Circuit Evaluation for Finite Semirings*

Moses Ganardi[1], Danny Hucke[2], Daniel König[3], and Markus Lohrey[4]

1 **University of Siegen, Siegen, Germany**
ganardi@eti.uni-siegen.de
2 **University of Siegen, Siegen, Germany**
hucke@eti.uni-siegen.de
3 **University of Siegen, Siegen, Germany**
koenig@eti.uni-siegen.de
4 **University of Siegen, Siegen, Germany**
lohrey@eti.uni-siegen.de

Abstract

The circuit evaluation problem for finite semirings is considered, where semirings are not assumed to have an additive or multiplicative identity. The following dichotomy is shown: If a finite semiring R (i) has a solvable multiplicative semigroup and (ii) does not contain a subsemiring with an additive identity 0 and a multiplicative identity $1 \neq 0$, then its circuit evaluation problem is in $\mathsf{DET} \subseteq \mathsf{NC}^2$. In all other cases, the circuit evaluation problem is P-complete.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases circuit value problem, finite semirings, circuit complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2017.35

1 Introduction

Circuit evaluation problems are among the most well-studied computational problems in complexity theory. In its most general formulation, one has an algebraic structure $\mathcal{A} = (D, f_1, \ldots, f_k)$, where the f_i are mappings $f_i : D^{n_i} \to D$. A circuit over the structure $\mathcal{A}$ is a directed acyclic graph (dag) where every inner node is labelled with one of the operations f_i and has exactly n_i incoming edges that are linearly ordered. The leaf nodes of the dag are labelled with elements of D (for this, one needs a suitable finite representation of elements from D), and there is a distinguished output node. The task is to evaluate this dag in the natural way, and to return the value of the output node.

In his seminal paper [19], Ladner proved that the circuit evaluation problem for the Boolean semiring $\mathbb{B}_2 = (\{0,1\}, \vee, \wedge)$ is P-complete. This result marks a cornerstone in the theory of P-completeness [15], and motivated the investigation of circuit evaluation problems for other algebraic structures. A large part of the literature is focused on commutative (possibly infinite) semirings [1, 23, 31] or circuits with certain structural restrictions (e.g. planar circuits [14, 18, 27] or tree-like circuits [9, 24]). In [25], Miller and Teng proved that circuits over any finite semiring can be evaluated with polynomially many processors in time $O((\log n)(\log dn))$ on a CRCW PRAM, where n is the size of the circuit and d is the formal degree of the circuit. The latter is a parameter that can be exponential in the circuit size n. On the other hand, the authors are not aware of any NC-algorithms for evaluating

---

* A full version of the paper is available at http://arxiv.org/abs/1602.04560.

34th Symposium on Theoretical Aspects of Computer Science (STACS 2017).
Editors: Heribert Vollmer and Brigitte Vallée; Article No. 35; pp. 35:1–35:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

general (exponential degree) circuits even for finite semirings. The lack of such algorithms is probably due to Ladner's result, which excludes efficient parallel algorithms in the presence of a Boolean subsemiring unless $\mathsf{P} = \mathsf{NC}$. On the other hand, in the context of semigroups, there exist NC-algorithms for circuit evaluation. In [8], the following dichotomy result was shown for finite semigroups: If the finite semigroup is solvable (meaning that every subgroup is a solvable group), then circuit evaluation is in NC (in fact, in DET, which is the class of all problems that are AC^0-reducible to the computation of an integer determinant [10, 11]), otherwise circuit evaluation is P-complete.

In this paper, we extend the work of [8] from finite semigroups to finite semirings. On first sight, Ladner's result seems to exclude efficient parallel algorithms: It is not hard to show that if the finite semiring has an additive identity 0 and a multiplicative identity $1 \neq 0$ (where 0 is not necessarily absorbing with respect to multiplication), then circuit evaluation is P-complete, see Lemma 6. Therefore, we take the most general reasonable definition of semirings: A semiring is a structure $(R, +, \cdot)$, where $(R, +)$ is a commutative semigroup, $(R, \cdot)$ is a semigroup, and $\cdot$ distributes (on the left and right) over $+$. In particular, we neither require the existence of a 0 nor a 1. Our main result states that in this general setting there are only two obstacles to circuit evaluation in NC: non-solvability of the multiplicative structure and the existence of a zero and a one (different from the zero) in a subsemiring. More precisely, we show the following two results, where a semiring is called $\{0, 1\}$-free if there exists no subsemiring with an additive identity 0 and a multiplicative identity $1 \neq 0$:

1. If a finite semiring is not $\{0, 1\}$-free, then the circuit evaluation problem is P-complete.
2. If a finite semiring $(R, +, \cdot)$ is $\{0, 1\}$-free, then its circuit evaluation problem can be solved with AC^0-circuits equipped with oracle gates for (a) graph reachability and (b) the circuit evaluation problems for the commutative semigroup $(R, +)$ and the semigroup $(R, \cdot)$.

Together with the dichotomy result from [8] (and the fact that commutative semigroups are solvable) we get the following result: For every finite semiring $(R, +, \cdot)$, the circuit evaluation problem is in NC (in fact, in DET) if $(R, \cdot)$ is solvable and $(R, +, \cdot)$ is $\{0, 1\}$-free. Moreover, if one of these conditions fails, then circuit evaluation is P-complete.

The hard part of the proof is to show the above statement 2. We will proceed in two steps. In the first step we reduce the circuit evaluation problem for a finite semiring R to the evaluation of a so-called type admitting circuit. This is a circuit where every gate evaluates to an element of the form eaf, where e and f are multiplicative idempotents of R. Moreover, these idempotents e and f have to satisfy a certain compatibility condition that will be expressed by a so-called type function. In a second step, we present a parallel evaluation algorithm for type admitting circuits. Only for this second step we need the assumption that the semiring is $\{0, 1\}$-free.

In Section 6 we present an application of our main result for circuit evaluation to formal language theory. We consider the intersection non-emptiness problem for a given context-free language and a fixed regular language L. If the context-free language is given by an arbitrary context-free grammar, then we show that the intersection non-emptiness problem is P-complete as long as L is not empty (Theorem 19). It turns out that the reason for this is non-productivity of nonterminals. We therefore consider a restricted version of the intersection non-emptiness problem, where every nonterminal of the input context-free grammar must be productive. To avoid a promise problem (testing productivity of a nonterminal is P-complete), we in addition provide a witness of productivity for every nonterminal. This witness consists of exactly one production $A \to w$ for every nonterminal of A where w may contain nonterminal symbols such that the set of all selected productions is an acyclic grammar $\mathcal{H}$. This ensures that $\mathcal{H}$ derives for every nonterminal A exactly one string that is a witness of the productivity of A. We then show that this restricted version of the intersection non-emptiness problem with the fixed regular language L is equivalent (with respect to constant depth reductions) to the circuit evaluation problem for a certain finite semiring that is derived from the syntactic monoid of the regular language L.

Full proofs can be found in the long version [12].

Further related work. We mentioned already existing work on circuit evaluation for (possibly infinite) semirings [1, 23, 25, 31]. For infinite groups, the circuit evaluation problem is also known as the compressed word problem [20]. In the context of parallel algorithms, the third and fourth author recently proved that the circuit evaluation problem for finitely generated (but infinite) nilpotent groups belongs to DET [17]. For finite non-associative groupoids, the complexity of circuit evaluation was studied in [26], and some of the results from [8] for semigroups were generalized to the non-associative setting. In [6], the problem of evaluating tensor circuits is studied. The complexity of this problem is quite high: Whether a given tensor circuit over the Boolean semiring evaluates to the (1×1)-matrix (0) is complete for nondeterministic exponential time. Finally, let us mention the papers [22, 30], where circuit evaluation problems are studied for the power set structures $(2^{\mathbb{N}}, +, \cdot, \cup, \cap, ^-)$ and $(2^{\mathbb{Z}}, +, \cdot, \cup, \cap, ^-)$, where $+$ and $\cdot$ are evaluated on sets via $A \circ B = \{a \circ b \mid a \in A, b \in B\}$. Completeness results for a large range of complexity classes are shown in [22, 30].

A variant of our intersection non-emptiness problem was studied in [29]. There, a context-free language L is fixed, a non-deterministic finite automaton $\mathcal{A}$ is the input, and the question is, whether $L \cap L(\mathcal{A}) = \emptyset$ holds. The authors present large classes of context-free languages such that for each member the intersection non-emptiness problem with a given regular language is P-complete (resp., NL-complete).

2 Computational complexity

For background in complexity theory the reader might consult [4]. We assume that the reader is familiar with the complexity classes NL (non-deterministic logspace) and P (deterministic polynomial time). A function is logspace-computable if it can be computed by a deterministic Turing-machine with a logspace-bounded work tape, a read-only input tape, and a write-only output tape. Note that the logarithmic space bound only applies to the work tape. P-hardness will refer to logspace reductions.

We use standard definitions concerning circuit complexity, see e.g. [33]. All circuit families in this paper are implicitly assumed to be DLOGTIME-uniform. We will consider the class AC^0 of all problems that can be recognized by a polynomial size circuit family of constant depth built up from NOT-gates (which have fan-in one) and AND- and OR-gates of unbounded fan-in. The class NC^k $(k \geq 1)$ is defined by polynomial size circuit families of depth $O(\log^k n)$ that use NOT-gates, and AND- and OR-gates of fan-in two. One defines $\mathsf{NC} = \bigcup_{k \geq 1} \mathsf{NC}^k$. The above language classes can be easily generalized to classes of functions by allowing circuits with several output gates. Of course, this only allows to compute functions $f : \{0,1\}^* \to \{0,1\}^*$ such that $|f(x)| = |f(y)|$ whenever $|x| = |y|$. If this condition is not satisfied, one has to consider a suitably padded version of f.

We use the standard notion of constant depth reducibility: For functions $f_1, \ldots, f_k$ let $\mathsf{AC}^0(f_1, \ldots, f_k)$ be the class of all functions that can be computed with a polynomial size circuit family of constant depth that uses NOT-gates and unbounded fan-in AND-gates, OR-gates, and f_i-oracle gates $(1 \leq i \leq k)$. Here, an f_i-oracle gate receives an ordered tuple of inputs $x_1, x_2, \ldots, x_n$ and outputs the bits of $f_i(x_1 x_2 \cdots x_n)$. By taking the characteristic function of a language, we can also allow a language $L_i \subseteq \{0,1\}^*$ in place of f_i. Note that the function class $\mathsf{AC}^0(f_1,\dots,f_k)$ is closed under composition (since the composition of two AC^0-circuits is again an AC^0-circuit). We write $\mathsf{AC}^0(\mathsf{NL},f_1,\dots,f_k)$ for $\mathsf{AC}^0(\mathsf{GAP},f_1,\dots,f_k)$, where GAP is the NL-complete graph accessibility problem. The class $\mathsf{AC}^0(\mathsf{NL})$ is studied in [3]. It has several alternative characterizations and can be viewed as a nondeterministic version of functional logspace. As remarked in [3], the restriction of $\mathsf{AC}^0(\mathsf{NL})$ to 0-1 functions is NL. Clearly, every logspace-computable function belongs to $\mathsf{AC}^0(\mathsf{NL})$: The NL-oracle can be used to directly compute the output bits of a logspace-computable function.

Let $\mathsf{DET} = \mathsf{AC}^0(\mathsf{det})$, where det is the function that maps a binary encoded integer matrix to the binary encoding of its determinant, see [10]. Actually, Cook originally defined DET as $\mathsf{NC}^1(\mathsf{det})$ [10], but later [11] remarked that the above definition via AC^0-circuits seems to be more natural. For instance, it implies that DET is equal to the #L-hierarchy.

We defined DET as a function class, but the definition can be extended to languages by considering their characteristic functions. It is well known that $\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{NC}^2$ [11]. From $\mathsf{NL} \subseteq \mathsf{DET}$, it follows easily that $\mathsf{AC}^0(\mathsf{NL},f_1,\dots,f_k) \subseteq \mathsf{DET}$ whenever $f_1,\dots,f_k \in \mathsf{DET}$.

3 Algebraic structures, semigroups, and semirings

An *algebraic structure* $\mathcal{A} = (D, f_1,\dots,f_k)$ consists of a non-empty *domain* D and operations $f_i : D^{n_i} \to D$ for $1 \le i \le k$. We often identify the domain with the structure, if it is clear from the context. A *substructure* of $\mathcal{A}$ is a subset $B \subseteq D$ that is closed under each of the operations f_i. We identify B with the structure $(B, g_1,\dots,g_k)$, where $g_i : B^{n_i} \to B$ is the restriction of f_i to B^{n_i} for all $1 \le i \le k$. We mainly deal with semigroups and semirings. In the following two subsection we present the necessary background. For further details concerning semigroup theory (resp., semiring theory) see [28] (resp., [13]).

3.1 Semigroups

A *semigroup* $(S,\circ)$ (or briefly S) is an algebraic structure with a single associative binary operation. We usually write st for $s \circ t$. If $st = ts$ for all $s,t \in S$, we call S *commutative*. A set $I \subseteq S$ is called a *semigroup ideal* if for all $s \in S$, $a \in I$ we have $sa, as \in I$. An element $e \in S$ is called *idempotent* if $ee = e$. It is well-known that for every finite semigroup S and $s \in S$ there exists an $n \ge 1$ such that s^n is idempotent. In particular, every finite semigroup contains an idempotent element. By taking the smallest common multiple of all these n, one obtains an $\omega \ge 1$ such that s^ω is idempotent for all $s \in S$. The set of all idempotents of S is denoted with $E(S)$. If S is finite, then $SE(S)S = S^n$ where $n = |S|$. Moreover, $S^n = S^m$ for all $m \ge n$.

A semigroup M with an identity element $1 \in M$, i.e. $1m = m1 = m$ for all $m \in M$, is called a *monoid*. With S^1 we denote the monoid that is obtained from a semigroup S by adding a fresh element 1, which becomes the identity element of S^1 by setting $1s = s1 = s$ for all $s \in S \cup \{1\}$. In case M is a monoid and N is a submonoid of M, we do not require that the identity element of N is the identity element of M. But, clearly, the identity element of the submonoid N must be an idempotent element of M. In fact, for every semigroup S and every idempotent $e \in E(S)$, the set $eSe = \{ese \mid s \in S\}$ is a submonoid of S with identity e, which is also called a *local submonoid* of S. The local submonoid eSe is the maximal submonoid of S whose identity element is e. A semigroup S is *aperiodic* if every subgroup of S is trivial. A semigroup S is *solvable* if every subgroup G of S is a solvable group, i.e., repeatedly taking the commutator subgroup leads from G to 1. Since Abelian groups are solvable, every commutative semigroup is solvable.

3.2 Semirings

A *semiring* $(R, +, \cdot)$ consists of a non-empty set R with two operations $+$ and $\cdot$ such that $(R, +)$ is a commutative semigroup, $(R, \cdot)$ is a semigroup, and $\cdot$ left- and right-distributes over $+$, i.e., $a \cdot (b + c) = ab + ac$ and $(b + c) \cdot a = ba + ca$ (as usual, we write ab for $a \cdot b$). Note that we neither require the existence of an additive identity 0 nor the existence of a multiplicative identity 1. We denote with $R_+ = (R, +)$ the additive semigroup of R and with $R_\bullet = (R, \cdot)$ the multiplicative semigroup of R. For $n \geq 1$ and $r \in R$ we write $n \cdot r$ or just nr for $r + \cdots + r$, where r is added n times. For a non-empty subset $T \subseteq R$ we denote by $\langle T \rangle$ the subsemiring generated by T, i.e., the smallest set containing T which is closed under addition and multiplication. An *ideal* of R is a subset $I \subseteq R$ such that for all $a, b \in I$, $s \in R$ we have $a + b, sa, as \in I$. Clearly, every ideal is a subsemiring. With $E(R)$ we denote the set of multiplicative idempotents of R, i.e., those $e \in R$ with $e^2 = e$. Note that for every multiplicative idempotent $e \in E(R)$, eRe is a subsemiring of R in which the multiplicative structure is a monoid. Let $\mathbb{B}_2 = (\{0, 1\}, \vee, \wedge)$ be the *Boolean semiring*.

A crucial definition in this paper is that of a $\{0, 1\}$-*free* semiring. This is a semiring R which does *not* contain a subsemiring T with an additive identity 0 and a multiplicative identity $1 \neq 0$. Note that it is not required that 0 is absorbing in T, i.e., $a \cdot 0 = 0 \cdot a = 0$ for all $a \in T$. The class of $\{0, 1\}$-free *finite* semirings has several characterizations:

▶ **Lemma 1.** *For a finite semiring R, the following are equivalent:*
1. *R is not $\{0, 1\}$-free.*
2. *$\mathbb{B}_2$ or $\mathbb{Z}_d$ for some $d \geq 2$ is a subsemiring of R.*
3. *$\mathbb{B}_2$ or $\mathbb{Z}_d$ for some $d \geq 2$ is a homomorphic image of a subsemiring of R.*
4. *There exist elements $0, 1 \in R$ such that $0 \neq 1$, $0 + 0 = 0$, $0 + 1 = 1$, $0 \cdot 1 = 1 \cdot 0 = 0 \cdot 0 = 0$, and $1 \cdot 1 = 1$ (but $1 + 1 \neq 1$ is possible).*

As a consequence of Lemma 1 (point 4), one can check in time $O(n^2)$ for a semiring of size n whether it is $\{0, 1\}$-free. We will not need this fact, since in our setting the semiring will be always fixed, i.e., not part of the input. Moreover, the class of all $\{0, 1\}$-free semirings is a pseudo-variety of finite semirings, i.e., it is closed under taking subsemirings (this is trivial), taking homomorphic images (by point 3), and direct products. For the latter, assume that $R \times R'$ is not $\{0, 1\}$-free. Hence, there exists a subsemiring T of $R \times R'$ with an additive zero $(0, 0')$ and a multiplicative one $(1, 1') \neq (0, 0')$. W.l.o.g. assume that $0 \neq 1$. Then the projection $\pi_1(T)$ onto the first component is a subsemiring of R, where 0 is an additive identity and $1 \neq 0$ is a multiplicative identity.

4 Circuit evaluation and main results

We define circuits over general algebraic structures. Let $\mathcal{A} = (D, f_1, \ldots, f_k)$ be an algebraic structure. A *circuit* over $\mathcal{A}$ is a triple $\mathcal{C} = (V, A_0, \mathsf{rhs})$ where V is a finite set of *gates*, $A_0 \in V$ is the *output gate* and rhs (for right-hand side) is a function that assigns to each gate $A \in V$ an element $a \in D$ or an expression of the form $f_i(A_1, \ldots, A_n)$, where $n = n_i$ and $A_1, \ldots, A_n \in V$ are called the *input gates for A*. Moreover, the binary relation $\{(A, B) \in V \times V \mid A \text{ is an input gate for } B\}$ must be acyclic. The reflexive and transitive closure of it is a partial order on V that we denote with $\leq_{\mathcal{C}}$. Every gate A evaluates to an element $[A]_{\mathcal{C}} \in A$ in the natural way: If $\mathsf{rhs}(A) = a \in D$, then $[A]_{\mathcal{C}} = a$ and if $\mathsf{rhs}(A) = f_i(A_1, \ldots, A_n)$ then $[A]_{\mathcal{C}} = f_i([A_1]_{\mathcal{C}}, \ldots, [A_n]_{\mathcal{C}})$. Moreover, we define $[\mathcal{C}] = [A_0]_{\mathcal{C}}$ (the value computed by $\mathcal{C}$). If the circuit $\mathcal{C}$ is clear from the context, we also write $[A]$ instead of $[A]_{\mathcal{C}}$. Two circuits $\mathcal{C}_1$ and $\mathcal{C}_2$ over the structure $\mathcal{A}$ are *equivalent* if $[\mathcal{C}_1] = [\mathcal{C}_2]$.

Sometimes we also use circuits without an output gate; such a circuit is just a pair (V, rhs). A subcircuit of $\mathcal{C}$ is the restriction of $\mathcal{C}$ to a downwards closed (w.r.t. $\leq_{\mathcal{C}}$) subset of V. A gate A with $\mathsf{rhs}(A) = f_i(A_1, \dots, A_n)$ is called an *inner gate*, otherwise it is an *input gate* of $\mathcal{C}$. Quite often, we view a circuit as a directed acyclic graph, where the inner nodes are labelled with an operations f_i, and the leaf nodes are labelled with elements from D. In our proofs, it is sometimes convenient to allow arbitrary terms built from $V \cup D$ using the operations $f_1, \dots, f_k$ in right-hand sides. For instance, over a semiring $(R, +, \cdot)$ we might have $\mathsf{rhs}(A) = s \cdot B \cdot t + C + s$ for $s, t \in R$ and $B, C \in V$. A circuit is in *normal form*, if all right-hand sides are of the form $a \in D$ or $f_i(A_1, \dots, A_n)$ with $A_1, \dots, A_n \in V$. We will make use of the following simple fact:

▶ **Lemma 2.** *A circuit can be transformed in logspace into an equivalent normal form circuit.*

The *circuit evaluation problem* $\mathsf{CEP}(\mathcal{A})$ for some algebraic structure $\mathcal{A}$ (say a semigroup or a semiring) is the following computational problem:

Input: A circuit $\mathcal{C}$ over $\mathcal{A}$ and an element $a \in D$ from its domain.
Output: Decide whether $[\mathcal{C}] = a$.

Note that for a finite structure $\mathcal{A}$, $\mathsf{CEP}(\mathcal{A})$ is basically equivalent to its computation variant, where one actually computes the output value $[\mathcal{C}]$ of the circuit: if $\mathsf{CEP}(\mathcal{A})$ belongs to a complexity class C, then the computation variant belongs to $\mathsf{AC}^0(\mathsf{C})$, and if the latter belongs to $\mathsf{AC}^0(\mathsf{C})$ then $\mathsf{CEP}(\mathcal{A})$ belongs to the decision fragment of $\mathsf{AC}^0(\mathsf{C})$.

Clearly, for every finite structure the circuit evaluation problem can be solved in polynomial time by evaluating all gates along the partial order $\leq_{\mathcal{C}}$. Ladner's classical P-completeness result for the Boolean circuit value problem [19] can be stated as follows:

▶ **Theorem 3** ([19])**.** $\mathsf{CEP}(\mathbb{B}_2)$ *is* P*-complete.*

For semigroups, the following dichotomy was shown in [8]:

▶ **Theorem 4** ([8])**.** *Let S be a finite semigroup.*
- *If S is aperiodic, then* $\mathsf{CEP}(S)$ *is in* NL*.*
- *If S is solvable, then* $\mathsf{CEP}(S)$ *belongs to* DET*.*
- *If S is not solvable, then* $\mathsf{CEP}(S)$ *is* P*-complete.*

In fact, in [8], the authors use the original definition $\mathsf{DET} = \mathsf{NC}^1(\mathsf{det})$ of Cook. But the arguments in [8] actually show that for a finite solvable semigroup, $\mathsf{CEP}(S)$ belongs to $\mathsf{AC}^0(\mathsf{det})$ (which is our definition of DET). Moreover, in [8], Theorem 4 is only shown for monoids, but the extension to semigroups is straightforward: If the finite semigroup S has a non-solvable subgroup, then $\mathsf{CEP}(S)$ is P-complete, since the circuit evaluation problem for a non-solvable finite group is P-complete. On the other hand, if S is solvable (resp., aperiodic), then also the monoid S^1 is solvable (resp., aperiodic). This holds, since the subgroups of S^1 are exactly the subgroups of S together with $\{1\}$. Hence, $\mathsf{CEP}(S^1)$ is in DET (resp., NL), which implies that $\mathsf{CEP}(S)$ is in DET (resp., NL).

Let us fix a *finite* semiring $R = (R, +, \cdot)$ for the rest of the paper. Note that $\mathsf{CEP}(R_+)$ (resp., $\mathsf{CEP}(R_\bullet)$) is the restriction of $\mathsf{CEP}(R)$ to circuits without multiplication (resp., addition) gates. Since every commutative semigroup is solvable, Theorem 4 implies that $\mathsf{CEP}(R_+)$ belongs to DET. The main result of this paper is:

▶ **Theorem 5.** *If the finite semiring R is $\{0,1\}$-free, then the problem* $\mathsf{CEP}(R)$ *belongs to the class* $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+), \mathsf{CEP}(R_\bullet))$*. Otherwise* $\mathsf{CEP}(R)$ *is* P*-complete.*

Note that CEP(R) can also be P-complete for a $\{0,1\}$-free semiring (namely in the case that CEP($R_\bullet$) is P-complete) and that $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+), \mathsf{CEP}(R_\bullet)) = \mathsf{AC}^0(\mathsf{CEP}(R_+), \mathsf{CEP}(R_\bullet))$ whenever CEP(R_+) or CEP($R_\bullet$) is NL-hard. For example, this is the case, if R_+ or $R_\bullet$ is an aperiodic nontrivial monoid [8, Proposition 4.14] (for aperiodic nontrivial monoids one can easily reduce the NL-complete of graph reachability problem to the circuit value problem).

The P-hardness statement in Theorem 5 is easy to show:

▶ **Lemma 6.** *If the finite semiring R is not $\{0,1\}$-free, then* CEP(R) *is* P*-complete.*

Proof. By Lemma 1, R contains either $\mathbb{B}_2$ or $\mathbb{Z}_d$ for some $d \geq 2$. In the former case, P-hardness follows from Ladner's theorem. Furthermore, one can reduce the P-complete Boolean circuit value problem over $\{0, 1, \wedge, \neg\}$ to CEP($\mathbb{Z}_d$): A gate $z = x \wedge y$ is replaced by $z = x \cdot y$ and a gate $y = \neg x$ is replaced by $y = 1 + (d-1) \cdot x$. ◀

Theorem 4 and 5 yield the following corollaries:

▶ **Corollary 7.** *Let R be a finite semiring.*

- *If R is $\{0,1\}$-free and $R_\bullet$ and R_+ are aperiodic, then* CEP(R) *belongs to* NL.
- *If R is $\{0,1\}$-free and $R_\bullet$ is solvable, then* CEP(R) *belongs to* DET.
- *If R is not $\{0,1\}$-free or $R_\bullet$ is not solvable, then* CEP(R) *is* P*-complete.*

Let us present an application of Corollary 7.

▶ **Example 8.** An important semigroup construction found in the literature is the power construction. For a finite semigroup S one defines the *power semiring* $\mathcal{P}(S) = (2^S \setminus \{\emptyset\}, \cup, \cdot)$ with the multiplication $A \cdot B = \{ab \mid a \in A, b \in B\}$. Notice that if one includes the empty set, then the semiring would not be $\{0,1\}$-free: Take an idempotent $e \in S$. Then $\emptyset$ and $\{e\}$ form a copy of $\mathbb{B}_2$. Hence, the circuit evaluation problem is P-complete.

Let us further assume that S is a monoid with identity 1 (the general case will be considered below). If S contains an idempotent $e \neq 1$ then also $\mathcal{P}(S)$ is not $\{0,1\}$-free: $\{e\}$ and $\{1, e\}$ form a copy of $\mathbb{B}_2$. On the other hand, if 1 is the unique idempotent of S, then S must be a group G. Assume that G is solvable; otherwise $\mathcal{P}(G)_\bullet$ is not solvable as well and has a P-complete circuit evaluation problem by Theorem 4. It is not hard to show that the subgroups of $\mathcal{P}(G)_\bullet$ correspond to the quotient groups of subgroups of G; see also [21]. Since G is solvable and the class of solvable groups is closed under taking subgroups and quotients, $\mathcal{P}(G)_\bullet$ is a solvable monoid. Moreover $\mathcal{P}(G)$ is $\{0,1\}$-free: Otherwise, Lemma 1 implies that there are non-empty subsets $A, B \subseteq G$ such that $A \neq B$, $A \cup B = B$ (and thus $A \subsetneq B$), $AB = BA = A^2 = A$, and $B^2 = B$. Hence, B is a subgroup of G and $A \subseteq B$. But then $B = AB = A$, which is a contradiction. By Corollary 7, CEP($\mathcal{P}(G)$) for a finite solvable group G belongs to DET.

Let us now classify the complexity of CEP($\mathcal{P}(S)$) for arbitrary semigroups S. A semigroup S is a *local group* if for all $e \in E(S)$ the local monoid eSe is a group. In a finite local group S of size n the minimal semigroup ideal is $S^n = SE(S)S$ [2, Proposition 2.3].

▶ **Theorem 9.** *Let S be a finite semigroup. If S is a local group and solvable, then* CEP($\mathcal{P}(S)$) *belongs to* DET. *Otherwise* CEP($\mathcal{P}(S)$) *is* P*-complete.*

Proof. If S is a solvable local group, then the multiplicative semigroup $\mathcal{P}(S)_\bullet$ is solvable as well [5, Corollary 2.7]. It remains to show that the semiring $\mathcal{P}(S)$ is $\{0,1\}$-free. Towards a contradiction assume that $\mathcal{P}(S)$ is not $\{0,1\}$-free. By Lemma 1, there exist non-empty sets $A \subsetneq B \subseteq S$ such that $AB = BA = A^2 = A$ and $B^2 = B$. Hence, B is a subsemigroup of S, which is also a local group, and A is a semigroup ideal in B. Since the minimal semigroup ideal of B is B^n for $n = |B|$ and $B^n = B$, we obtain $A = B$, which is a contradiction.

If S is not a local group, then there exists a local monoid eSe which is not a group and hence contains an idempotent $f \neq e$. Since $\{\{f\}, \{e, f\}\}$ forms a copy of $\mathbb{B}_2$ it follows that $\mathsf{CEP}(\mathcal{P}(S))$ is P-complete. Finally, if S is not solvable, then also $\mathcal{P}(S)$ is not solvable and $\mathsf{CEP}(\mathcal{P}(S))$ is P-complete by Theorem 4. ◀

5 Proof of Theorem 5

The proof of Theorem 5 will proceed in two steps. In the first step we reduce the problem to evaluating circuits in which the computation admits a type-function defined in the following. In the second step, we show how to evaluate such circuits.

▶ **Definition 10.** Let $E = E(R)$ be the set of multiplicative idempotents. Let $\mathcal{C} = (V, \mathsf{rhs})$ be a circuit in normal form such that $[A]_\mathcal{C} \in ERE$ for all $A \in V$. A type-function for $\mathcal{C}$ is a mapping $\mathsf{type} : V \to E \times E$ such that for all gates $A \in V$:

- If $\mathsf{type}(A) = (e, f)$, then $[A]_\mathcal{C} \in eRf$.
- If A is an addition gate with $\mathsf{rhs}(A) = B + C$, then $\mathsf{type}(A) = \mathsf{type}(B) = \mathsf{type}(C)$.
- If A is a multiplication gate with $\mathsf{rhs}(A) = B \cdot C$, $\mathsf{type}(B) = (e, e')$, and $\mathsf{type}(C) = (f', f)$, then $\mathsf{type}(A) = (e, f)$.

A circuit is called *type admitting* if it admits a type-function.

A function $\alpha : R^m \to R$ ($m \geq 0$) is called *affine* if there are $a_1, b_1, \ldots, a_m, b_m, c \in R$ such that $\alpha(x_1, \ldots, x_m) = \sum_{i=1}^m a_i x_i b_i + c$ or $\alpha(x_1, \ldots, x_m) = \sum_{i=1}^m a_i x_i b_i$ for all $x_1, \ldots, x_m \in R$. We represent this affine function by the tuple $(a_1, b_1, \ldots, a_m, b_m, c)$ or $(a_1, b_1, \ldots, a_m, b_m)$. Theorem 5 is an immediate corollary of the following two propositions (and the obvious fact that an affine function with a constant number of inputs can be evaluated in AC^0).

▶ **Proposition 11.** *Given a circuit $\mathcal{C}$ over the finite semiring R, one can compute in* $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+))$

- *an affine function $\alpha : R^m \to R$ for some $0 \leq m \leq |R|^4$,*
- *a type admitting circuit $\mathcal{C}' = (V', \mathsf{rhs}')$, and*
- *a list of gates $A_1, \ldots, A_m \in V'$ such that $[\mathcal{C}] = \alpha([A_1]_{\mathcal{C}'}, \ldots, [A_m]_{\mathcal{C}'})$.*

▶ **Proposition 12.** *If R is $\{0, 1\}$-free, then the restriction of* $\mathsf{CEP}(R)$ *to type admitting circuits is in* $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+), \mathsf{CEP}(R_\bullet))$.

Notice that in Proposition 12 we do not need explicitly a type function as part of the input. Moreover, it is not clear how to test efficiently whether a circuit is type admitting. On the other hand, this is not a problem for us, since we will apply Proposition 12 only to circuits resulting from Proposition 11, which are type admitting by construction.

5.1 Step 1: Reduction to typing admitting circuits

In this section, we sketch a proof of Proposition 11. Let $\mathcal{C}$ be a circuit in normal form over our fixed finite semiring $(R, +, \cdot)$ of size $n = |R| \geq 2$ (the case $n = 1$ is trivial). Let $E = E(R)$. Note that $R^n = RER$ is closed under multiplication with elements from R. Thus, $\langle R^n \rangle$ is an ideal. Every element of $\langle R^n \rangle$ is a finite sum of elements from R^n.

In a first step, we compute from $\mathcal{C}$ in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+))$ a semiring element r and a circuit $\mathcal{D}$ over the subsemiring $\langle R^n \rangle = \langle RER \rangle$ such that $[\mathcal{C}] = r + [\mathcal{D}]$, where r or $\mathcal{D}$ (but not both) can be missing. For the proof of this, we interpret the circuit $\mathcal{C}$ over the *free* *semiring* $\mathbb{N}[R]$. It consists of all mappings $f : R^+ \to \mathbb{N}$ (where R^+ is the set of non-empty words over the alphabet R) such that $\mathsf{supp}(f) := \{w \in R^+ \mid f(w) \neq 0\}$ (the support of f) is finite and non-empty. We view an element $f \in \mathbb{N}[R]$ as a polynomial $\sum_{w \in \mathsf{supp}(f)} f(w) \cdot w$, where R is a set of non-commuting variables. Addition and multiplication of such non-commuting polynomials is defined as usual. Words $w \in \mathsf{supp}(f)$ are also called *monomials* of f. Let $h : \mathbb{N}[R] \to R$ be the canonical evaluation homomorphism, which evaluates a given non-commutative polynomial in R. Thereby a monomial $w = a_1 a_2 \cdots a_n$ is mapped to the corresponding product in R. Since a semiring is not assumed to have a multiplicative identity (resp., additive identity), we have to exclude the empty word from $\mathsf{supp}(f)$ for every $f \in \mathbb{N}[R]$ (resp., exclude the mapping f with $\mathsf{supp}(f) = \emptyset$ from $\mathbb{N}[R]$).

The idea is to split each polynomial computed in a gate A into two parts: Those monomials (i.e., non-empty words over R) that have length $< n = |R|$ (called the short part of A) and those monomials that have length $\geq n$ (called the long part of A). Of course the short (resp. long) part of a gate can be empty. We then compute from the circuit $\mathcal{C}$ the following data: (i) for every gate A the h-image of the short part of A if it is non-empty and (ii) a circuit over $\langle R^n \rangle$ that contains for every gate A of $\mathcal{C}$ the h-image of its long part (if it exists). For (i), we need oracle access to $\mathsf{CEP}(R_+)$. Oracle access to NL is needed to compute those gates whose short (resp., long) part is non-empty.

In a second step, we compute from a circuit $\mathcal{D}$ over $\langle RER \rangle$ a type admitting circuit $\mathcal{C}'$ such that the value of $\mathcal{D}$ is an affine combination of certain gate values in $\mathcal{C}'$. The main idea is the following: In the circuit $\mathcal{D}$ all input values are sums of elements of the form *set* ($e \in E$, $s, t \in R$), which we can write as se^3t. Hence, if we evaluate the circuit freely in $\mathbb{N}[R]$, then every monomial that arises at a gate A is of the form $segft$, where g starts (resp., ends) with the symbol $e \in E$ (resp., $f \in E$) and $s, t \in R$. Let P_A is the set of all tuples (s, e, f, t) such that at gate A a monomial of the form $segft$ arises. One can show that P_A can be computed in $\mathsf{AC}^0(\mathsf{NL})$. The circuit $\mathcal{C}'$ contains for every $(s, e, f, t) \in P_A$ a gate $A_{s,e,f,t}$ that computes the sum of all monomials g such that $segft$ is a monomial that appears at gate A. The type of gate $A_{s,e,f,t}$ is (e, f). Moreover, $[A]_{\mathcal{D}}$ is equal to $\sum_{(s,e,f,t) \in P_A} (se)[A_{s,e,f,t}]_{\mathcal{C}'}(ft)$. This shows that $[\mathcal{D}]$ is indeed an affine combination of certain gate values in $\mathcal{C}'$.

5.2 Step 2: A parallel evaluation algorithm for type admitting circuits

In this section we prove Proposition 12. We present a parallel evaluation algorithm for type admitting circuits. This algorithm terminates after at most $|R|$ rounds, if R has a so-called rank-function, which we define first. As before, let $E = E(R)$.

▶ **Definition 13.** We call a function $\mathsf{rank} : R \to \mathbb{N} \setminus \{0\}$ a *rank-function* for R if it satisfies the following conditions for all $a, b \in R$:

1. $\mathsf{rank}(a) \leq \mathsf{rank}(a \circ b)$ and $\mathsf{rank}(b) \leq \mathsf{rank}(a \circ b)$ for $\circ \in \{+, \cdot\}$.
2. If $a, b \in eRf$ for some $e, f \in E$ and $\mathsf{rank}(a) = \mathsf{rank}(a + b)$, then $a = a + b$.

If $R_\bullet$ is a monoid, then one can choose $e = 1 = f$ in the second condition in Definition 13, which is therefore equivalent to: If $\mathsf{rank}(a) = \mathsf{rank}(a + b)$ for $a, b \in R$, then $a = a + b$.

▶ **Example 14** (Example 8 continued)**.** Let G be a finite group and consider the semiring $\mathcal{P}(G)$. One can verify that the function $A \mapsto |A|$, where $\emptyset \neq A \subseteq G$, is a rank-function for $\mathcal{P}(G)$. On the other hand, if S is a finite semigroup, which is not a group, then S cannot be cancellative. Assume that $ab = ac$ for $a, b, c \in S$ with $b \neq c$. Then $\{a\} \cdot \{b, c\} = \{ab\}$. This shows that the function $A \mapsto |A|$ is not a rank-function for $\mathcal{P}(S)$.

▶ **Theorem 15.** *If the finite semiring R has a rank-function* rank*, then the restriction of* $\mathsf{CEP}(R)$ *to type admitting circuits belongs to* $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+), \mathsf{CEP}(R_\bullet))$.

Proof. Let $\mathcal{C} = (V, A_0, \mathsf{rhs})$ be a circuit with the type function type. We present an algorithm which partially evaluates the circuit in a constant number of phases, where each phase can be carried out in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+), \mathsf{CEP}(R_\bullet))$ and the following invariant is preserved:

Invariant: After phase k all gates A with $\mathsf{rank}([A]_\mathcal{C}) \leq k$ are evaluated, i.e., are input gates in phase $k+1$ onwards.

Initially, i.e., for $k = 0$, the invariant holds, since 0 is not in the range of the rank-function. After $\max\{\mathsf{rank}(a) \mid a \in R\}$ (which is a constant) many phases, the output gate A_0 is evaluated. We present phase k of the algorithm, assuming that the invariant holds after phase $k-1$. Thus, all gates A with $\mathsf{rank}([A]_\mathcal{C}) < k$ of the current circuit $\mathcal{C}$ are input gates. In phase k we evaluate all gates A with $\mathsf{rank}([A]_\mathcal{C}) = k$. For this, we proceed in two steps:

Step 1. As a first step the algorithm evaluates all subcircuits that only contain addition and input gates. This maintains the invariant and is possible in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(R_+))$. After this step, every addition-gate A has at least one inner input gate, which we denote by $\mathsf{inner}(A)$ (if both input gates are inner gates, then choose one arbitrarily). The NL-oracle access is needed to compute the set of all gates A for which no multiplication gate $B \leq_\mathcal{C} A$ exists.

Step 2. Define the multiplicative circuit $\mathcal{C}' = (V, A_0, \mathsf{rhs}')$ by

$$\mathsf{rhs}'(A) = \begin{cases} \mathsf{inner}(A) & \text{if } A \text{ is an addition-gate,} \\ \mathsf{rhs}(A) & \text{if } A \text{ is a multiplication gate or input gate.} \end{cases} \tag{1}$$

The circuit $\mathcal{C}'$ can be brought in logspace into normal form by Lemma 2 and then evaluated in $\mathsf{AC}^0(\mathsf{CEP}(R_\bullet))$. A gate $A \in V$ is called *locally correct* if (i) A is an input gate or multiplication gate of $\mathcal{C}$, or (ii) A is an addition gate of $\mathcal{C}$ with $\mathsf{rhs}(A) = B + C$ and $[A]_{\mathcal{C}'} = [B]_{\mathcal{C}'} + [C]_{\mathcal{C}'}$. We compute the set $W := \{A \in V \mid \text{ all gates } B \text{ with } B \leq_\mathcal{C} A \text{ are locally correct}\}$ in $\mathsf{AC}^0(\mathsf{NL})$. A simple induction shows that for all $A \in W$ we have $[A]_\mathcal{C} = [A]_{\mathcal{C}'}$. Hence we can set $\mathsf{rhs}(A) = [A]_{\mathcal{C}'}$ for all $A \in W$. This concludes phase k of the algorithm.

To prove that the invariant holds after phase k, we show that for each gate $A \in V$ with $\mathsf{rank}([A]_\mathcal{C}) \leq k$ we have $A \in W$. This is shown by induction over the depth of A in $\mathcal{C}$. Assume that $\mathsf{rank}([A]_\mathcal{C}) \leq k$. By the first condition from Definition 13, all gates $B <_\mathcal{C} A$ satisfy $\mathsf{rank}([B]_\mathcal{C}) \leq k$. Thus, the induction hypothesis yields $B \in W$ and hence $[B]_\mathcal{C} = [B]_{\mathcal{C}'}$ for all gates $B <_\mathcal{C} A$. It remains to show that A is locally correct, which is clear if A is an input gate or a multiplication gate. So assume that $\mathsf{rhs}(A) = B + C$ where $B = \mathsf{inner}(A)$, which implies $[A]_{\mathcal{C}'} = [B]_{\mathcal{C}'}$ by (1). Since B is an inner gate, which is not evaluated after phase $k-1$, it holds that $\mathsf{rank}([B]_\mathcal{C}) \geq k$ and therefore $\mathsf{rank}([A]_\mathcal{C}) = \mathsf{rank}([B]_\mathcal{C}) = k$. By Definition 10 there exist idempotents $e, f \in E$ with $\mathsf{type}(B) = \mathsf{type}(C) = (e, f)$ and thus $[B]_\mathcal{C}, [C]_\mathcal{C} \in eRf$. The second condition from Definition 13 implies that $[A]_\mathcal{C} = [B]_\mathcal{C} + [C]_\mathcal{C} = [B]_\mathcal{C}$. We finally get $[A]_{\mathcal{C}'} = [B]_{\mathcal{C}'} = [B]_\mathcal{C} = [A]_\mathcal{C} = [B]_\mathcal{C} + [C]_\mathcal{C} = [B]_{\mathcal{C}'} + [C]_{\mathcal{C}'}$. Therefore A is locally correct. ◀

▶ **Example 16** (Example 8 continued)**.** Figure 1 shows a circuit $\mathcal{C}$ over the power semiring $\mathcal{P}(G)$ of the group $G = (\mathbb{Z}_5, +)$. Recall from Example 14 that the function $A \mapsto |A|$ is a rank function for $\mathcal{P}(G)$. We illustrate one phase of the algorithm. All gates A with $\mathsf{rank}([A]) < 3$ are evaluated in the circuit $\mathcal{C}$ shown on the left. The goal is to evaluate all gates A with $\mathsf{rank}([A]) = 3$. The first step would be to evaluate maximal $\cup$-circuits, which is already done.

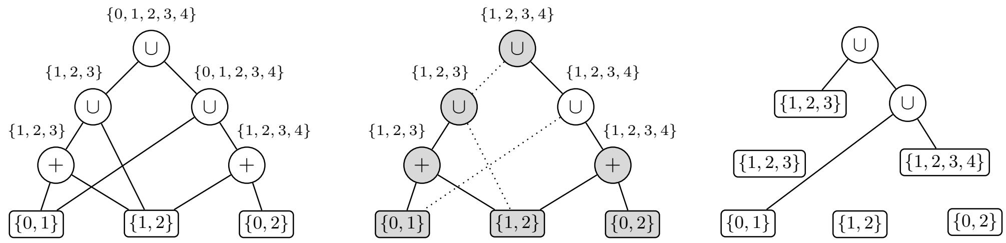

Figure 1 The parallel evaluation algorithm over the power semiring $\mathcal{P}(\mathbb{Z}_5)$.

In the second step the circuit $\mathcal{C}'$ (shown in the middle) from the proof of Theorem 15 is computed and evaluated using the oracle for $\mathsf{CEP}(\mathbb{Z}_5, +)$. The dotted wires do not belong to the circuit $\mathcal{C}'$. All locally correct gates are shaded. Note that the output gate is locally correct but its right child is not locally correct. All other shaded gates form a downwards closed set, which is the set W from the proof. These gates can be evaluated such that in the resulting circuit (shown on the right) all gates which evaluate to elements of rank 3 are evaluated.

To show Proposition 12, it remains to equip every finite $\{0, 1\}$-free semiring with a rank-function.

▶ **Lemma 17.** *If R is $\{0, 1\}$-free and $e, f \in E(R)$ are such that $ef = fe = f + f = f$, then $e + f = f$.*

Proof. With $f = 0$, $e+f = 1$ all equations from Lemma 1 (point 4) hold; hence $e+f = f$. ◀

▶ **Lemma 18.** *If the finite semiring R is $\{0, 1\}$-free, then R has a rank-function.*

Proof. For $a, b \in R$ we define $a \preceq b$ if b can be obtained from a by iterated additions and left- and right-multiplications of elements from R. This is equivalent to the existence of $\ell, r, c \in R$ such that $b = \ell a r + c$, where each of the elements ℓ, r, c can be missing. Since $\preceq$ is a preorder on R, there is a function $\mathsf{rank} : R \to \mathbb{N} \setminus \{0\}$ such that for all $a, b \in R$ we have (i) $\mathsf{rank}(a) = \mathsf{rank}(b)$ if and only if $a \preceq b \preceq a$, and (ii) $\mathsf{rank}(a) \leq \mathsf{rank}(b)$ if $a \preceq b$.

We claim that rank satisfies the conditions of Definition 13. The first condition is clear, since $a \preceq a + b$ and $a, b \preceq ab$. For the second condition, let $e, f \in E$, $a, b \in eRf$ such that $\mathsf{rank}(a + b) = \mathsf{rank}(a)$, which is equivalent to $a + b \preceq a$. Assume that $a = \ell(a + b)r + c = \ell ar + \ell br + c$ for some $\ell, r, c \in R$ (the case without c can be handled in the same way). Since $a = eaf$ and $b = ebf$, we have $a = \ell e(a + b)fr + c$ and hence we can assume that ℓ and r are not missing. Moreover, $a = eaf = (e\ell e)(a + b)(frf) + (ecf)$, so we can assume that $\ell = e\ell e$ and $r = frf$. After m applications of $a = \ell ar + \ell br + c$ we get

$$a = \ell^m a r^m + \sum_{i=1}^{m} \ell^i b r^i + \sum_{i=0}^{m-1} \ell^i c r^i. \tag{2}$$

Let $n \geq 1$ such that nx is additively idempotent and x^n is multiplicatively idempotent for all $x \in R$. Hence nx^n is both additively and multiplicatively idempotent for all $x \in R$. If we choose $m = n^2$, the right hand side of (2) contains the partial sum $P := \sum_{i=1}^{n} \ell^{in} b r^{in}$. Furthermore, $e(n\ell^n) = (n\ell^n)e = n\ell^n$ and $f(nr^n) = (nr^n)f = nr^n$. Therefore, Lemma 17 implies that $n\ell^n = n\ell^n + e$ and $nr^n = nr^n + f$, and hence:

$$\begin{aligned} P = \sum_{i=1}^{n} \ell^{in} b r^{in} &= n(\ell^n b r^n) = n^2(\ell^n b r^n) = (n\ell^n) b (nr^n) = (n\ell^n + e) b (nr^n) \\ &= (n\ell^n) b (nr^n) + eb(nr^n) = (n\ell^n) b (nr^n) + eb(nr^n + f) \\ &= (n\ell^n) b (nr^n) + eb(nr^n) + ebf = \left(\sum_{i=1}^{n} \ell^{in} b r^{in} \right) + b = P + b. \end{aligned}$$

Thus, the partial sum P in (2) can be replaced by $P + b$, which shows $a = a + b$. ◀

6 An application to formal language theory

In this section we briefly report on an application of Corollary 7 to a particular intersection non-emptiness problem. We assume some familiarity with context-free grammars. A circuit over the free monoid Σ^* can be seen as a context-free grammar producing exactly one word. Such a circuit is also called a *straight-line program*, briefly SLP. It is an acyclic context-free grammar $\mathcal{H}$ that contains for every non-terminal A exactly one rule with left-hand side A. We denote with $\mathsf{val}_{\mathcal{H}}(A)$ the unique terminal word that can be derived from A.

For an alphabet Σ and a language $L \subseteq \Sigma^*$, the *intersection non-emptiness problem for* L, denoted by $\mathsf{CFG\text{-}IP}(L, \Sigma)$, is the following decision problem: Given a context-free grammar $\mathcal{G}$ over Σ, does $L(\mathcal{G}) \cap L \neq \emptyset$ hold? For every regular language L, this problem belongs to P: One constructs in polynomial time a context-free grammar for $L(\mathcal{G}) \cap L$ from $\mathcal{G}$ and a finite automaton for L and tests this grammar for emptiness, which is possible in polynomial time. However, testing emptiness of a given context-free language is P-complete. An easy reduction shows that the problem $\mathsf{CFG\text{-}IP}(L, \Sigma)$ is P-complete for every $L \neq \emptyset$:

▶ **Theorem 19.** *For every non-empty language* $L \subseteq \Sigma^*$, $\mathsf{CFG\text{-}IP}(L, \Sigma)$ *is* P*-complete.*

By Theorem 19 we have to put some restriction on context-free grammars in order to get NC-algorithms for intersection non-emptiness. It turns out that productivity of all non-terminals is the right assumption. Thus, we require that every non-terminal A is productive, i.e., a terminal word can be derived from A. In order to avoid a promise problem (testing productivity of a non-terminal is P-complete [16]) we add to the input grammar $\mathcal{G}$ an SLP $\mathcal{H}$, which *uniformizes* $\mathcal{G}$ in the sense that $\mathcal{H}$ contains for every non-terminal A exactly one rule $A \to \alpha$ from $\mathcal{G}$. Hence, the word $\mathsf{val}_{\mathcal{H}}(A)$ is a witness for the productivity of A. For instance, a uniformizing SLP for the grammar $S \to SS \mid aSb \mid A$, $A \to aA \mid B$, $B \to bB \mid b$ would be $S \to A$, $A \to B$, $B \to b$.

We define the following restriction $\mathsf{PCFG\text{-}IP}(L, \Sigma)$ of $\mathsf{CFG\text{-}IP}(L, \Sigma)$: Given a productive context-free grammar $\mathcal{G}$ over Σ and a uniformizing SLP $\mathcal{H}$ for $\mathcal{G}$, does $L(\mathcal{G}) \cap L \neq \emptyset$ hold? The theorem below classifies regular languages $L \subseteq \Sigma^*$ by the complexity of $\mathsf{PCFG\text{-}IP}(L, \Sigma)$. To do this we use the standard notion of the syntactic monoid M_L of L (which is a finite monoid for L regular). There is a surjective morphism $h : \Sigma^* \to L$ and a subset $F \subseteq M_L$ such that $L = h^{-1}(M_L)$. Let us fix the regular language $L \subseteq \Sigma^*$, $M = M_L$, $h : \Sigma^* \to M$ and $F \subseteq M$. Define the equivalence relation $\sim_F$ on $\mathcal{P}(M)$ by: $A_1 \sim_F A_2$ $(A_1, A_2 \in \mathcal{P}(M))$ if and only if $\forall \ell, r \in M \colon \ell A_1 r \cap F \neq \emptyset \iff \ell A_2 r \cap F \neq \emptyset$. It can be shown that $\sim_F$ is a congruence relation. In particular, $\mathcal{P}(M)/{\sim_F}$ is a semiring.

▶ **Theorem 20.** $\mathsf{PCFG\text{-}IP}(L, \Sigma)$ *is equivalent to* $\mathsf{CEP}(\mathcal{P}(M)/{\sim_F})$ *with respect to constant depth reductions. Hence,* $\mathsf{PCFG\text{-}IP}(L, \Sigma)$ *is in* DET *(resp.,* NL*) if* $(\mathcal{P}(M)/{\sim_F})_\bullet$ *is solvable (resp., aperiodic) and* $\mathcal{P}(M)/{\sim_F}$ *is* $\{0,1\}$*-free; otherwise* $\mathsf{PCFG\text{-}IP}(L, \Sigma)$ *is* P*-complete.*

As an application of Theorem 20 one can show that $\mathsf{PCFG\text{-}IP}(L, \Sigma)$ is in NL for every language of the form $L = \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \ldots a_k \Sigma^*$ for $a_1, \ldots, a_k \in \Sigma$.

7 Conclusion and outlook

We proved a dichotomy result for the circuit evaluation problem for finite semirings: If (i) the semiring has no subsemiring with an additive and multiplicative identity and both are different and (ii) the multiplicative subsemigroup is solvable, then the circuit evaluation problem is in $\mathsf{DET} \subseteq \mathsf{NC}^2$, otherwise it is P-complete.

The ultimate goal would be to obtain such a dichotomy for all finite algebraic structures. One might ask whether for every finite algebraic structure $\mathcal{A}$, $\mathsf{CEP}(\mathcal{A})$ is P-complete or in NC. It is known that under the assumption $\mathsf{P} \neq \mathsf{NC}$ there exist problems in $\mathsf{P} \setminus \mathsf{NC}$ that are not P-complete [32]. In [7] it is shown that every circuit evaluation problem $\mathsf{CEP}(\mathcal{A})$ is equivalent to a circuit evaluation problem $\mathsf{CEP}(A, \circ)$, where $\circ$ is a binary operation.

Acknowledgement. We are grateful to Ben Steinberg for fruitful discussions and to Volker Diekert for pointing out to us the proof of the implication ($3 \Rightarrow 4$) in the proof of Lemma 1.

References

1 Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theor. Comput. Sci.*, 209(1-2):47–86, 1998.

2 Jorge Almeida, Stuart Margolis, Benjamin Steinberg, and Mikhail Volkov. Representation theory of finite semigroups, semigroup radicals and formal language theory. *Transactions of the American Mathematical Society*, 361(3):1429–1461, 2009.

3 Carme Àlvarez, José L. Balcázar, and Birgit Jenner. Functional oracle queries as a measure of parallel time. In *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1991*, volume 480 of *Lecture Notes in Computer Science*, pages 422–433. Springer, 1991.

4 Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.

5 Karl Auinger and Benjamin Steinberg. Constructing divisions into power groups. *Theoretical Computer Science*, 341(1–3):1–21, 2005.

6 Martin Beaudry and Markus Holzer. The complexity of tensor circuit evaluation. *Computational Complexity*, 16(1):60–111, 2007.

7 Martin Beaudry and Pierre McKenzie. Circuits, matrices, and nonassociative computation. *Journal of Computer and System Sciences*, 50(3):441–455, 1995.

8 Martin Beaudry, Pierre McKenzie, Pierre Péladeau, and Denis Thérien. Finite monoids: From word to circuit evaluation. *SIAM Journal on Computing*, 26(1):138–152, 1997.

9 S. Buss, S. Cook, A. Gupta, and V. Ramachandran. An optimal parallel algorithm for formula evaluation. *SIAM Journal on Computing*, 21(4):755–780, 1992.

10 Stephan A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64:2–22, 1985.

11 Stephen A. Cook and Lila Fontes. Formal theories for linear algebra. *Logical Methods in Computer Science*, 8(1), 2012.

12 Moses Ganardi, Danny Hucke, Daniel König, and Markus Lohrey. Circuit evaluation for finite semirings. Technical report, arXiv.org, 2016. `http://arxiv.org/abs/1602.04560`.

13 Jonathan S. Golan. *Semirings and their Applications*. Springer, 1999.

14 Leslie M. Goldschlager. The monotone and planar circuit value problems are log space complete for P. *SIGACT News*, 9(2):25–99, 1977.

15 Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory.* Oxford University Press, 1995.

16 Neil D. Jones and William T. Laaser. Complete problems for deterministic polynomial time. *Theor. Comput. Sci.*, 3(1):105–117, 1976.

17 Daniel König and Markus Lohrey. Evaluating matrix circuits. In *Proceedings of the 21st International Conference on Computing and Combinatorics, COCOON 2015*, volume 9198 of *Lecture Notes in Computer Science*, pages 235–248. Springer, 2015.

18 S. Rao Kosaraju. On parallel evaluation of classes of circuits. In *Proceedings of the 10th Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 1990*, volume 472 of *Lecture Notes in Computer Science*, pages 232–237. Springer, 1990.

19 Richard E. Ladner. The circuit value problem is log space complete for P. *SIGACT News*, 7(1):18–20, 1975.

20 Markus Lohrey. *The Compressed Word Problem for Groups.* SpringerBriefs in Mathematics. Springer, 2014.

21 Donald J. McCarthy and David L. Hayes. Subgroups of the power semigroup of a group. *Journal of Combinatorial Theory, Series A*, 14(2):173–186, 1973.

22 Pierre McKenzie and Klaus W. Wagner. The complexity of membership problems for circuits over sets of natural numbers. *Computational Complexity*, 16(3):211–244, 2007.

23 Gary L. Miller, Vijaya Ramachandran, and Erich Kaltofen. Efficient parallel evaluation of straight-line code and arithmetic circuits. *SIAM J. Comput.*, 17(4):687–695, 1988.

24 Gary L. Miller and Shang-Hua Teng. Tree-based parallel algorithm design. *Algorithmica*, 19(4):369–389, 1997. doi:10.1007/PL00009179.

25 Gary L. Miller and Shang-Hua Teng. The dynamic parallel complexity of computational circuits. *SIAM J. Comput.*, 28(5):1664–1688, 1999.

26 Cristopher Moore, Denis Thérien, François Lemieux, Joshua Berman, and Arthur Drisko. Circuits and expressions with nonassociative gates. *J. Comput. Syst. Sci.*, 60(2):368–394, 2000.

27 Vijaya Ramachandran and Honghua Yang. An efficient parallel algorithm for the general planar monotone circuit value problem. *SIAM J. Comput.*, 25(2):312–339, 1996.

28 John Rhodes and Benjamin Steinberg. *The q-theory of Finite Semigroups.* Springer, 2008.

29 Alexander A. Rubtsov and Mikhail N. Vyalyi. Regular realizability problems and context-free languages. In *Proceedings of the 17th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2015*, volume 9118 of *Lecture Notes in Computer Science*, pages 256–267. Springer, 2015.

30 Stephen D. Travers. The complexity of membership problems for circuits over sets of integers. *Theor. Comput. Sci.*, 369(1-3):211–229, 2006.

31 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983.

32 Heribert Vollmer. The gap-language-technique revisited. In *Proceedings of the 4th Workshop on Computer Science Logic, CSL'90*, volume 533 of *Lecture Notes in Computer Science*, pages 389–399. Springer, 1990.

33 Heribert Vollmer. *Introduction to Circuit Complexity.* Springer, 1999.