paper are in terms of many-one logspace reducibility. We assume any circuit and formula encoding in which the gates are sorted topologically and in which immediate predecessors are readily available (say in AC^0). Viewed as graphs, circuits and formulas are not necessarily connected. Numbers are encoded in binary notation.

The following results are known from the literature:

Theorem 2.3 *1.* [SM73] *The problem* $\text{MF}(\cup, +)$ *is* NP*-complete.*

2. [SM73] *The problems* $\text{MF}(\cup, \cap, ^-, +)$ *and* $\text{MF}(\cup, ^-, +)$ *are* PSPACE*-complete.*

3. [Wa84] *The problem* $\text{MC}(\cup, +)$ *is in* PSPACE.

4. [Ya00] *The problem* $\text{MC}(\cup, +, \times)$ *is* PSPACE*-complete.*

In [Wa84] it is shown that the problem MC($\cup, +$) restricted to $(\cup, +)$-circuits for which every $+$-gate has at least one input gate as predecessor is NP-complete.

By De Morgan's laws we have

Proposition 2.4 *For every $\mathcal{O} \subseteq \{+, \times\}$,*

1. $\mathrm{MC}(\{\cup, \cap, ^- \} \cup \mathcal{O}) \equiv_{\mathrm{m}}^{\log} \mathrm{MC}(\{\cup, ^- \} \cup \mathcal{O}) \equiv_{\mathrm{m}}^{\log} \mathrm{MC}(\{\cap, ^- \} \cup \mathcal{O})$.
2. $\mathrm{MF}(\{\cup, \cap, ^- \} \cup \mathcal{O}) \equiv_{\mathrm{m}}^{\log} \mathrm{MF}(\{\cup, ^- \} \cup \mathcal{O}) \equiv_{\mathrm{m}}^{\log} \mathrm{MF}(\{\cap, ^- \} \cup \mathcal{O})$.

Hence we can omit $\mathrm{MC}(\{\cup, ^- \} \cup \mathcal{O})$, $\mathrm{MC}(\{\cap, ^- \} \cup \mathcal{O})$, $\mathrm{MF}(\{\cup, ^- \} \cup \mathcal{O})$, and $\mathrm{MF}(\{\cap, ^- \} \cup \mathcal{O})$ from our exhaustive study.

3 Multiplication versus Addition

In this section we will establish a relationship between the complexity of the membership problems for $(\mathcal{O} \cup \{\times\})$-circuits and $(\mathcal{O} \cup \{+\})$-circuits, for $\mathcal{O} \subseteq \{\cup, \cap, ^- \}$. To this end we need the following problem. Let $\gcd(a, b)$ be the greatest common divisor of the numbers $a, b \geq 1$.

Gcd-Free Basis (GFB)
Given: Numbers $a_1, a_2, \ldots, a_n \geq 1$.
Compute: Numbers $m \geq 1$, $q_1, \ldots, q_m \geq 2$, and $e_{11}, \ldots, e_{nm} \geq 0$ such that
$\gcd(q_i, q_j) = 1$ for $i \neq j$ and $a_i = \prod_{j=1}^{m} q_j^{e_{ij}}$ for $i = 1, \ldots, n$.

Despite the fact that factoring may not be possible in polynomial time, the following is known (see [BS96]):

Proposition 3.1 *Gcd-Free Basis can be computed in polynomial time.*

As an auxiliary tool we need the generalized membership problems $\mathrm{MC}^*(\mathcal{O})$ and $\mathrm{MF}^*(\mathcal{O})$ for arithmetic circuit and formulae, resp., with addition. These problems deal with elements of $\mathbb{N}^m \cup \{\infty\}$, for an $m \geq 1$ prescribed on input, where the addition on m-tuples is defined componentwise and $a + \infty = \infty + a = \infty + \infty = \infty$ for every $a \in \mathbb{N}^m$. Note that polynomial space many-one reducibility is understood to be performed by polynomial space computable polynomially bounded functions.

Lemma 3.2
1. *For $\mathcal{O} \subseteq \{\cup, \cap\}$, the problem $\mathrm{MC}(\mathcal{O} \cup \{\times\})$ is polynomial time many-one reducible to the problem $\mathrm{MC}^*(\mathcal{O} \cup \{+\})$.*
2. *For $\mathcal{O} \subseteq \{\cup, \cap\}$, the problem $\mathrm{MF}(\mathcal{O} \cup \{\times\})$ is polynomial time many-one reducible to the problem $\mathrm{MF}^*(\mathcal{O} \cup \{+\})$.*
3. *For $\mathcal{O} \subseteq \{\cup, \cap, ^- \}$, the problem $\mathrm{MC}(\mathcal{O} \cup \{\times\})$ is polynomial space many-one reducible to the problem $\mathrm{MC}^*(\mathcal{O} \cup \{+\})$.*
4. *For $\mathcal{O} \subseteq \{\cup, \cap, ^- \}$, the problem $\mathrm{MF}(\mathcal{O} \cup \{\times\})$ is polynomial space many-one reducible to the problem $\mathrm{MF}^*(\mathcal{O} \cup \{+\})$.*

In each of the 4 cases above, if we make the added hypotheses that if the prime number decompositions of the inputs and of the "target number" b are known, then the complexity of the reduction drops down to logspace many-one.

Proof. 1. Let $\mathcal{O} \subseteq \{\cup, \cap\}$, let C be a $(\mathcal{O} \cup \{\times\})$-circuit with the input gates $u_1, \ldots, u_s$, and let $b \in \mathbb{N}$. Observe that the absence of $+$ in C entails that any number in $I(C)$ is expressible as a monomial in the inputs. Compute in polynomial time (Proposition 3.1) numbers $q_1, \ldots, q_m \geq 2$ and $e_{11}, \ldots, e_{sm}, e_1, \ldots, e_m \geq 1$ such that $\gcd(q_i, q_j) = 1$ for $i \neq j$, $\alpha(u_i) = \prod_{j=1}^{m} q_j^{e_{ij}}$ for $i = 1, \ldots, s$ such that $\alpha(u_i) > 0$ and $b = \prod_{j=1}^{m} q_j^{e_j}$ if $b > 0$. Let $M =_{\text{def}} \{ \prod_{j=1}^{m} q_j^{d_j} : d_1, \ldots, d_m \geq 0 \} \subseteq \mathbb{N}$ and let $\sigma : M \cup \{0\} \to \mathbb{N}^m \cup \{\infty\}$ be defined by $\sigma(\prod_{j=1}^{m} q_j^{d_j}) =_{\text{def}} (d_1, \ldots, d_m)$ and $\sigma(0) = \infty$. Obviously, σ is a monoid isomorphism between $(M \cup \{0\}, \times)$ and $(\mathbb{N}^m \cup \{\infty\}, +)$. Because $M \cup \{0\}$ is closed under $\times$, the set of numbers computed by any gate in C is included in $M \cup \{0\}$. Furthermore, the following holds for any $S_1, S_2 \subseteq M \cup \{0\}$:

$$\begin{aligned} \sigma(S_1 \times S_2) &= \sigma(S_1) + \sigma(S_2), \\ \sigma(S_1 \cup S_2) &= \sigma(S_1) \cup \sigma(S_2), \\ \sigma(S_1 \cap S_2) &= \sigma(S_1) \cap \sigma(S_2). \end{aligned}$$

The reduction therefore consists of converting C into a $(\mathcal{O} \cup \{+\})$-circuit C' which has the same structure as C where a $\times$-gate in C becomes a $+$-gate in C' and an input gate u_i gets label $\sigma(\alpha(u_i))$. An induction using the three identities above shows that for all $a \in M \cup \{0\}$ the following holds: $a \in I(v)$ in $C \Leftrightarrow \sigma(a) \in I(v)$ in C'. This concludes the proof because $b \in M \cup \{0\}$.

2. Same as above because the construction preserves the circuit structure.

3. and 4. If we have complementation then it is no longer true that every number computed within C has a decomposition into $q_1, \ldots, q_m$. To salvage the above construction, the isomorphism σ must therefore be extended to convey information about $\mathbb{N} \setminus M$. A slick way to do this is to begin from the *full* prime decomposition of the numbers $u_1, \ldots, u_s, b$ and to trade the former isomorphism for a homomorphism. Indeed let $q_1, \ldots, q_m, q_{m+1}, q_{m+2}, \ldots$ be the sequence of all primes in natural order, and let $q_1, \ldots, q_m$ exhaust at least all the distinct prime divisors of $\alpha(u_1), \ldots, \alpha(u_s), b$. For every number $\prod_{j=1}^{\infty} q_j^{d_j} \in \mathbb{N} \setminus \{0\}$ define $\sigma(\prod_{j=1}^{\infty} q_j^{d_j}) =_{\text{def}} (d_1, \ldots, d_m, \sum_{j>m} d_j)$, and define $\sigma(0) =_{\text{def}} \infty$. Let $M =_{\text{def}} \{ \prod_{j=1}^{m} q_j^{d_j} : d_1, \ldots, d_m \geq 0 \} \subseteq \mathbb{N}$ and $K_i =_{\text{def}} \{\prod_{j>m} q_j^{d_j} : \sum_{j>m} d_j = i\}$ for $i \geq 0$. Because the full prime decomposition was used, σ is a well-defined monoid homomorphism from $(\mathbb{N}, \times)$ onto $(\mathbb{N}^{m+1} \cup \{\infty\}, +)$, where $\sigma(0) = \infty$ and $\sigma(M \times K_i) = (\mathbb{N}^m \otimes \{i\})$ for $i \geq 0$ ($\otimes$ denotes direct product). Notice that $M = M \times K_0$ and that σ is one-one on this part, so that the following holds for any $S_1, S_2 \subseteq M$:

$$\begin{aligned} \sigma(S_1 \times S_2) &= \sigma(S_1) + \sigma(S_2), \\ \sigma(S_1 \cup S_2) &= \sigma(S_1) \cup \sigma(S_2), \\ \sigma(M \setminus S_1) &= \sigma(M) \setminus \sigma(S_1). \end{aligned}$$

The reduction then again consists of converting C into a $(\mathcal{O} \cup \{+\})$-circuit C' having the same structure as C where a $\times$-gate in C becomes a $+$-gate in C' and an input gate u_i gets label $\sigma(\alpha(u_i))$. Let $e_i =_{\text{def}} (0, \ldots, 0, i) \in \mathbb{N}^{m+1}$ for $i \geq 0$.

Claim. For every gate v in C there exist $T \subseteq \{0\}$ and $S_0, S_1, S_2, \cdots \subseteq M$ such that $I(v) = T \cup \bigcup_{i \geq 0}(S_i \times K_i)$ in C and $I(v) = \sigma(T) \cup \bigcup_{i \geq 0}(\sigma(S_i) + e_i)$ in C'.

Proof of the Claim. The proof is by induction on the structure of C. Without loss of generality assume that C has no $\cap$-gates.

Let v be an input gate. If $\alpha(v) = 0$ then $I(v) = \{0\} \cup \bigcup_{i \geq 0}(\emptyset \times K_i)$ in C and $I(v) = \{\infty\} = \sigma(\{0\}) \cup \bigcup_{i \geq 0}(\sigma(\emptyset) + e_i)$ in C'.

For a $\times$-gate or a $\cup$-gate v let v_1 and v_2 be the predecessor gates, and for a $^-$-gate v let v_1 be the predecessor gate. By the induction hypothesis there are $T_1, T_2 \subseteq \{0\}$ and $S_0^1, S_1^1, S_2^1, \ldots S_0^2, S_1^2, S_2^2, \cdots \subseteq M$ such that $I(v_r) = T_r \cup \bigcup_{i\geq 0}(S_i^r \times K_i)$ in C and $I(v_r) = \sigma(T_r) \cup \bigcup_{i\geq 0}(\sigma(S_i^r) + e_i)$ in C' for $r = 1, 2$.

For a $\cup$-gate v we obtain in C:

$$\begin{aligned} I(v) &= I(v_1) \cup I(v_2) = (T_1 \cup \textstyle\bigcup_{i\geq 0}(S_i^1 \times K_i)) \cup (T_2 \cup \bigcup_{i\geq 0}(S_i^2 \times K_i)) \\ &= (T_1 \cup T_2) \cup \textstyle\bigcup_{i\geq 0}((S_i^1 \cup S_i^2) \times K_i)) \end{aligned}$$

and in C':

$$\begin{aligned} I(v) &= I(v_1) \cup I(v_2) = (\sigma(T_1) \cup \textstyle\bigcup_{i\geq 0}(\sigma(S_i^1) + e_i)) \cup (\sigma(T_2) \cup \bigcup_{i\geq 0}(\sigma(S_i^2) + e_i)) \\ &= \sigma(T_1 \cup T_2) \cup \textstyle\bigcup_{i\geq 0}(\sigma(S_i^1 \cup S_i^2) + e_i). \end{aligned}$$

For a $^-$-gate v we obtain in C:

$$I(v) = \overline{I(v_1)} = \overline{T_1 \cup \textstyle\bigcup_{i\geq 0}(S_i^1 \times K_i)} = (\{0\} \setminus T_1) \cup \textstyle\bigcup_{i\geq 0}((M \setminus S_i^1) \times K_i)$$

and in C':

$$\begin{aligned} I(v) &= \overline{I(v_1)} = \overline{\sigma(T_1) \cup \textstyle\bigcup_{i\geq 0}(\sigma(S_i^1) + e_i)} \\ &= (\{\infty\} \setminus \sigma(T_1)) \cup \textstyle\bigcup_{i\geq 0}((\mathbb{N}^m \otimes \{i\}) \setminus (\sigma(S_i^1) + e_i)) \\ &= \sigma(\{0\} \setminus T_1) \cup \textstyle\bigcup_{i\geq 0}(((\mathbb{N}^m \otimes \{0\}) \setminus \sigma(S_i^1)) + e_i) \\ &= \sigma(\{0\} \setminus T_1) \cup \textstyle\bigcup_{i\geq 0}(\sigma(M \setminus S_i^1) + e_i). \end{aligned}$$

For a $\times$-gate v we obtain in C:

$$\begin{aligned} I(v) &= I(v_1) \times I(v_2) = (T_1 \cup \textstyle\bigcup_{i\geq 0}(S_i^1 \times K_i)) \times (T_2 \cup \bigcup_{i\geq 0}(S_i^2 \times K_i)) \\ &= T \cup \textstyle\bigcup_{i\geq 0}((\bigcup_{j=0}^{i}(S_j^1 \times S_{i-j}^2)) \times K_i) \end{aligned}$$

where $T = ((T_1 \times T_2) \cup (T_1 \times \bigcup_{i\geq 0}(S_i^2 \times K_i)) \cup (T_2 \times \bigcup_{i\geq 0}(S_i^1 \times K_i)))$, and in C':

$$\begin{aligned} I(v) &= I(v_1) + I(v_2) = (\sigma(T_1) \cup \textstyle\bigcup_{i\geq 0}(\sigma(S_i^1) + e_i)) + (\sigma(T_2) \cup \bigcup_{i\geq 0}(\sigma(S_i^2) + e_i)) \\ &= ((\sigma(T_1) + \sigma(T_2)) \cup (\sigma(T_1) + \textstyle\bigcup_{i\geq 0}(\sigma(S_i^2) + e_i)) \cup (\sigma(T_2) + \bigcup_{i\geq 0}(\sigma(S_i^1) + e_i))) \\ &\qquad \cup \textstyle\bigcup_{i\geq 0}(\bigcup_{j=0}^{i}(\sigma(S_j^1) + \sigma(S_{i-j}^2) + e_i)) \\ &= \sigma(T) \cup \textstyle\bigcup_{i\geq 0}\bigcup_{j=0}^{i}(\sigma(S_j^1 \times S_{i-j}^2) + e_i) = \sigma(T) \cup \bigcup_{i\geq 0}((\bigcup_{j=0}^{i}\sigma(S_j^1 \times S_{i-j}^2)) + e_i) \\ &= \sigma(T) \cup \textstyle\bigcup_{i\geq 0}(\sigma(\bigcup_{j=0}^{i}(S_j^1 \times S_{i-j}^2)) + e_i). \end{aligned}$$

This completes the proof of the claim.

Now we prove that there holds $a \in I(v)$ in $C \Leftrightarrow \sigma(a) \in I(v)$ in C' for every gate v and every $a \in \mathbb{N}$. By our claim there exist $T \subseteq \{0\}$ and $S_0, S_1, S_2, \cdots \subseteq M$ such that $I(v) = T \cup \bigcup_{i\geq 0}(S_i \times K_i)$ in C and $I(v) = \sigma(T) \cup \bigcup_{i\geq 0}(\sigma(S_i) + e_i)$ in C'. We consider two cases. In the case $a = 0$ we obtain:

$$0 \in I(v) \text{ in } C \Leftrightarrow T = \{0\} \Leftrightarrow \sigma(T) = \{\infty\} \Leftrightarrow \infty \in I(v) \text{ in } C' \Leftrightarrow \sigma(0) \in I(v) \text{ in } C'.$$

In the case $a = \prod_{j\geq 1} q_j^{d_j}$ we obtain (define $d =_{\text{def}} \sum_{j>m} d_j$):

$$\begin{aligned} \textstyle\prod_{j\geq 1} q_j^{d_j} \in I(v) \text{ in } C &\Leftrightarrow \textstyle\prod_{j\geq 1} q_j^{d_j} \in \bigcup_{i\geq 0}(S_i \times K_i) \Leftrightarrow \prod_{j\geq 1} q_j^{d_j} \in S_d \times K_d \\ &\Leftrightarrow \textstyle\prod_{j=1}^{m} q_j^{d_j} \in S_d \Leftrightarrow \sigma(\prod_{j=1}^{m} q_j^{d_j}) \in \sigma(S_d) \\ &\Leftrightarrow \sigma(\textstyle\prod_{j=1}^{m} q_j^{d_j}) + e_d \in \sigma(S_d) + e_d \Leftrightarrow \sigma(\prod_{j\geq 1} q_j^{d_j}) \in \sigma(S_d) + e_d \\ &\Leftrightarrow \sigma(\textstyle\prod_{j\geq 1} q_j^{d_j}) \in \bigcup_{i\geq 0}(\sigma(S_i) + e_i) \Leftrightarrow \sigma(\prod_{j\geq 1} q_j^{d_j}) \in I(v) \text{ in } C'. \end{aligned}$$

Applying the equivalence $a \in I(v)$ in $C \Leftrightarrow \sigma(a) \in I(v)$ in C' to the output gate yields the desired reduction. The polynomial space is needed to perform the prime decomposition (a pos-

sibly weaker reducibility, like a many-one polynomial time reduction with an NP oracle, would suffice). □

In some cases the generalized membership problems used above are logspace equivalent to their standard versions:

Lemma 3.3 *Let* $\{\cup\} \subseteq \mathcal{O} \subseteq \{\cup, \cap\}$.

1. $\mathrm{MC}^*(\mathcal{O} \cup \{+\}) \equiv_{\mathrm{m}}^{\log} \mathrm{MC}(\mathcal{O} \cup \{+\})$.
2. $\mathrm{MF}^*(\mathcal{O} \cup \{+\}) \equiv_{\mathrm{m}}^{\log} \mathrm{MF}(\mathcal{O} \cup \{+\})$.

Proof 1. Let C be a $(\mathcal{O} \cup \{+\})$-circuit processing m-tuples of natural numbers. For every gate v of C and every $(a_1, \ldots, a_m) \in I(v)$ we have $N =_{\mathrm{def}} 2^{|C|} > a_i$ for $i = 1, \ldots, m$. We consider an $(\mathcal{O} \cup \{+\})$-circuit C' which has the same structure as C but an input with label $(a_1, \ldots, a_m)$ in C gets label $\sum_{i=1}^{m} a_i \cdot N^{i-1}$ in C', and an input u with label ∞ is fed with an $(\mathcal{O} \cup \{+\})$-formula $F(C)$ such that $I(u) = \{N^m, N^m+1, N^m+2, \ldots, N^m+N^{m+1}\}$ in C'. (For example, define $G(0) =_{\mathrm{def}} (0 \cup 1)$, $G(n+1) =_{\mathrm{def}} (0 \cup 2^n) + G(n)$, and $F(C) =_{\mathrm{def}} (N^m + G((m+1) \cdot |C|))$.) Now we obtain for every gate v:

$$(a_1, \ldots, a_m) \in I(v) \text{ in } C \Leftrightarrow \textstyle\sum_{i=1}^{m} a_i \cdot N^{i-1} \in I(v) \text{ in } C'$$

where $(a_1, \ldots, a_m) \in I(v)$ in C implies $\sum_{i=1}^{m} a_i \cdot N^{i-1} < N^m$.

Moreover, if v is a level-i-gate in C (input gates being level-0-gates) then (by induction on i):

$$\infty \in I(v) \text{ in } C \Leftrightarrow [2^i \cdot N^m, N^m + N^{m+1}] \subseteq I(v) \text{ in } C'.$$

Hence, we have for every gate v:

$$\infty \in I(v) \text{ in } C \Leftrightarrow N^m + N^{m+1} \in I(v) \text{ in } C'.$$

This yields one of the desired reductions. The other is obvious.

2. Here we proceed in the same way but we have $N =_{\mathrm{def}} |C| > a$ for every $a \in I(v)$. □

4 NP-Complete Membership Problems

Lemma 4.1 *The problem* $\mathrm{MF}(\cup, \cap, +, \times)$ *is in* NP.

Proof Let $F = (G, E, g_F, \alpha)$ be a $\{\cup, \cap, +, \times\}$-formula, and let $b \in \mathbb{N}$. Observe that $M =_{\mathrm{def}} \prod_{v \text{ input gate of } F} (\alpha(v) + 1)$ is an upper bound on every number in $\bigcup_{g \text{ gate of } F} I(g)$. The NP-algorithm just works as follows: Guess nondeterministically a number $\leq M$ to every gate of F. Then check whether these numbers really prove that b is in $I(g_F)$. □

The following is a nontrivial improvement over the known PSPACE upper bound for $\mathrm{MC}(\cup, +)$:

Lemma 4.2 *The problems* $\mathrm{MC}(\cup, +)$ *and* $\mathrm{MC}(\cup, \times)$ *are in* NP.

Proof. Let $C = (G, E, g_C, \alpha)$ be a $\{\cup, +\}$-circuit such that g_C is the only gate of C with outdegree 0, and let T_C be the result of unfolding C into a tree. A subtree T of T_C is called *computation tree* of C iff

- the output gate of T_C is in T,

- both predecessors of a +-gate of T are in T, and
- exactly one predecessor of a $\cup$-gate of T is in T.

Hence T describes one of the many ways to compute a number from $I(C)$.

A gate g in C corresponds to several copies of it in T_C (and hence also in a computation tree T of C). Let $\beta_{C,T}(g)$ be the number of copies of g in T. In the same way, an edge in C corresponds to several copies of it in T_C (and hence also in a computation tree T of C). Let $\beta_{C,T}(e)$ be the number of copies of e in T.

Defining $\mathrm{s}(C,T) =_{\mathrm{def}} \sum_{\text{input gate } g \text{ of } C} \alpha(g)\cdot\beta_{C,T}(g)$

we obtain immediately $I(C) = \{\mathrm{s}(C,T) : T \text{ is a computation tree of } C\}$.

A function $\beta : G \cup E \to \mathbb{N}$ is a *valuation function* of the C if the following holds:
- $\beta(g_C) = 1$,
- if g is a +-gate with the incoming edges e_1 and e_2 then $\beta(g) = \beta(e_1) = \beta(e_2)$,
- if g is a $\cup$-gate with the incoming edges e_1 and e_2 then $\beta(g) = \beta(e_1)+\beta(e_2)$,
- if g is a gate with the outgoing edges $e_1, \ldots e_k$ then $\beta(g) = \beta(e_1)+\ldots+\beta(e_k)$.

For a valuation function β of C define $\mathrm{s}(C,\beta) =_{\mathrm{def}} \sum_{\text{input gate } g \text{ of } C} \alpha(g)\cdot\beta(g)$.

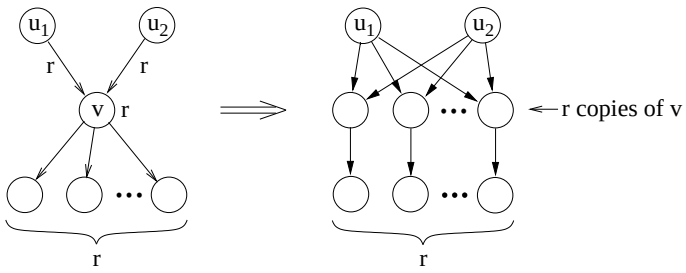
Claim 1: If T is a computation tree of C then there exists a valuation function β of C such that $\mathrm{s}(C,\beta) = \mathrm{s}(C,T)$.

Proof. This is easy because $\beta_{C,T}$ has the desired properties.

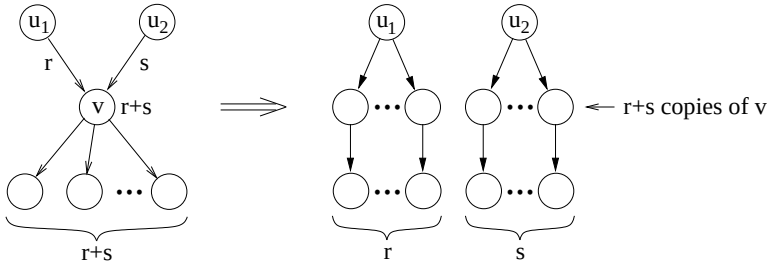
Claim 2: If β is a valuation function of C then there exists a computation tree T of C such that $\mathrm{s}(C,T) = \mathrm{s}(C,\beta)$.

Proof. For a valuation function β of C there can be many computation trees T of C such that $\beta_{C,T} = \beta$ and hence $\mathrm{s}(C,T) = \mathrm{s}(C,\beta)$. We construct one of them by applying the following replacement rules to the gates v of C (starting with the predecessors of the output gate and going upward). The labels at the gates and edges represent the β-values.

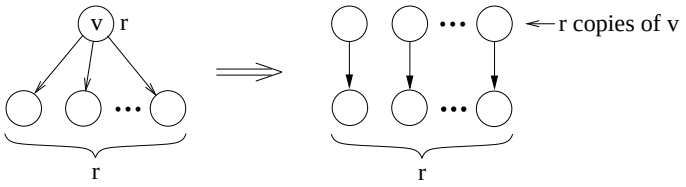
If v is a +-gate then apply:



If v is a $\cup$-gate then apply:



If v is an input gate then apply:



After the application of a rule to a gate v the following holds:

- the definition of a computation tree applies to the copies of v,
- the number of copies of v is $\beta(v)$, and
- for every incoming edge e to v, the number of copies of e is $\beta(e)$.

In such a way we obtain $\beta_{C,T} = \beta$ and hence $\mathrm{s}(C,T) = \mathrm{s}(C,\beta)$. This completes the proof of Claim 2.

Now we obtain $I(C) = \{\mathrm{s}(C,\beta) : \beta \text{ is a valuation function of } C\}$, and hence $a \in I(C) \Leftrightarrow \exists\beta(\beta \text{ is a valuation function of } C \text{ and } \mathrm{s}(C,\beta) = a)$. However, the latter property is in NP. This completes the proof of $\mathrm{MC}(\cup,+) \in \mathrm{NP}$. From this, Lemma 3.2, and Lemma 3.3 we obtain $\mathrm{MC}(\cup,\times) \in \mathrm{NP}$. □

Lemma 4.3 $\mathrm{MF}(\cup,\times)$ *is* NP*-hard.*

Proof. We prove the NP-hardness of $\mathrm{MF}(\cup,\times)$ by showing 3-SAT $\leq_{\mathrm{m}}^{\log}$ $\mathrm{MF}(\cup,\times)$. Let $H(x_1,\ldots,x_m) = \bigwedge_{j=1}^{n} H_j$ where every H_j is a clause with three literals. Further, for $j \geq 1$, let p_j be the j-th prime. For $i = 1,\ldots,m$ define $a_i =_{\mathrm{def}} \prod_{x_i \text{ in } H_j} p_j$ and $b_i =_{\mathrm{def}} \prod_{\overline{x_i} \text{ in } H_j} p_j$. Obviously, $H \in$ 3-SAT $\Leftrightarrow \prod_{j=1}^{n} p_j^3 \in I(\prod_{i=1}^{m}(a_i \cup b_i)\cdot\prod_{j=1}^{n}(1 \cup p_j \cup p_j^2))$. □

As an immediate consequence of the preceding lemmas and Theorem 2.3.1 we obtain:

Theorem 4.4 *The problems* $\mathrm{MF}(\cup,\cap,+,\times)$, $\mathrm{MF}(\cup,\cap,+)$, $\mathrm{MF}(\cup,\cap,\times)$, $\mathrm{MF}(\cup,+,\times)$, $\mathrm{MF}(\cup,\times)$, $\mathrm{MC}(\cup,+)$, *and* $\mathrm{MC}(\cup,\times)$ *are* NP*-complete.*

5 PSPACE-Complete Membership Problems

Lemma 5.1 *The problems* $\mathrm{MF}(\cup,\cap,^{-},\times)$, $\mathrm{MC}(\cup,\cap,\times)$, *and* $\mathrm{MC}(\cup,\cap,+)$ *are* PSPACE*-hard.*

Proof. We present a logspace many-one reduction from the PSPACE-complete quantified boolean 3-CNF formula problem to $\mathrm{MF}(\cup,\cap,^{-},\times)$ and $\mathrm{MC}(\cup,\cap,\times)$. Since in these constructions we use only numbers whose prime number decompositions are known, the latter reduction can be converted by Lemma 3.2 and Lemma 3.3 into a logspace many-one reduction to $\mathrm{MC}(\cup,\cap,+)$.

Let $Q_1x_1\ldots Q_mx_mH(x_1,\ldots,x_m)$ be a quantified boolean 3-CNF formula such that $H = \bigwedge_{j=1}^{n} H_j$, the formulae $H_1,\ldots H_n$ are clauses with three literals each, and $Q_1\ldots Q_m \in \{\exists,\forall\}$.

For $k = m, m-1,\ldots 1, 0$ we construct a $(\cup,\cap,\times)$-circuit (a $(\cup,\cap,^{-},\times)$-formula, resp.) C_k such that for all $\alpha_1,\ldots,\alpha_k \in \{0,1\}$ the following holds:

$$Q_{k+1}x_{k+1}\ldots Q_mx_mH(\alpha_1,\ldots,\alpha_k,x_{k+1},\ldots,x_m) \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k} p_{n+i}^{\alpha_i} \cdot \prod_{i=k+1}^{m} p_{n+i} \in I(C_k) \quad (*)$$

where p_i is the i-th prime.

Then, for $k = 0$, we obtain $Q_1 x_1 \dots Q_m x_m H(x_1, \dots, x_m) \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{m} p_{n+i} \in I(C_0)$, the desired reduction.

For $k = m$ we define $a_i =_{\text{def}} p_{n+i} \cdot \prod_{x_i \text{ in } H_j} p_j$ and $b_i =_{\text{def}} \prod_{\overline{x_i} \text{ in } H_j} p_j$ for $i = 1, \dots, m$, and $C_m =_{\text{def}} \prod_{i=1}^{m}(a_i \cup b_i) \cdot \prod_{j=1}^{n}(1 \cup p_j \cup p_j^2)$. It is easy to see that $H(\alpha_1, \dots, \alpha_m) \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{m} p_{n+i}^{\alpha_i} \in I(C_m)$.

Now, for the step from k to $k-1$, assume (*). In the case $Q_k = \exists$ we have:

$$\begin{aligned}
&\exists x_k Q_{k+1} x_{k+1} \dots Q_m x_m H(\alpha_1, \dots, \alpha_{k-1}, x_k, x_{k+1}, \dots, x_m) \Leftrightarrow \\
&\quad \Leftrightarrow Q_{k+1} x_{k+1} \dots Q_m x_m H(\alpha_1, \dots, \alpha_{k-1}, 0, x_{k+1}, \dots, x_m) \vee \\
&\qquad\qquad \vee Q_{k+1} x_{k+1} \dots Q_m x_m H(\alpha_1, \dots, \alpha_{k-1}, 1, x_{k+1}, \dots, x_m) \\
&\quad \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k+1}^{m} p_{n+i} \in I(C_k) \vee \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I(C_k) \\
&\quad \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I((C_k \times p_{n+k}) \cup C_k) = I(C_{k-1})
\end{aligned}$$

where $C_{k-1} =_{\text{def}} C_k \times (p_{n+k} \cup 1)$. Since C_k appears only once in the definition of C_{k-1}, this works for the circuit case as well as for the formula case.

In the case $Q_k = \forall$ the reasoning becomes:

$$\begin{aligned}
&\forall x_k Q_{k+1} x_{k+1} \dots Q_m x_m H(\alpha_1, \dots, \alpha_{k-1}, x_k, x_{k+1}, \dots, x_m) \Leftrightarrow \\
&\quad \Leftrightarrow Q_{k+1} x_{k+1} \dots Q_m x_m H(\alpha_1, \dots, \alpha_{k-1}, 0, x_{k+1}, \dots, x_m) \wedge \\
&\qquad\qquad \wedge Q_{k+1} x_{k+1} \dots Q_m x_m H(\alpha_1, \dots, \alpha_{k-1}, 1, x_{k+1}, \dots, x_m) \\
&\quad \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k+1}^{m} p_{n+i} \in I(C_k) \wedge \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I(C_k) \\
&\quad \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I((C_k \times p_{n+k}) \cap C_k) = I(C_{k-1})
\end{aligned}$$

where $C_{k-1} =_{\text{def}} (C_k \times p_{n+k}) \cap C_k$. Since C_k appears twice in the definition of C_{k-1} this works only for the circuit case. For the formula case we continue as follows:

$$\begin{aligned}
&\quad \Leftrightarrow \neg \Big(\prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k+1}^{m} p_{n+i} \in I(\overline{C_k}) \vee \\
&\qquad\qquad \vee \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I(\overline{C_k}) \Big) \\
&\quad \Leftrightarrow \neg \Big(\prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I(\overline{C_k} \times (p_{n+k} \cup 1)) \Big) \\
&\quad \Leftrightarrow \prod_{j=1}^{n} p_j^3 \cdot \prod_{i=1}^{k-1} p_{n+i}^{\alpha_i} \cdot \prod_{i=k}^{m} p_{n+i} \in I(\overline{\overline{C_k} \times (p_{n+k} \cup 1)}) = I(C_{k-1})
\end{aligned}$$

where $C_{k-1} =_{\text{def}} \overline{\overline{C_k} \times (p_{n+k} \cup 1)}$. □

Lemma 5.2 *For every gate g of a generalized $(\cup, \cap, ^-, +)$-circuit, if $I(g) \neq \emptyset$ then $I(g) \cap (\{0, 1, \dots, 2^{|C|+1}\}^m \cup \{\infty\}) \neq \emptyset$.*

Proof. Let $C = (G, E, g_C, \alpha)$ be a generalized $(\cup, \cap, ^-, +)$-circuit over $\mathbb{N}^m \cup \{\infty\}$. By De Morgan's law we can assume that C has no $\cap$-gates. For the gates g of C we define inductively $\sigma(g) =_{\text{def}} \max\{a_i : i = 1, \dots, m\}$ for an input gate g with $\alpha(g) = (a_1, \dots, a_m)$, $\sigma(g) =_{\text{def}} 0$ for an input gate g with $\alpha(g) = \infty$, $\sigma(g) =_{\text{def}} \sigma(g_1) + \sigma(g_2) + 1$ for $\cup$-gates and $+$-gates g with predecessors g_1 and g_2, and $\sigma(g) =_{\text{def}} \sigma(g_1)$ for $^-$-gates g with predecessor g_1. Obviously we have $\sigma(g) \leq 2^{|C|}$ for every gate g. Let $[a] =_{\text{def}} \{0, 1, \dots, a\}^m \cup \{\infty\}$ for $a \in \mathbb{N}$. The lemma follows directly from

- $I(g) \subseteq [\sigma(g)]$ or $\overline{[\sigma(g)]} \subseteq I(g)$ for every gate g

which we will prove by induction.
1. For an input gate g we have $I(g) \subseteq [\sigma(g)]$.
2. Let g be a $\cup$-gate with predecessors g_1 and g_2.

Case 1: If $I(g_1) \subseteq [\sigma(g_1)]$ and $I(g_2) \subseteq [\sigma(g_2)]$ then
$$I(g) = I(g_1) \cup I(g_2) \subseteq [\max(\sigma(g_1), \sigma(g_2))] \subseteq [\sigma(g)].$$
Case 2: If $\overline{[\sigma(g_1)]} \subseteq I(g_1)$ or $\overline{[\sigma(g_2)]} \subseteq I(g_2)$ then
$$\overline{[\sigma(g)]} \subseteq \overline{[\max(\sigma(g_1), \sigma(g_2))]} \subseteq I(g_1) \cup I(g_2) = I(g).$$
3. Let g be a $^-$-gate with predecessor g_1.

Case 1: If $I(g_1) \subseteq [\sigma(g_1)]$ then $\overline{[\sigma(g)]} \subseteq \overline{[\sigma(g_1)]} \subseteq \overline{I(g_1)} = I(g)$.

Case 2: If $\overline{[\sigma(g_1)]} \subseteq I(g_1)$ then $I(g) = \overline{I(g_1)} \subseteq [\sigma(g_1)] \subseteq [\sigma(g)]$.

4. Let g be a $+$-gate with predecessors g_1 and g_2.

Case 1: If $I(g_1) \subseteq [\sigma(g_1)]$ and $I(g_2) \subseteq [\sigma(g_2)]$ then
$$I(g) = I(g_1) + I(g_2) \subseteq [\sigma(g_1)] + [\sigma(g_2)] = [\sigma(g_1) + \sigma(g_2)] \subseteq [\sigma(g)]$$
Case 2: If $I(g_1) = \emptyset$ and $\overline{[\sigma(g_2)]} \subseteq I(g_2)$ then
$$I(g) = I(g_1) + I(g_2) = \emptyset + I(g_2) = \emptyset \subseteq [\sigma(g)].$$
Case 3: If $I(g_1) = \{\infty\}$ and $\overline{[\sigma(g_2)]} \subseteq I(g_2)$ then
$$I(g) = I(g_1) + I(g_2) = \{\infty\} + I(g_2) = \{\infty\} \subseteq [\sigma(g)].$$
Case 4: If $(a_1, \ldots, a_m) \in I(g_1) \subseteq [\sigma(g_1)]$ and $\overline{[\sigma(g_2)]} \subseteq I(g_2)$ then
$$\begin{aligned} I(g) = \; & I(g_1) + I(g_2) \supseteq \{(a_1, \ldots, a_m)\} + \overline{[\sigma(g_2)]} \\ & \supseteq \{(\sigma(g_1), \ldots, \sigma(g_1))\} + \overline{[\sigma(g_2)]} \supseteq \overline{[\sigma(g_1) + \sigma(g_2)]} \supseteq \overline{[\sigma(g)]}. \end{aligned}$$
Case 5: If $\overline{[\sigma(g_1)]} \subseteq I(g_1)$ and $\overline{[\sigma(g_2)]} \subseteq I(g_2)$ then
$$I(g) = I(g_1) + I(g_2) \supseteq \overline{[\sigma(g_1)]} + \overline{[\sigma(g_2)]} \supseteq \overline{[\sigma(g_1) + \sigma(g_2) + 1]} = \overline{[\sigma(g)]}. \qquad \square$$

Lemma 5.3 *The problem* $\mathrm{MC}^*(\cup, \cap, ^-, +)$ *is in* PSPACE.

Proof. For a generalized $\{\cup, \cap, ^-, +\}$-circuit $C = (G, E, g_C, \alpha)$ and $b \in \mathbb{N}^m \cup \{\infty\}$, the following alternating polynomial time algorithm defines on each of its accepting subtrees a proof that $b \in I(F)$. Moreover, in every step on every path the algorithm defines $g \in G$, $a \in \mathbb{N}^m \cup \{\infty\}$, and $\sigma(g) \in \{0, 1\}$. Afterwards, i.e. in the computation tree above this configuration, the algorithm checks whether $a \in I(g)$ if $\sigma = 1$ and $a \notin I(g)$ if $\sigma = 0$.

For a gate g of indegree 2 let g_1 and g_2 be the predecessors of g, for a gate g of indegree 1 let g_1 be the predecessor of g. The justification for choosing c_1 and c_2 below from $M =_{\text{def}} \{0, 1, \ldots, 2^{|C|+1}\}^m \cup \{\infty\}$ is provided by Lemma 5.2.

1. $g := g_F$; $a := b$; $\sigma = 1$
2. `while` g is not an input gate `do`:
   - 2.1. `if` g is a $+$-gate and $\sigma = 1$ `then`: choose existentially $c_1, c_2 \in M$ such that $a = c_1 + c_2$; choose universally $i \in \{1, 2\}$; $g := g_i$; $a := c_i$
   - 2.2. `if` g is a $+$-gate and $\sigma = 0$ `then`: choose universally $c_1, c_2 \in M$ such that $a = c_1 + c_2$; choose existentially $i \in \{1, 2\}$; $g := g_i$; $a := c_i$
   - 2.3. `if` g is a $\cup$-gate with $\sigma = 1$ `then`: choose existentially $i \in \{1, 2\}$; $g := g_i$
   - 2.4. `if` g is a $\cup$-gate with $\sigma = 0$ `then`: choose universally $i \in \{1, 2\}$; $g := g_i$

2.5. **if** g is a $\cap$-gate with $\sigma = 1$ **then**: choose universally $i \in \{1,2\}$; $g := g_i$

2.6. **if** g is a $\cap$-gate with $\sigma = 0$ **then**: choose existentially $i \in \{1,2\}$; $g := g_i$

2.7. **if** g is a $^-$-gate **then**: $g := g_1$; $\sigma = 1 - \sigma$

3. **if** g is an input gate **then**: **if** $(\sigma = 1 \Leftrightarrow a = \alpha(g))$ **then** accept **else** reject .

Since every proof of $b \in I(F)$ is considered on one accepting subtree of this algorithm, the latter really checks whether $b \in I(F)$. □

As a direct consequence of Lemma 5.1, Lemma 5.3, and Lemma 3.2 we obtain:

Theorem 5.4 *The problems* $\mathrm{MC}^*(\cup,\cap,^-,+)$, $\mathrm{MC}(\cup,\cap,^-,+)$, $\mathrm{MC}(\cup,\cap,+)$, $\mathrm{MC}(\cup,\cap,^-,\times)$, $\mathrm{MC}(\cup,\cap,\times)$, *and* $\mathrm{MF}(\cup,\cap,^-,\times)$ *are* PSPACE*-complete. The problem* $\mathrm{MF}(\cup,\cap,^-,+,\times)$ *is* PSPACE*-hard.*

6 Beyond PSPACE

As an auxiliary tool to prove the main results of this section we introduce the following $(\cup,\cap,\cdot)$-circuits which compute finite sets of words. Such a circuit C only has gates of indegrees 0 and 2. Every input gate (i.e., gate of indegree 0) g is labelled with a word from a given alphabet Σ^*, and every gate of indegree 2 is labelled with $\cup$, $\cap$, or $\cdot$ (concatenation). For each of its gates g, the circuit computes a set $I(g) \subseteq \Sigma^*$ inductively defined as follows: If g is an input gate with label v then $I(g) =_{\mathrm{def}} \{v\}$. If g is an ω-gate ($\omega \in \{\cup,\cap,\cdot\}$) with predecessors g_l, g_r then $I(g) =_{\mathrm{def}} I(g_l)\,\omega\,I(g_r)$. Finally, $I(C) =_{\mathrm{def}} I(g_C)$ where g_C is the output gate of C. A $(\cup,\cap,\cdot)$-circuit C is called *special* if for every gate g of C there exists a $\sigma(g) \in \mathbb{N}$ such that $I(g) \subseteq \Sigma^{\sigma(g)}$ and the following holds for every gate g with predecessors g_1 and g_2: If g is a $\cap$-gate or a $\cup$-gate then $\sigma(g) = \sigma(g_1) = \sigma(g_2)$, and if g is a $\cdot$-gate then $\sigma(g) = \sigma(g_1) + \sigma(g_2)$. Let $\mathrm{NE}(\cup,\cap,\cdot)$ be the nonemptyness problem for special $(\cup,\cap,\cdot)$-circuits, i.e., $\mathrm{NE}(\cup,\cap,\cdot) =_{\mathrm{def}} \{C :$ C is a special $(\cup,\cap,\cdot)$-circuit such that $I(C) \neq \emptyset\}$.

Lemma 6.1 *The problem* $\mathrm{NE}(\cup,\cap,\cdot)$ *is* NEXPTIME*-hard.*

Proof. We describe a logarithmic space many-one master reduction from NEXPTIME to $\mathrm{NE}(\cup,\cap,\cdot)$. More specifically, given a nondeterministic one-tape Turing machine M which runs in exponential time, we construct in logarithmic space for every input x a special $(\cup,\cap,\cdot)$-circuit C_x such that: M accepts x if and only if $I(C_x) \neq \emptyset$.

Let B be the tape alphabet of M including the blank symbol $\Box$, let $A \subseteq B$ be the input alphabet, and let S be the set of states including the initial state s_0 and the accepting state s_1. Without loss of generality we may assume that M starts with the head in tape cell 1 which carries the leftmost input symbol, that M never moves its head left to tape cell 1 and that it accepts with the head in tape cell 1 and an empty tape and then working stationary. Furthermore, let p be a polynomial such that every computation path of M on input x has at most $2^{p(|x|)}$ steps.

Next we see how to describe configurations and computations of M on an input x of length n as words over the alphabet $\Sigma =_{\mathrm{def}} B \cup S \cup \{\boxtimes\}$ where $\boxtimes$ be a new symbol. A situation which occurs during M's work on input x can be completely described by the *configuration* $\boxtimes b_1 \ldots b_{i-1} s b_i b_{i+1} \ldots b_{2^{p(n)}} \boxtimes$ meaning that $b_1, \ldots b_{2^{p(n)}}$ are the symbols in the tape cells $1, \ldots, 2^{p(n)}$, resp., and that the head scans tape cell i with the state s. An accepting computation of M on input x can be described by the sequence $c_0 c_1 \ldots c_{2^{p(n)}}$ of such configuratons

where $c_0 = \boxtimes s_0 x \square^{2^{p(n)}-n} \boxtimes$, $c_{2^{p(n)}} = \boxtimes s_1 \square^{2^{p(n)}} \boxtimes$, and c_i is a successor configuration of c_{i-1} for $i = 1, \ldots 2^{p(n)}$ (i.e., c_i is a configuration which describes a possible successor situation of the situation described by c_{i-1}). Let $\mathrm{comp}_M(x) \subseteq \Sigma^{(2^{p(n)}+1)\cdot(2^{p(n)}+3)}$ be the set of all such descriptions of computations of M on input x.

In what follows we will construct in logarithmic space a special $(\cup, \cap, \cdot)$-circuit C_x such that $I(C_x) = \mathrm{comp}_M(x)$. Then, M accepts $x \Leftrightarrow \mathrm{comp}_M(x) \neq \emptyset \Leftrightarrow I(C_x) \neq \emptyset$, and we are done.

Obviously, if c' is a successor configuration of c then c and c' can differ by at most three symbols, i.e., there are $u, v \in \Sigma^*$ and $a, b, c, a', b', c' \in \Sigma$ such that $c = uabcv$, $c' = ua'b'c'v$, and $abca'b'c' \in P_M$ where $P_M \subseteq \Sigma^6$ is defined by the program of M. Let $R_M =_{\mathrm{def}} \{(uabcv, ua'b'c'v) : u, v \in \Sigma^* \wedge abca'b'c' \in P_M\}$.

Now we can write down $\mathrm{comp}_M(x)$ as

$\mathrm{comp}_M(x) = \mathrm{first}_M(x) \cap \mathrm{next}_M(x) \cap \mathrm{last}_M(x)$ where

$\mathrm{first}_M(x) =_{\mathrm{def}} \{\boxtimes s_0 x \square^{2^{p(n)}-n} \boxtimes\} \cdot \Sigma^{2^{p(n)}(2^{p(n)}+3)}$
(all strings of length $(2^{p(n)}+1)(2^{p(n)}+3)$ which start with the correct configuration)

$\mathrm{next}_M(x) =_{\mathrm{def}} \{c_0 c_1 \ldots c_{2^{p(n)}} : |c_i| = 2^{p(n)}+3 \wedge (c_{i-1}, c_i) \in R_M\}$
(all strings of length $(2^{p(n)}+1)(2^{p(n)}+3)$ which have the correct next configuration property)

$\mathrm{last}_M(x) =_{\mathrm{def}} \Sigma^{2^{p(n)}(2^{p(n)}+3)} \cdot \{\boxtimes s_1 \square^{2^{p(n)}} \boxtimes\}$
(all strings of length $(2^{p(n)}+1)(2^{p(n)}+3)$ which end with the correct configuration)

We now define sets $E(k)$ and $K_a(k)$ for $a \in \Sigma$, and we show how to construct in logarithmic space (in the *length* of the parameter k) special $(\cup, \cap, \cdot)$-circuits which compute them.

$E(k) =_{\mathrm{def}} \Sigma^k$

because of $E(0) = \{\varepsilon\}$, $E(2k) = E(k) \cdot E(k)$, and $E(2k+1) = E(k) \cdot E(k) \cdot \Sigma$.

$K_a(k) =_{\mathrm{def}} \{a^k\}$ for $a \in \Sigma$

because of $K_a(0) = \{\varepsilon\}$, $K_a(2k) = K_a(k) \cdot K_a(k)$, and $K_a(2k+1) = K_a(k) \cdot K_a(k) \cdot \{a\}$.

Now we can write:

$\mathrm{first}_M(x) = \{\boxtimes s_0 x\} \cdot K_\square(2^{p(n)}-n) \cdot \{\boxtimes\} \cdot E(2^{p(n)} \cdot (2^{p(n)}+3))$
$\mathrm{last}_M(x) = E(2^{p(n)} \cdot (2^{p(n)}+3)) \cdot \{\boxtimes s_1\} \cdot K_\square(2^{p(n)}) \cdot \{\boxtimes\}$

which shows that special $(\cup, \cap, \cdot)$-circuits can be constructed in logarithmic space (in the length of x) which compute $\mathrm{first}_M(x)$ and $\mathrm{last}_M(x)$. To do the same with $\mathrm{next}_M(x)$ we define some auxiliary sets with parameter $k \in \mathbb{N}$, and we show how to construct in logarithmic space (in the *value* of the parameter k) special $(\cup, \cap, \cdot)$-circuits which compute them.

$D(k, m) =_{\mathrm{def}} \{wvw : |w| = 2^m \wedge |v| = 2^k - 2^m + 3\} \subseteq \Sigma^{2^k+2^m+3}$

for $m = 0, 1, \ldots, k-1$ because of $D(k, 0) = \bigcup_{a \in \Sigma} \{a\} \cdot E(2^k+2) \cdot \{a\}$ and
$D(k, m+1) = (D(k, m) \cdot E(2^m)) \cap (E(2^m) \cdot D(k, m))$.

$S(k, m) =_{\mathrm{def}} \{wvw' : |w| = |w'| = 2^m+3 \wedge |v| = 2^k - 2^m \wedge (w, w') \in R_M\} \subseteq \Sigma^{2^k+2^m+6}$

for $m = 0, 1, \ldots, k$ because of
$S(k,0) = \bigcup_{a\in\Sigma} \bigcup_{ww'\in P_M} ((aw \cdot E(2^k{-}1) \cdot aw') \cup (wa \cdot E(2^k{-}1) \cdot w'a))$ and $S(k,m{+}1) =$
$((D(k,m) \cdot E(2^m{+}3)) \cap (E(2^m) \cdot S(k,m))) \cup ((S(k,m) \cdot E(2^m)) \cap (E(2^m{+}3) \cdot D(k,m)))$.

$$S(k) =_{\text{def}} \{ww' : w, w' \in \Sigma^* \wedge |w| = |w'| = 2^k{+}3 \wedge (w, w') \in R_M\} \subseteq \Sigma^{2^{k+1}+6}$$

because of $S(k) = S(k,k)$.

$$R(k,m) =_{\text{def}} \{w_0 w_1 \ldots w_{2^m} : w_{i-1}w_i \in S(k) \text{ for } i = 1, \ldots, 2^m\} \subseteq \Sigma^{(2^m+1)(2^k+3)}$$

because of $R(k,0) = S(k)$ and
$R(k,m{+}1) = (R(k,m) \cdot E((2^k{+}3)2^m)) \cap (E((2^k{+}3)2^m)) \cdot R(k,m))$.

Finally we obtain $\text{next}_M(x) = R(p(n), p(n))$. □

Theorem 6.2 *The problem* $\text{MC}(\cup, \cap, +, \times)$ *is* NEXPTIME*-complete.*

Proof. 1. To prove $\text{MC}(\cup, \cap, +, \times)$ is NEXPTIME-hard we show $\text{NE}(\cup, \cap, \cdot) \leq^{\log}_{\text{m}} \text{MC}(\cup, \cap, +, \times)$. For $w \in \{0,1\}^*$ let $\text{bin}^{-1}(w)$ be that natural number whose binary description is w (possibly with leading zeros), and for $L \subseteq \{0,1\}^*$ let $\text{bin}^{-1}(L) =_{\text{def}} \{\text{bin}^{-1}(w) : w \in L\}$.

Given a special $(\cup, \cap, \cdot)$-circuit C we construct in logarithmic space a $(\cup, \cap, +, \times)$-circuit C' such that $I(C') = \text{bin}^{-1}(I(C))$. The circuit C' basically has the same structure as C: An input gate in C with label w becomes an input gate in C' with label $\text{bin}^{-1}(w)$, a $\cup$-gate in C becomes a $\cup$-gate in C', and a $\cap$-gate in C becomes a $\cap$-gate in C'. A $\cdot$-gate g in C with predecessor g_1, g_2 such that $I(g_2) \subseteq \{0,1\}^k$ is replaced in C' by a subcircuit which computes $\text{bin}^{-1}(I(g_1) \cdot I(g_2)) = (2^k \times \text{bin}^{-1}(I(g_1))) + \text{bin}^{-1}(I(g_2))$. The 2^k can be computed by an $(\cup, \cap, \times)$-circuit C'' which has the same structure as C but instead of an input w it has the input $2^{|w|}$, and instead of a $\cdot$-gate it has a $\times$-gate. Here it is important that C is a *special* $(\cup, \cap, \cdot)$-circuit. Now, $I(C) \neq \emptyset \Leftrightarrow I(C') \neq \emptyset \Leftrightarrow 0 \in (\{0\} \times I(C'))$.

2. To see that $\text{MC}(\cup, \cap, +, \times)$ is in NEXPTIME, simply unfold a given $(\cup, \cap, +, \times)$-circuit into a (possibly exponentially larger) $(\cup, \cap, +, \times)$-formula and apply the NP-algorithm from Lemma 4.1. □

Corollary 6.3 *The problem* $\text{NE}(\cup, \cap, \cdot)$ *is* NEXPTIME*-complete.*

As an immediate consequence of Theorem 6.2 we obtain also:

Theorem 6.4 *The problem* $\text{MC}(\cup, \cap, ^-, +, \times)$ *is* NEXPTIME*-hard.*

Remark. Since $\text{MC}(\cup, \cap, ^-, +, \times) \equiv^{\log}_{\text{m}} \overline{\text{MC}(\cup, \cap, ^-, +, \times)}$ these problems cannot be in NEXPTIME unless NEXPTIME = co-NEXPTIME. In fact, there is evidence suggesting that $\text{MF}(\cup, \cap, ^-, +, \times)$ might not be decidable. Indeed, Christian Glaßer in Würzburg was the first to observe that there is a simple $\{\cup, \cap, ^-, +, \times\}$-formula G (see the Example 2.2 in Section 2) having the property that $(G, 0) \in \text{MF}(\cup, \cap, ^-, +, \times)$ if and only if Goldbach's Conjecture is true. Hence, a decision procedure for $\text{MF}(\cup, \cap, ^-, +, \times)$ would provide a terminating algorithm to test Goldbach's Conjecture; this would be surprising.

7 P-Complete Membership Problems

Theorem 7.1 *The problem* $\text{MC}(+, \times)$ *is* P*-complete.*

Proof. To prove $\mathrm{MC}(+,\times) \in \mathrm{P}$, let C be a $(+,\times)$-circuit and $b \in \mathbb{N}$. Any gate in C computes a singleton. If $I(C) = \{b\}$ then $I(g) = \{a\}$ such that $a > b$ for some gate g can only occur if this a is multiplied with 0 in a subsequent gate. Hence $b+1$ does the same job as a in this computation. Therefore a polynomial time algorithm to decide $I(C) = \{b\}$ consists in just evaluating C but replacing every $a > b$ with $b+1$.

The hardness proof is by showing that the P-complete *monotone boolean circuit value problem* [Go77] can be reduced to $\mathrm{MC}(+,\times)$. To do so we convert a monotone boolean circuit C into a $(+,\times)$-circuit C' of exactly the same structure where every boolean input 0 (1, resp.) in C becomes the integer input 0 (1, resp.) in C', every $\vee$-gate in C becomes a $+$-gate in C', and every $\wedge$-gate in C becomes a $\times$-gate in C'. It is easy to see that v evaluates to 0 in C if and only if $0 \in I(v)$ in C'. Hence, C evaluates to 0 if and only if $0 \in I(C')$. □

Theorem 7.2 *The problems* $\mathrm{MC}(\cup,\cap)$ *and* $\mathrm{MC}(\cup,\cap,^-)$ *are* P*-complete.*

Proof. To prove $\mathrm{MC}(\cup,\cap,^-) \in \mathrm{P}$, let C be a $(\cup,\cap,^-)$-circuit and $b \in \mathbb{N}$. Define $S =_{\mathrm{def}} \bigcup_{v \text{ input gate}} I(v)$. We prove that (*) for every gate v there are sets $P, N \subseteq S$ such that $I(v) = P \cup \overline{N}$. Then we can compute in polynomial time all $I(v)$ from the inputs down to the output by storing only the sets P and N.

To see (*) let $P_1, P_2, N_1, N_2 \subseteq S$ and observe $(P_1 \cup \overline{N_1}) \cup (P_2 \cup \overline{N_2}) = (P_1 \cup P_2) \cup \overline{(N_1 \cap N_2)}$, $(P_1 \cup \overline{N_1}) \cap (P_2 \cup \overline{N_2}) = ((P_1 \cap P_2) \cup (P_1 \setminus N_2) \cup (P_2 \setminus N_1)) \cup \overline{(N_1 \cup N_2)}$, and $\overline{P_1 \cup \overline{N_1}} = N_1 \setminus P_1$.

The hardness proof is by showing that the P-complete *monotone boolean circuit value problem* can be reduced to $\mathrm{MC}(\cup,\cap)$. To do so we convert a monotone boolean circuit C into a $(\cup,\cap)$-circuit C' of almost the same structure where every input gate in C with boolean value 1 becomes an input gate in C' with integer value 1, every input gate in C with boolean value 0 becomes an $\cap$-gate in C' with two input gates with labels 0 and 1, resp., as predecessors, every $\vee$-gate in C becomes a $\cup$-gate in C', and every $\wedge$-gate in C becomes a $\cap$-gate in C'. It is easy to see that v evaluates to 0 in C if and only if $I(v) = \emptyset$ in C', and v evaluates to 1 in C if and only if $I(v) = \{1\}$ in C'. Hence, C evaluates to 1 if and only if $1 \in I(C')$. □

8 Circuits with Intersection as the Only Set Operation

Circuits with intersection as the only set operation are special in the sense that every node computes a singleton or the empty set. Thus these circuits bear some relationship to circuits of the same type without intersection. For $\mathcal{O} \subseteq \{+,\times\}$ define $\mathrm{EQ}(\mathcal{O}) =_{\mathrm{def}} \{(C_1, C_2) : C_1, C_2 \text{ are } \mathcal{O}\text{-circuits such that } I(C_1) = I(C_2)\}$.

Lemma 8.1 *1.* $\mathrm{MC}(\cap,+) \leq_{\mathrm{m}}^{\log} \mathrm{EQ}(+)$

2. $\mathrm{MC}(\cap,+,\times) \equiv_{\mathrm{m}}^{\log} \mathrm{EQ}(+,\times)$

Proof. 1. Let C be a $(\cap,+)$-circuit with output gate g_C. Without loss of generality, every predecessor of a $\cap$-gate is a $+$-gate having outdegree 1. For a gate g of C, let C_g be that $(+)$-circuit which arises from C by defining g as the output gate and by removing all $\cap$-gates h where all gates fed with the output of h are now fed with output of the right predecessor of h. Further, let C' be the $(+)$-circuit which arises from C by replacing every $\cap$-gate by a $+$-gate and every input by 0. Finally, let C'_g be the $(+)$-circuit constructed from C' and C_g

as follows: Cut the vertices going into g in C', and identify gate g of C' with gate g of C_g. The output gate of C'_g is the output gate g_C of C'. If $n(g)$ is the number of paths in C from g to g_C then we have $I(C'_g) = n(g) \cdot I(C_g)$. Let A be the set of all $\cap$-gates from which g_C is accessible in C. Observe that C_g and C'_g can be computed in logarithmic space and that, if $I(\text{left predecessor of } h) = I(\text{right predecessor of } h)$ for every gate $h \in A$ then $I(g) = I(C_g)$ for every gate g of C from which g_C is accessible. For $b \in \mathbb{N}$ let $C(b)$ be the circuit which consists only of an input gate with label b. We obtain:

$$\begin{aligned} b \in I(C) &\Leftrightarrow b \in I(C_{g_C}) \wedge \textstyle\bigwedge_{h \in A \text{ with predecessors } h_1,h_2} I(C_{h_1}) = I(C_{h_2}) \\ &\Leftrightarrow b \in I(C_{g_C}) \wedge \textstyle\bigwedge_{h \text{ is a } \cap\text{-gate with predecessors } h_1,h_2} I(C'_{h_1}) = I(C'_{h_2}) \\ &\Leftrightarrow I(C_{g_C}) = I(C(b)) \wedge \textstyle\bigwedge_{h \text{ is a } \cap\text{-gate with predecessors } h_1,h_2} I(C'_{h_1}) = I(C'_{h_2}) \end{aligned}$$

Thus it is sufficient to show that there exists a logarithmic space algorithm converting the $(+)$-circuits $C_0, \ldots C_r, C'_0, \ldots C'_r$ into $(+)$-circuits C, C' such that

$$I(C) = I(C') \Leftrightarrow \big(I(C_i) = I(C'_i) \text{ for } i = 0, 1, \ldots r\big).$$

Defining $N =_{\text{def}} 2^{\sum_{i=0}^r |C_i| + \sum_{i=0}^r |C'_i|}$ we get $a \leq N-1$ for $a \in \bigcup_{i=0}^r I(C_i) \cup \bigcup_{i=0}^r I(C'_i)$. Now it is easy to construct $(+)$-circuits C, C' such that $I(C) = \sum_{i=0}^r I(C_i) \cdot N^i$ and $I(C') = \sum_{i=0}^r I(C'_i) \cdot N^i$. However, for $a_0, \ldots, a_r, b_0, \ldots, b_r \in \{0, 1, \ldots N-1\}$ there holds $\sum_{i=0}^r a_i \cdot N^i = \sum_{i=0}^r b_i \cdot N^i \Leftrightarrow (a_i = b_i \text{ for } i = 0, 1, \ldots r)$.

2. The $\leq^{\log}_{\text{m}}$-part is very similar to the proof of Item 1. Here it is sufficient to show that there exists an logarithmic space algorithm converting the $(+, \times)$-circuits $C_0, \ldots C_r, C'_0, \ldots C'_r$ into $(+, \times)$-circuits C, C' such that

$$I(C) = I(C') \Leftrightarrow \big(I(C_i) = I(C'_i) \text{ for } i = 0, 1, \ldots r\big).$$

Defining $N =_{\text{def}} 2^{\sum_{i=0}^r |C_i| + \sum_{i=0}^r |C'_i|}$ we get $a \leq 2^N - 3$ for $a \in \bigcup_{i=0}^r I(C_i) \cup \bigcup_{i=0}^r I(C'_i)$. Now it is easy to construct $(+, \times)$-circuits C, C' such that $I(C) = \prod_{i=0}^r (I(C_i)+2)^{N^i}$ and $I(C') = \prod_{i=0}^r (I(C'_i)+2)^{N^i}$. However, for $a_0, \ldots, a_r, b_0, \ldots, b_r \in \{2, 3, \ldots 2^N - 1\}$ there holds $\prod_{i=0}^r a_i^{N^i} = \prod_{i=0}^r b_i^{N^i} \Leftrightarrow (a_i = b_i \text{ for } i = 0, 1, \ldots r)$.

For the $\geq^{\log}_{\text{m}}$-part observe $I(C) = I(C') \Leftrightarrow 0 \in ((I(C) \cap I(C')) \times \{0\})$. □

For the following we need the complexity classes #L and $\text{C}_=\text{L}$. For a nondeterministic logarithmic space machine M, define $\text{n}_M(x)$ as the number of accepting paths of M on input x. The class #L precisely consists of these functions n_M. A set A is in $\text{C}_=\text{L}$ if and only if there exist $f \in \#\text{L}$ and a logarithmic space computable function g such that $x \in A \Leftrightarrow f(x) = g(x)$ for every x.

Observe that this definition is equivalent to: A set A is in $\text{C}_=\text{L}$ if and only if there exist $f, g \in \#\text{L}$ such that $x \in A \Leftrightarrow f(x) = g(x)$ for every x. For a survey on these and other counting classes see [All97].

Theorem 8.2 $\text{MC}(\cap, +)$, $\text{MC}(+)$, *and* $\text{EQ}(+)$ *are* $\leq^{\log}_{\text{m}}$*-complete for* $\text{C}_=\text{L}$.

Proof. 1. For the membership in $\text{C}_=\text{L}$ it is sufficient to prove that $\text{EQ}(+) \in \text{C}_=\text{L}$ (Lemma 8.1). For a $(+)$-circuit C, let C' be that circuit which arises from C by adding a new input gate g_I labelled with 1 and replacing every old input gate with label a by a $(+)$-circuit computing a from the sole input gate g_I. Consequently,

$$I(C) = I(C') = \text{ the number of paths in } C' \text{ from the input to the output,}$$

and hence I is a #L-function. For given (+)-circuits C_1, C_2 define the #L-functions f, g by $f(C_1, C_2) =_{\text{def}} I(C_1)$ and $g(C_1, C_2) =_{\text{def}} I(C_2)$. Now, $(C_1, C_2) \in \text{EQ}(+) \Leftrightarrow f(C_1, C_2) = g(C_1, C_2)$.

2. For the $\text{C}_=\text{L}$-hardness of MC(+) let $A \in \text{C}_=\text{L}$. By definition there exist a function $f \in \#\text{L}$ and a logarithmic space computable function g such that $x \in A \Leftrightarrow f(x) = g(x)$. Let M be a nondeterministic logspace Turing machine such that $f(x)$ is the number of accepting paths of M on input x. Let further C_x be the transition graph of M on input x. If we consider C_x to be a (+)-circuit whose accepting and rejecting nodes are inputs with labels 1 and 0 respectively, then we have $I(C_x) = \{f(x)\}$. Consequently

$$x \in A \Leftrightarrow f(x) = g(x) \Leftrightarrow g(x) \in I(C_x) \Leftrightarrow (C_x, g(x)) \in \text{MC}(+). \qquad \square$$

Theorem 8.3 *1. The problem* $\text{MC}(\cap, +, \times)$ *is in* co-NP. [1]

2. The problem $\text{MC}(\cap, +, \times)$ *is* P*-hard.*

Proof. 1. Because of Lemma 8.1 it is sufficient to prove $\text{EQ}(+, \times) \in$ co-NP. Let C, C' be $(+, \times)$-circuits. For every gate g of C or C' there holds $I(g) \leq 2^{2^{|C|+|C'|}}$. By the Chinese Remainder Theorem we obtain

$$I(C) = I(C') \Leftrightarrow \forall b (b \leq 2^{|C|+|C'|} \rightarrow I(C) \equiv I(C') \text{ modulo } b).$$

Since $b \leq 2^{|C|+|C'|}$, we can evaluate the circuits C and C' modulo b in polynomial time.

2. This is a direct consequence of Theorem 7.1 $\square$

Theorem 8.4 *1. The problem* $\text{MC}(\cap, \times)$ *is in* P.

2. The problem $\text{MC}(\cap, \times)$ *is* $\text{C}_=\text{L}$*-hard.*

Proof. 1. By Lemma 3.2 it suffices to prove that $\text{MC}^*(\cap, +)$ is in P. Let C be a $\{\cap, +\}$-circuit processing values from $\mathbb{N}^m \cup \{\infty\}$. There is at most one m-tuple or ∞ in $I(v)$ for every gate v. The size of these m-tuples is polynomial in the size of the input. So we can evaluate C step by step in polynomial time.

2. For the $\text{C}_=\text{L}$-hardness of $\text{MC}(\cap, \times)$ let $A \in \text{C}_=\text{L}$. Hence there exist functions $f_1, f_2 \in \#\text{L}$ such that $x \in A \Leftrightarrow f_1(x) = f_2(x)$. For $i = 1, 2$, let M_i be a nondeterministic logspace Turing machine such that $f_i(x)$ is the number of accepting paths of M_i on input x. Let further C_i^x be the transition graph of M_i on input x. If we consider C_i^x to be a $(\times)$-circuit whose accepting nodes are inputs with label 2 then we have $I(C_i^x) = \{2^{f_i(x)}\}$. Consequently

$$x \in A \Leftrightarrow f_1(x) = f_2(x) \Leftrightarrow I(C_1^x) = I(C_2^x) \Leftrightarrow 0 \in ((I(C_1^x) \cap I(C_2^x)) \times \{0\}). \qquad \square$$

Theorem 8.5 *The problem* $\text{MC}(\times)$ *is* NL*-complete.*

Proof. 1. To prove the hardness we reduce the NL-complete graph accessibility problem GAP for directed acyclic graphs to the problem $\text{MC}(\times)$. Let $G = (V, E)$ be a directed acyclic graph and $s, t \in V$ be the source and target vertices, resp. Without loss of generality assume every vertex has indegree 0 or 2 and that s has indegree 0. Now convert G into a $(\times)$-circuit C by labelling every vertex with indegree 2 by $\times$, the source vertex s by 0 and all other vertices with

[1] Christian Glaßer, Würzburg, recently proved that $\text{MC}(\cap, +, \times)$ is in co-R.

indegree 0 by 1. We obtain

$$(G,s,t) \in \mathrm{GAP} \Leftrightarrow \text{ there is a path in } G \text{ from } s \text{ to } t \Leftrightarrow 0 \in I(C) \Leftrightarrow (C,0) \in \mathrm{MC}(\times).$$

2. Now we describe a many-one reduction from MC($\times$) to the iterated multiplication decision problem IMD $=_{\mathrm{def}} \{(a_1,\ldots a_r,b) : a_1,\ldots a_r,b \in \mathbb{N} \wedge \prod_{i=1}^{r} a_i = b\}$ via a function which is logspace computable with an oracle from NL. Since IMD $\in$ L ([CDL], or better yet, uniform TC^0 [ABH01, He01]) we obtain $\mathrm{MC}(\times) \in \mathrm{L}^{\mathrm{NL}}$. However, $\mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$.

Let C be a ($\times$)-circuit and $b \in \mathbb{N}$. Let $g_1,\ldots g_r$ be the input gates of C, let g_C be the output gate of C, and let $n(C,g,g')$ be the number of different paths in C from gate g to gate g'. Obviously, $n \in \#\mathrm{L}$, and we obtain $I(C) = \prod_{i=1}^{r} I(g_i)^{n(C,g_i,g_C)}$. Defining $s(i) =_{\mathrm{def}} \min\{n(C,g_i,g_C), |b|+1\}$ for $i = 1,\ldots,r$ we obtain

$$\begin{aligned} b \in I(C) &\Leftrightarrow b = \textstyle\prod_{i=1}^{r} I(g_i)^{n(C,g_i,g_C)} \Leftrightarrow b = \prod_{i=1}^{r} I(g_i)^{s(i)} \\ &\Leftrightarrow (\underbrace{I(g_1),\ldots,I(g_1)}_{s(1)},\ldots,\underbrace{I(g_r),\ldots,I(g_r)}_{s(r)},b) \in \mathrm{IMD} \end{aligned}$$

The latter tuple of numbers is generated as follows: for $i = 1,\ldots r$ and $k = 1,\ldots,|b|+1$ ask $n(C,g_i,g_C) \geq k$, which are queries to a NL-set [ARZ99]. (This owes to the fact that only small k, i.e. k whose values are polynomially bounded in the length of the input, are considered.) If such a query $n(C,g_i,g_C) \geq k$ is answered in the affirmative, output $I(g_i)$, and finally output b. □

9 Further Results

Theorem 9.1 *The problems* MC($\cup$) *and* MC($\cap$) *are* NL*-complete.*

Proof. Upper bounds. For a $\cup$-circuit C and $b \in \mathbb{N}$, there holds $b \in I(C)$ if and only if there exists a path in C from an input gate with label b to the output gate. This can be checked by an NL-algorithm.

For a $\cap$-circuit C and $b \in \mathbb{N}$, there holds $b \in I(C)$ if and only if all input gates from which there exists a path to the output gate have label b. This can be checked by an co-NL-algorithm. However, co-NL = NL.

Lower bounds. We reduce the NL-complete graph accessibility problem GAP for directed acyclic graphs to the problems MC($\cup$) and MC($\cap$). Let $G = (V,E)$ be a directed acyclic graph and $s,t \in V$ be the source and target vertices, resp. Without loss of generality assume every vertex has indegree 0 or 2 and that s has indegree 0. Now convert G into a ($\cup$)-circuit C by labelling every vertex with indegree 2 by $\cup$, the source vertex s by 0 and all other vertices with indegree 0 by 1. We obtain

$$(G,s,t) \in \mathrm{GAP} \Leftrightarrow \text{ there is a path in } G \text{ from } s \text{ to } t \Leftrightarrow 0 \in I(C) \Leftrightarrow (C,0) \in \mathrm{MC}(\cup).$$

Furthermore, convert G into a ($\cap$)-circuit C' by labelling every vertex with indegree 2 by $\cap$, the source vertex s by 0 and all other vertices with indegree 0 by 1. We obtain

$$(G,s,t) \notin \mathrm{GAP} \Leftrightarrow \text{ there is no path in } G \text{ from } s \text{ to } t \Leftrightarrow 1 \in I(C) \Leftrightarrow (C,1) \in \mathrm{MC}(\cap).$$

Consequently, $\overline{\mathrm{GAP}} \leq_{\mathrm{m}}^{\log} \mathrm{MC}(\cap)$. However, because of NL = co-NL we have $\mathrm{GAP} \leq_{\mathrm{m}}^{\log} \overline{\mathrm{GAP}}$. □

Theorem 9.2 *The problems* $\mathrm{MF}(\cap,+,\times)$ *and* $\mathrm{MF}(+,\times)$ *are in* DLOGCFL.

Proof. For a $(\cap,+,\times)$-formula F, no number computed in a gate of F can be greater than $2^{|F|}$. Hence, by the Chinese Remainder Theorem,

$$b \in I(F) \Leftrightarrow \forall a(a \leq |F| \rightarrow b \equiv I(F) \text{ modulo } a).$$

Now "$b \equiv I(F)$ modulo a" can be tested for all $a \leq |F|$ one after the other. The current a can be stored in logarithmic space, and the pushdown is used to evaluate the formula F in a depth-first manner. □

Theorem 9.3 *The problems* $\mathrm{MF}(\cup,\cap,^-)$, $\mathrm{MF}(\cup,\cap)$, $\mathrm{MF}(\cup)$, $\mathrm{MF}(\cap)$, $\mathrm{MF}(\cap,+)$, $\mathrm{MF}(\cap,\times)$, $\mathrm{MF}(+)$ *and* $\mathrm{MF}(\times)$ *are* L*-complete under* AC^0*-reducibility.*

Proof. *Hardness.* Any one of the associative binary operations $\cup,\cap,+$ and $\times$ is able to simulate the boolean $\vee$ operation once an appropriate representation for the truth values is chosen. For example, representing $\sigma(\text{true}) = \emptyset$ and $\sigma(\text{false}) = \{0\}$ allows $\cap$ to simulate $\vee$ since $a \vee b$ is true iff $\sigma(a) \cap \sigma(b) = \emptyset$, and $a \vee b$ is false iff $\sigma(a) \cap \sigma(b) = \{0\}$. Hence, to prove all eight hardness claims, it suffices to prove that evaluating a boolean formula (in our encoding, as a possibly disconnected outdegree-one circuit) with $\vee$ as the only operation is L-hard. But this is clear, and is explicitly stated under NC^1-reducibility in [BM95] (where the formulas involve both $\vee$ and $\wedge$, but $\wedge$ is used only to enforce the [BM95] requirement that every formula be connected). The more restrictive AC^0-reducibility is attained by the usual refinements.

Upper bounds. It suffices to prove that $\mathrm{MF}(\cup,\cap,^-)$, $\mathrm{MF}(\cap,+)$ and $\mathrm{MF}(\cap,\times)$ are in L.

One way to solve $\mathrm{MF}(\cup,\cap,^-)$ is to reduce in logspace the problem to that of evaluating a boolean $(\vee,\wedge,\neg)$-formula, then transforming in logspace (as mentioned in [BM95, P. 444]) the connected part of this boolean formula into infix notation, and finally evaluating the infix boolean formula in logspace [Ly77] (or in $\mathrm{NC}^1 \subseteq \mathrm{L}$ using Buss's algorithm [Bu87]). To see the first step, namely reducing a $(\cup,\cap,^-)$-formula F with $b \in \mathbb{N}$ to a boolean $(\vee,\wedge,\neg)$-formula F', let F' have exactly the same structure as F, with every input in F labelled b (resp. having label $\neq b$) giving rise in F' to an input labelled 1 (resp. 0), and with every $\cup$-gate in F (resp. $\cap$-gate, $^-$-gate) giving rise to a $\vee$-gate in F' (resp. $\wedge$-gate, $\neg$-gate). Then for every gate g, ($b \in I(g)$ in F) $\Leftrightarrow$ ($I(g) = 1$ in F'). Hence $b \in I(F) \Leftrightarrow I(F') = 1$.

The case $\mathrm{MF}(\cap,+)$ is reduced in logspace to $\mathrm{EQ}(+)$ as in Lemma 8.1. The $\mathrm{EQ}(+)$ instance now involves two $\{+\}$-formulas. Evaluating a $\{+\}$-formula can be done in logspace by identifying (in L) the inputs that contribute to the output, and forming the iterated sum of those inputs in $\mathrm{TC}^0 \subseteq \mathrm{L}$.

The case $\mathrm{MF}(\cap,\times)$ is handled in the same way, except that the final step now involves evaluating an iterated product, and this can be done in $\mathrm{TC}^0 \subseteq \mathrm{L}$ by [ABH01, CDL]. □

The L-hardness of the MF problems considered in this section owe to our choice of formula encoding. At such low complexity levels, a more appropriate choice is infix notation; then $\mathrm{MF}(\cup,\cap,^-)$ becomes NC^1-complete by equivalence with the Boolean formula value problem [Bu87], and some of the other restrictions considered in Theorem 9.3 drop down into yet smaller classes.

10 Conclusion

Table I summarizes the known complexity status of the membership problems for arithmetic circuits over subsets of $\mathbb{N}$. Several open questions are apparent from the table, most notably the intriguing question of finding an upper bound on the complexity of $\mathrm{MC}(\cup, \cap, ^-, +, \times)$ or proving the problem undecidable.

We observe that the problems $\mathrm{MC}(\times)$ and $\mathrm{MC}(+)$, complete for NL and for $\mathrm{C_=L}$ respectively, offer an interesting new perspective on these two classes. If one could reduce $\mathrm{MC}(+)$ to $\mathrm{MC}(\times)$, then it would follow that $\mathrm{NL} = \mathrm{C_=L}$.

Finally, some cases of interest were left out of our analysis, namely those in which complementation is the only set operation. Klaus Reinhardt (private communication) has shown that $\mathrm{MF}(^-, +, \times)$ is already powerful enough to express a weak form of Goldbach's conjecture. He also extended our techniques to prove that $\mathrm{MC}(^-, \times)$, $\mathrm{MC}(^-, +)$, $\mathrm{MF}(^-, \times)$ and $\mathrm{MF}(^-, +)$ are PSPACE-complete, and he observed that $\mathrm{MC}(^-)$ is L-complete.

Acknowledgements. We are grateful to Eric Allender for his help with the upper bound in Theorem 8.5, to Heribert Vollmer, Christian Glaßer, Daniel Meister, Stephen Travers, and Hans-Georg Breunig for very useful discussions, and to anonymous referees for valuable suggestions, including the need to correct our choice of formula encoding and to clarify its ramifications.

$\mathcal{O}$	$\mathrm{MC}(\mathcal{O})$ lower bound	$\mathrm{MC}(\mathcal{O})$ upper bound	Th.	$\mathrm{MF}(\mathcal{O})$ low. bound	$\mathrm{MF}(\mathcal{O})$ upp. bound	Th.
$\cup, \cap, ^-, +, \times$	NEXPTIME	?	6.4	PSPACE	?	5.4
$\cup, \cap, +, \times$	NEXPTIME	NEXPTIME	6.2	NP	NP	4.4
$\cup, +, \times$	PSPACE	PSPACE	2.3	NP	NP	4.4
$\cap, +, \times$	P	co-R	8.3	L	DLOGCFL	9.2
$+, \times$	P	P	7.1	L	DLOGCFL	9.2
$\cup, \cap, ^-, +$	PSPACE	PSPACE	5.4	PSPACE	PSPACE	2.3
$\cup, \cap, +$	PSPACE	PSPACE	5.4	NP	NP	4.4
$\cup, +$	NP	NP	4.4	NP	NP	2.3
$\cap, +$	$\mathrm{C_=L}$	$\mathrm{C_=L}$	8.2	L	L	9.3
$+$	$\mathrm{C_=L}$	$\mathrm{C_=L}$	8.2	L	L	9.3
$\cup, \cap, ^-, \times$	PSPACE	PSPACE	5.4	PSPACE	PSPACE	5.4
$\cup, \cap, \times$	PSPACE	PSPACE	5.4	NP	NP	4.4
$\cup, \times$	NP	NP	4.4	NP	NP	4.4
$\cap, \times$	$\mathrm{C_=L}$	P	8.4	L	L	9.3
$\times$	NL	NL	8.5	L	L	9.3
$\cup, \cap, ^-$	P	P	7.2	L	L	9.3
$\cup, \cap$	P	P	7.2	L	L	9.3
$\cup$	NL	NL	9.1	L	L	9.3
$\cap$	NL	NL	9.1	L	L	9.3

TABLE I: State of the art. The results on $\mathrm{MF}(\cup, +)$, $\mathrm{MF}(\cup, \cap, ^-, +)$ as well as on $\mathrm{MC}(\cup, +, \times)$ were already known from the literature, please refer to the relevant sections for the appropriate credit. Lower bounds of course refer to hardness results.

Literatur

[AAD00] M. Agrawal, E. Allender, and S. Datta, On TC^0, AC^0, and arithmetic circuits, *J. Computer and System Sciences* 60 (2000), pp. 395–421.

[All97] E. Allender, Making computation count: Arithmetic circuits in the Nineties, in the Complexity Theory Column, *SIGACT NEWS* 28 (4) (1997) pp. 2-15.

[ARZ99] E. Allender, K. Reinhardt, S. Zhou, Isolation, matching, and counting: Uniform and nonuniform upper bounds, *J. Computer and System Sciences* 59(1999), pp. 164–181.

[ABH01] E. Allender, D. Barrington, and W. Hesse, Uniform constant-depth threshold circuits for division and iterated multiplication, *Proceedings 16th Conference on Computational Complexity*, 2001, pp. 150–159.

[AJMV98] E. Allender, J. Jiao, M. Mahajan and V. Vinay, Non-commutative arithmetic circuits: depth-reduction and depth lower bounds, *Theoretical Computer Science* Vol. 209 (1,2) (1998), pp. 47-86.

[BS96] E. Bach, J. Shallit, Algorithmic Number Theory, Volume I: Efficient Algorithms, MIT Press 1996.

[BM95] M. Beaudry and P. McKenzie, Circuits, matrices and nonassociative computation, *J. Computer and System Sciences* 50 (1995), pp. 441–455.

[BMPT97] M. Beaudry, P. McKenzie, P. Péladeau, D. Thérien, Finite monoids: from word to circuit evaluation, *SIAM J. Computing* 26 (1997), pp. 138–152.

[BCGR92] S. Buss, S. Cook, A. Gupta, V. Ramachandran, An optimal parallel algorithm for formula evaluation, *SIAM J. Computing* 21 (1992), pp. 755–780.

[Bu87] S.R. Buss, The boolean formula value problemis in ALOGTIME, Proceedings 19th ACM Symp, on the Theory of Computing, 1987, pp. 123–131.

[CMTV98] H. Caussinus, P. McKenzie, D. Thérien, H. Vollmer, Nondeterministic NC^1 computation, *J. Computer and System Sciences*, 57 (2), 1998, pp. 200–212.

[CSV84] A.K. Chandra, L. Stockmeyer, U. Vishkin, Constant depth reducibility, *SIAM Journal on Computing*, 13, 1984, pp. 423–439.

[CDL] A. Chiu, G. Davida, and B. Litow, *NC^1 division*, available at http://www.cs.jcu.edu.au/ bruce/papers/crr00_3.ps.gz

[Go77] L.M. Goldschlager, The monotone and planar circuit value problems are logspace complete for P, *SIGACT News*, 9, 1977, pp. 25–29.

[He01] W. Hesse, Division in uniform TC^0, Proceedings of the 28th International Colloquium on Automata, Languages, and Programming 2001, Lecture Notes in Computer Science 2076, pp. 104–114

[Ga84] J. von zur Gathen, Parallel algorithms for algebraic problems, *SIAM J. on Computing* 13(4), (1984), pp. 802-824.

[GHR95] R. Greenlaw, J. Hoover and L. Ruzzo, *Limits to parallel computation, P-completeness theory,* Oxford University Press, 1995, 311 pages.

[Ly77] N. Lynch, Logarithmic space recognition and translation of parenthesis languages, *Journal of the ACM*, 24, 1977, pp. 583–590.

[Og98] M. Ogihara, The PL hierarchy collapses, *SIAM Journal of Computing*, 27, 1998, pp. 1430–1437.

[SM73] L.J. Stockmeyer, A.R. Meyer, Word Problems Requiring Exponential Time, Proceedings 5th ACM Symposium on the Theory of Computing, 1973, pp. 1–9.

[Vo00] H. Vollmer, Circuit complexity, Springer, 2000.

[Wa84] K. W. Wagner, The complexity of problems concerning graphs with regularities, Proceedings 11th Mathematical Foundations of Computer Science 1984, Lecture Notes in Computer Science 176, pp. 544–552. Full version as TR N/84/52, Friedrich-Schiller-Universität Jena, 1984.

[Ya00] K. Yang, Integer circuit evaluation is PSPACE-complete, Proceedings 15th Conference on Computational Complexity, 2000, pp. 204–211.