comput. complex. (2023) 32:10

1016-3328/23/020001-76
*published online* September 13, 2023
https://doi.org/10.1007/s00037-023-00244-x

computational complexity

# PARALLEL ALGORITHMS FOR POWER CIRCUITS AND THE WORD PROBLEM OF THE BAUMSLAG GROUP

Caroline Mattes and Armin Weiß

**Abstract.** Power circuits have been introduced in 2012 by Myasnikov, Ushakov and Won as a data structure for non-elementarily compressed integers supporting the arithmetic operations addition and $(x, y) \mapsto x \cdot 2^y$. The same authors applied power circuits to give a polynomial time solution to the word problem of the Baumslag group, which has a non-elementary Dehn function.
In this work, we examine power circuits and the word problem of the Baumslag group under parallel complexity aspects. In particular, we establish that the word problem of the Baumslag group can be solved in NC—even though one of the essential steps is to compare two integers given by power circuits and this, in general, is shown to be P-complete. The key observation is that the depth of the occurring power circuits is logarithmic and such power circuits can be compared in NC.

**Keywords.** Word problem, Baumslag group, power circuit, parallel complexity

**Subject classification.** Primary 20F10; Secondary 20-08, 68Q25

## 1. Introduction

The *word problem* of a finitely generated group $G$ is as follows: does a given word over the generators of $G$ represent the identity of $G$? It was first studied by Dehn (1911) as one of the basic algorithmic problems in group theory. Already in the 1950 s,

Novikov (1955) and Boone (1959) succeeded to construct finitely presented groups with an undecidable word problem. Nevertheless, many natural classes of groups have an (efficiently) decidable word problem—most prominently the class of linear groups (groups embeddable into a matrix group over some field): their word problem is in LOGSPACE (Lipton & Zalcstein 1977; Simon 1979)—hence, in particular, in NC, i.e., decidable by Boolean circuits of polynomial size and polylogarithmic depth (or, equivalently, decidable in polylogarithmic time using polynomially many processors).

There are various other results on word problems of groups in small parallel complexity classes defined by circuits. For example, the word problems of solvable linear groups are even in $\mathsf{TC}^0$ (constant depth with threshold gates) (König & Lohrey 2018) and the word problems of Baumslag–Solitar groups and of right-angled Artin groups are $\mathsf{AC}^0$-Turing-reducible to the word problem of a non-abelian free group (Kausch 2017; Weiß 2016). Moreover, Thompson's groups are co-context-free (Lehnert & Schweitzer 2007) and hyperbolic groups have word problem in LOGCFL (Lohrey 2005). All these complexity classes are contained within NC. On the other hand, there are also finitely presented groups with a decidable word problem but with arbitrarily high complexity (Sapir *et al.* 2002).

A mysterious class of groups under this point of view are one-relator groups, i.e., groups that can be written as a free group modulo a normal subgroup generated by a single element (*relator*). Magnus (1932) showed that one-relator groups have a decidable word problem; his algorithm is called the Magnus breakdown procedure (see also Lyndon & Schupp 2001; Magnus *et al.* 2004). Nevertheless, the complexity remains an open problem—although it is not even clear whether the word problems of one-relator groups are solvable in elementary time, the question has been raised whether they are actually decidable in polynomial time (Baumslag *et al.* 2002).

In 1969 Gilbert Baumslag found an example of a one-relator group with certain remarkable properties:

$$\mathbf{G}_{1,2} = \left\langle a, b \mid bab^{-1}a = a^2bab^{-1} \right\rangle .$$

It is infinite and non-abelian, but all its finite quotients are cyclic

and, thus, it is not residually finite (Baumslag 1969). Moreover, Gersten (1991) showed that the Dehn function of $\mathbf{G}_{1,2}$ is non-elementary and Platonov (2004) made this more precise by proving that it is (roughly) $\tau(\log n)$ where $\tau(0) = 1$ and $\tau(i+1) = 2^{\tau(i)}$ for $i \geq 0$ is the tower function (note that he calls the Baumslag–Gersten group). Since the Dehn function gives an upper bound on the complexity of the word problem, the Baumslag group was a candidate for a group with a very difficult word problem. Indeed, when applying the Magnus breakdown procedure to an input word of length $n$, one obtains as intermediate results words of the form $v_1^{x_1} \cdots v_m^{x_m}$ where $v_i \in \{a, b, bab^{-1}\}$, $x_i \in \mathbb{Z}$, and $m \leq n$. The issue is that the $x_i$ might grow up to $\tau(\log n)$; hence, this algorithm has non-elementary running time. However, as foreseen by the above-mentioned conjecture, Myasnikov, Ushakov & Won (2011) succeeded to show that the word problem of $\mathbf{G}_{1,2}$ is, indeed, decidable in polynomial time. Their crucial contribution was to introduce so-called *power circuits* for compressing the $x_i$ in the description above (Myasnikov *et al.* 2012).

Roughly speaking, a *power circuit* is a directed acyclic graph (a dag) where the edges are labeled by $\pm 1$. One can define an evaluation of a vertex $P$ as two raised to the power of the (signed) sum of the successors of $P$. Note that this way the value $\tau(n)$ of the tower function can be represented by an $n+1$-vertex power circuit—thus, power circuits allow for a non-elementary compression. The crucial feature for the application to the Baumslag group is that power circuits not only efficiently support the operations $+$, $-$, and $(x, y) \mapsto x \cdot 2^y$, but also the test whether $x = y$ or $x < y$ for two integers represented by power circuits can be done in polynomial time. The main technical part of the comparison algorithm is the so-called reduction process, which computes a certain normal form for power circuits.

Based on these striking results, Diekert, Laun & Ushakov (2013) improved the algorithm for power circuit reduction and managed to decrease the running time for the word problem of the Baumslag group from $\mathcal{O}(n^7)$ down to $\mathcal{O}(n^3)$. They also describe a polynomial-time algorithm for the word problem of the famous Higman group $H_4$ introduced by Higman (1951). These algorithms have

been implemented in C++ (see Myasnikov & Ushakov 2004–2013). Subsequently, more applications of power circuits to these groups emerged: Laun (2014) gave a polynomial time solution to the word problem in generalized Baumslag and Higman groups, Diekert, Myasnikov & Weiß (2016) showed that the conjugacy problem of the Baumslag group is strongly generically in $\mathsf{P}$ and Baker (2020) does the same for the conjugacy problem of the Higman group. Here "generically" roughly means that the algorithm works for most inputs (for details on the concept of generic complexity, see Kapovich *et al.* 2003).

Other examples where compression techniques lead to efficient algorithms in group theory can be found, e.g., in Dison *et al.* (2018) or Lohrey (2014, Theorems 4.6, 4.8 and 4.9). Finally, notice that Miasnikov & Nikolaev (2020) examine the word search problem for the Baumslag group using parametrized complexity.

**Contribution.** The aim of this work is to analyze power circuits and the word problem of the Baumslag group under the view of parallel (circuit) complexity. For doing so, we first examine so-called *compact* representations of integers (already considered, e.g., by Güntzer & Paul (1987); Jedwab & Mitchell (1989); Reitwiesner (1960); Shallit (1993) under different names) and show that ordinary binary representations can be converted into compact representations by constant depth circuits (i.e., in $\mathsf{AC}^0$—see Section 3). We apply this result in the power circuit reduction process, which is the main technical contribution of this paper. While Diekert, Laun & Ushakov (2013); Myasnikov, Ushakov & Won (2012) give only polynomial time algorithms, we present a more refined method and analyze it in terms of parametrized circuit complexity. The parameter here is the depth $D$ of the power circuit. More precisely, we present threshold circuits of depth $\mathcal{O}(D)$ for power circuit reduction—implying our first main result:

Proposition A. *The problem of comparing two integers given by power circuits of logarithmic depth is in* $\mathsf{TC}^1$ *(decidable by logarithmic depth, polynomial-size threshold circuits).*

We then analyze the word problem of the Baumslag group carefully. A crucial step is to show that all appearing power circuits have logarithmic depth. Using Proposition A we succeed to describe a $\mathsf{TC}^1$ algorithm for computing the Britton reduction of $uv$ given that $u$ and $v$ are already Britton-reduced (Britton reductions are the basic step in the Magnus breakdown procedure—see Section 5 for a definition). This leads to the following result:

THEOREM B. *The word problem of the Baumslag group* $\mathbf{G}_{1,2}$ *is in* $\mathsf{TC}^2$.

In the final part of the paper, we prove lower bounds on comparison in power circuits and, thus, on power circuit reduction. In particular, this emphasizes the relevance of Proposition A and shows that our parametrized analysis of power circuit reduction is essentially the best one can hope for. Moreover, Theorem C highlights the importance of the logarithmic depth bound for the power circuits appearing during the proof of Theorem B.

THEOREM C. *The problem of comparing two integers given by power circuits is* $\mathsf{P}$*-complete.*

Power circuits can be seen in the broader context of arithmetic circuits and arithmetic complexity. Thus, results on power circuits also give further insight into these arithmetic circuits. Notice that Semenov (1983) showed that the corresponding logic over natural numbers with addition and $2^x$ is decidable. In Proposition 4.11 we show that, indeed, for every power circuit with a marking $M$ there is an arithmetic circuit of polynomial size with +-, −-, and $2^x$-gates evaluating to the same number and vice versa. Moreover, the transformation between these two models can be done efficiently.

This work is the full and extended version of the conference publication (Mattes & Weiß 2021). Besides giving full proofs of all results, here we explore the connections between power circuits and arithmetic circuits with +-, −-, and $2^x$-gates and in Theorem 6.1 we give a refined variant of Theorem C which also yields hardness results for power circuits of logarithmic depth.

## 2. Notation and preliminaries

**General notions.** We use standard $\mathcal{O}$-notation for functions from $\mathbb{N}$ to nonnegative reals $\mathbb{R}^{\geq 0}$, see, e.g., (Cormen *et al.* 2009). Throughout, the logarithm log is with respect to base two. The *tower function* $\tau\colon \mathbb{N} \to \mathbb{N}$ is defined by $\tau(0) = 1$ and $\tau(i+1) = 2^{\tau(i)}$ for $i \geq 0$. It is primitive recursive, but $\tau(6)$ written in binary cannot be stored in the memory of any conceivable real-world computer. Moreover, we set $\log^*(n) = \min\{i \mid \tau(i) \geq n\}$.

The *support* of a function $f\colon X \to \mathbb{R}$ is denoted by $\sigma(f) = \{x \in X \mid f(x) \neq 0\}$. Furthermore, we denote the interval of integers $\{i, \ldots, j\} \subseteq \mathbb{Z}$ by $[i\,..\,j]$ and we define $[n] = [0\,..\,n-1]$. We write $\mathbb{Z}[1/2] = \{p/2^q \in \mathbb{Q} \mid p, q \in \mathbb{Z}\}$ for the set of dyadic fractions.

Let $\Sigma$ be a set. The set of all words over $\Sigma$ is denoted by $\Sigma^* = \bigcup_{n\in\mathbb{N}} \Sigma^n$. The length of a word $w \in \Sigma^*$ is denoted by $|w|$. A dag is a directed acyclic graph. For a dag $\Gamma$ we write $\mathrm{depth}(\Gamma)$ for its depth, which is the length (number of edges) of a longest path in $\Gamma$.

### 2.1. Complexity.

We assume the reader to be familiar with the complexity classes LOGSPACE and P (polynomial time); see, e.g., (Arora & Barak 2009) for details. Most of the time, however, we use circuit complexity within NC.

Throughout, we assume that inputs to and outputs of functions $f$ are encoded over the binary alphabet $\{0,1\}$. Let $k \in \mathbb{N}$. A function $f$ is in $\mathsf{AC}^k$ if there is a family of polynomial-size Boolean circuits of depth $\mathcal{O}(\log^k n)$ (where $n$ is the input length) computing $f$. More precisely, a Boolean circuit is a dag (directed acyclic graph) where the vertices are either input gates $x_1, \ldots, x_n$, or Not-, And-, or Or-gates. Some of these gates are marked as output gates $o_1, \ldots, o_m$. All gates may have unbounded fan-in (i.e., there is no bound on the number of incoming wires). A function $f : \{0,1\}^* \to \{0,1\}^*$ belongs to $\mathsf{AC}^k$ if there exists a family $(C_n)_{n\in\mathbb{N}}$ of Boolean circuits such that, for each $x \in \{0,1\}^*$, the $i$-th output gate $o_i$ of the circuit $C_n$ evaluates to the $i$-th bit of $f(x)$ when assigning $x = x_1 \cdots x_n$ to the input gates of $C_n$ (where $n = |x|$). Moreover, $C_n$ may contain at most $n^{\mathcal{O}(1)}$ gates and have depth $\mathcal{O}(\log^k n)$. Here, the depth of a circuit is the length of the longest

path from an input gate to an output gate. Likewise, a language $L$ is in $\mathsf{AC}^k$ if its characteristic function is $\mathsf{AC}^k$-computable.

The class $\mathsf{TC}^k$ is defined analogously with the difference that also MAJORITY gates are allowed (a MAJORITY gate outputs 1 if its input contains more 1s than 0s). Moreover, $\mathsf{NC} = \bigcup_{k\geq 0} \mathsf{TC}^k = \bigcup_{k\geq 0} \mathsf{AC}^k$. For more details on circuits, we refer to Vollmer (1999). Our algorithms (or circuits) rely on two basic building blocks which can be done in $\mathsf{TC}^0$:

EXAMPLE 2.1. Iterated addition is the following problem:

| | |
|---|---|
| Input: | $n$ numbers $A_1, \ldots, A_n$ each having $n$ bits |
| Output: | $\sum_{i=1}^{n} A_i$ |

This is well-known to be in $\mathsf{TC}^0$. ◊

EXAMPLE 2.2. Let $(k_1, v_1), \ldots, (k_n, v_n)$ be a list of $n$ key-value pairs $(k_i, v_i)$ equipped with a total order on the keys $k_i$ such that it can be decided in $\mathsf{TC}^0$ whether $k_i < k_j$. Then the problem of sorting the list according to the keys is in $\mathsf{TC}^0$: the desired output is a list $(k_{\pi(1)}, v_{\pi(1)}), \ldots, (k_{\pi(n)}, v_{\pi(n)})$ for some permutation $\pi$ such that $k_{\pi(i)} \leq k_{\pi(j)}$ for all $i < j$.

We briefly describe a circuit family to do so: The first layer compares all pairs of keys $k_i, k_j$ in parallel. For all $i$ and $j$ the next layer computes a Boolean value $P(i, j)$ which is true if and only if $|\{\ell \mid k_\ell < k_i\}| = j$. The latter is computed by iterated addition. As a final step the $j$-th output pair is set to $(k_i, v_i)$ if and only if $P(i, j)$ is true. ◊

REMARK 2.3. *The class* $\mathsf{NC}$ *is contained in* $\mathsf{P}$ *if we consider uniform circuits. A family of circuits is called* $\mathsf{LOGSPACE}$*-uniform (or simply uniform) if the function* $1^n \mapsto C_n$ *is computable in* $\mathsf{LOGSPACE}$ *(where* $1^n$ *is the string consisting of* $n$ *ones and* $C_n$ *is given as some reasonable encoding). Be aware that for classes below* $\mathsf{LOGSPACE}$ *usually even stronger uniformity conditions are imposed. In order not to overload the presentation, throughout, we state all our results in the non-uniform case—all uniformity considerations are left to the reader.*

**Parametrized circuit complexity.** In our work we also need some parametrized version of the classes $\mathsf{TC}^k$, which we call *depth-parametrized* $\mathsf{TC}^k$. Let $\mathrm{par}\colon \{0,1\}^* \to \mathbb{N}$ (called the *parameter*). Consider a family of circuits $(C_{n,D})_{n,D\in\mathbb{N}}$ such that $C_{n,D}$ contains at most $n^{\mathcal{O}(1)}$ gates (independently of $D$) and has depth $\mathcal{O}(D \cdot \log^k n)$ (note that in our application the parameter $D$ is bounded by the input size $n$, which means that letting the size of $C_{n,D}$ be a polynomial in both $n$ and $D$ would not change the actual class). We say that a function $f\colon \{0,1\}^* \to \{0,1\}^*$ is computed by $(C_{n,D})_{n,D\in\mathbb{N}}$ if for all $n$ and $D$ and all $x \in \{0,1\}^n$ with $\mathrm{par}(x) \leq D$ the circuit $C_{n,D}$ evaluates to $f(x)$ on input $x$. As above, a language $L$ is said to be accepted by such a circuit family if its characteristic function is computed by $(C_{n,D})_{n,D\in\mathbb{N}}$. We define $\mathsf{DepParaTC}^k$ as the class of functions (resp. languages) for which there are such parametrizations $\mathrm{par}\colon \{0,1\}^* \to \mathbb{N}$ and families of circuits $(C_{n,D})_{n,D\in\mathbb{N}}$. Note that this is not a standard definition—but it perfectly fits our purposes.

Lemma 2.4. *Let $C > 0$, $k, \ell \in \mathbb{N}$ and $\mathrm{par}\colon \{0,1\}^* \to \mathbb{N}$ such that $\left\{ w \in \{0,1\}^* \;\middle|\; \mathrm{par}(w) \leq C \cdot \lfloor \log |w| \rfloor^\ell \right\} \in \mathsf{TC}^{k+\ell}$ and $f : \{0,1\}^* \to \{0,1\}^*$ is in $\mathsf{DepParaTC}^k$ (parametrized by* par*). Let*

$$\tilde{f} : \{0,1\}^* \to \{0,1\}^* \cup \{\bot\}$$

*defined by*

$$\tilde{f}(w) = \begin{cases} f(w), & \text{if } \mathrm{par}(w) \leq C \cdot \lfloor \log |w| \rfloor^\ell \\ \bot, & \text{otherwise.} \end{cases}$$

*Then $\tilde{f} \in \mathsf{TC}^{k+\ell}$.*

Proof. Let $w \in \{0,1\}^n$ be some input. First decide whether $\mathrm{par}(w) \leq C \cdot \lfloor \log n \rfloor^\ell$ (by the hypothesis this is in $\mathsf{TC}^{k+\ell}$). If yes, the circuit $C_{n,C\cdot\lfloor \log n\rfloor^\ell}$ can be used to calculate $f(w)$; if no, the output is $\bot$. Clearly, the combined circuit has polynomial size. Its depth is $\mathcal{O}(\log^{k+\ell} n)$ for the first step plus $\mathcal{O}(C \cdot \lfloor \log n \rfloor^\ell \cdot \log^k n) = \mathcal{O}(\log^{k+\ell} n)$ for the second step. Hence, we have obtained a $\mathsf{TC}^{k+\ell}$ circuit. □

We introduce this parametrized $\mathsf{TC}^k$ classes because later for computing reduced power circuits we apply a non-constant number of $\mathsf{TC}^0$ computations $f$ one after each other. The number of these computations is the depth of the power circuit. The crucial step is to show that after any number of applications of $f$, the output is still polynomially bounded. Putting things together, we obtain a $\mathsf{DepParaTC}^0$ computation parametrized by the depth of the power circuit. Let us formalize this idea:

Denote the $i$-fold composition of $f$ by $f^{(i)}$ (i.e., $f^{(0)}$ is the identity function and $f^{(i)} = f \circ f^{(i-1)}$ for $i \geq 1$).

In order to allow circuits to compute functions having outputs of different lengths for inputs of the same length, we can assume that each output gate also carries an enable bit (or equivalently we can think that there is an additional padding symbol in the output alphabet).

LEMMA 2.5. *Let $f\colon \{0,1\}^* \to \{0,1\}^*$ be $\mathsf{TC}^k$-computable such that for all $x \in \{0,1\}^*$ there is some $\omega_x \leq |x|$ with $f^{(\omega_x)}(x) = f^{(\omega_x+1)}(x)$. Further, assume that there is some polynomial $p$ such that for all $x \in \{0,1\}^*$ and for all $i \in \mathbb{N}$ we have $\left|f^{(i)}(x)\right| \leq p(|x|)$.*

*Then $x \mapsto f^{(\omega_x)}(x)$ is in $\mathsf{DepParaTC}^k$ where the parameter $\mathrm{par}\colon \{0,1\}^* \to \mathbb{N}$ is defined by $x \mapsto \omega_x$.*

PROOF. Let $(C_n)_{n\in\mathbb{N}}$ be the family of $\mathsf{TC}^k$ circuits computing $f$. We construct a new family of circuits $(C_{n,\omega})_{n,\omega\in\mathbb{N}}$. Let $\tilde{C}_m$ be a circuit consisting of $C_i$ for all $i \in [0\,..\,m]$ in parallel. We can compose $\tilde{C}_{p(n)} \circ C_n$ by feeding the outputs of $C_n$ into the $C_i$ (as part of $\tilde{C}_{p(n)}$) with the appropriate number of input bits. By iterating this, we obtain a circuit $\tilde{C}_{p(n)} \circ \cdots \circ \tilde{C}_{p(n)} \circ C_n$ consisting of $C_n$ followed by $\omega-1$ layers of $\tilde{C}_{p(n)}$. By the hypothesis of the lemma, we can assume $\omega \leq n$, so this circuit contains at most $n \cdot p(n) \cdot s_{p(n)}$ gates where $s_{p(n)}$ is the maximum number of gates in $C_i$ for $i \leq p(n)$. Moreover, the depth of $\tilde{C}_{p(n)}$ is $\mathcal{O}(\log^k p(n)) = \mathcal{O}(\log^k n)$, so the depth of $C_{n,\omega}$ is $\mathcal{O}(\omega \cdot \log^k(n))$. □

**2.2. Power circuits.** Consider a pair $(\Gamma, \delta)$ where $\Gamma$ is a set of $n$ vertices and $\delta$ is a mapping $\delta\colon \Gamma \times \Gamma \to \{-1, 0, +1\}$. The support of $\delta$ is the subset $\sigma(\delta) \subseteq \Gamma \times \Gamma$ consisting of those $(P, Q)$ with

$\delta(P,Q) \neq 0$. Thus, $(\Gamma, \sigma(\delta))$ is a directed graph without multi-edges. Throughout we require that $(\Gamma, \sigma(\delta))$ is acyclic—i.e., it is a dag. In particular, $\delta(P,P) = 0$ for all vertices $P$. A *marking* is a mapping $M\colon \Gamma \to \{-1,0,+1\}$. Each node $P \in \Gamma$ is associated in a natural way with a marking $\Lambda_P\colon \Gamma \to \{-1,0,+1\},\ Q \mapsto \delta(P,Q)$ called its successor marking. The support of $\Lambda_P$ consists of the target nodes of outgoing edges from $P$. We denote the marking with empty support by $\emptyset$. We define the *evaluation* $\varepsilon(P)$ of a node ($\varepsilon(M)$ of a marking resp.) bottom-up in the dag by induction:

$$\begin{aligned} \varepsilon(\emptyset) &= 0, \\ \varepsilon(P) &= 2^{\varepsilon(\Lambda_P)} && \text{for a node } P, \\ \varepsilon(M) &= \sum_P M(P)\varepsilon(P) && \text{for a marking } M. \end{aligned}$$

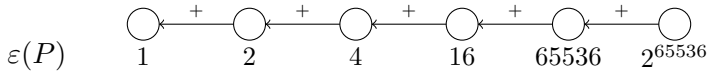
We have $\varepsilon(\Lambda_P) = \log_2(\varepsilon(P))$, i.e., the marking $\Lambda_P$ plays the role of a logarithm. Note that leaves (nodes of out-degree 0) evaluate to 1 and every node evaluates to a positive real number. However, we are only interested in the case that all nodes evaluate to integers:

Definition 2.6. *A* power circuit *is a pair* $(\Gamma, \delta)$ *with* $\delta\colon \Gamma \times \Gamma \to \{-1,0,+1\}$ *such that* $(\Gamma, \sigma(\delta))$ *is a dag and all nodes evaluate to some positive natural number in* $2^{\mathbb{N}}$.

The size of a power circuit is the number of nodes $|\Gamma|$. By abuse of language, we also simply call $\Gamma$ a power circuit and suppress $\delta$ whenever it is clear. If $M$ is a marking on $\Gamma$ and $S \subseteq \Gamma$, we write $M|_S$ for the restriction of $M$ to $S$. Let $(\Gamma', \delta')$ be a power circuit, $\Gamma \subseteq \Gamma'$, $\delta = \delta'|_{\Gamma\times\Gamma}$, and $\delta'|_{\Gamma\times(\Gamma'\setminus\Gamma)} = 0$. Then $(\Gamma, \delta)$ itself is a power circuit. We call it a *sub-power circuit* and denote this by $(\Gamma, \delta) \leq (\Gamma', \delta')$ or, if $\delta$ is clear, by $\Gamma \leq \Gamma'$.

If $M$ is a marking on $S \subseteq \Gamma$, we extend $M$ to $\Gamma$ by setting $M(P) = 0$ for $P \in \Gamma\setminus S$. With this convention, every marking on $\Gamma$ also can be seen as a marking on $\Gamma'$ if $\Gamma \leq \Gamma'$.

Example 2.7. A power circuit of size $n+1$ can realize $\tau(n)$ since a directed path of $n+1$ nodes represents $\tau(n)$ as the evaluation of the last node. The following power circuit realizes $\tau(5)$ using 6 nodes:




◊

Example 2.8. We can represent every integer in the range $[-2^n - 1, 2^n - 1]$ as the evaluation of some marking in a power circuit with node set $\{P_0, \ldots, P_{n-1}\}$ with $\varepsilon(P_i) = 2^i$ for $i \in [n]$. Thus, we can convert the binary notation of an $n$-bit integer into a power circuit with $n$ vertices, $\mathcal{O}(n \log n)$ edges (each successor marking requires at most $\lfloor \log n \rfloor + 1$ edges) and depth at most $\log^* n$. For an example of a marking representing the integer 23, see Figure 2.1. ◊

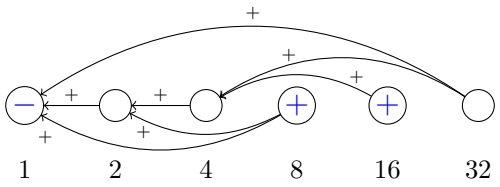



Figure 2.1: Each integer $z \in [-63\,..\,63]$ can be represented by a marking in the following power circuit. The marking given in blue is representing the number 23.

Definition 2.9. *We call a marking $M$* compact *if for all $P, Q \in \sigma(M)$ with $P \neq Q$ we have $|\varepsilon(\Lambda_P) - \varepsilon(\Lambda_Q)| \geq 2$. A* reduced power circuit *of size $n$ is a power circuit $(\Gamma, \delta)$ with $\Gamma$ given as a sorted list $\Gamma = (P_0, \ldots, P_{n-1})$ such that all successor markings are compact and $\varepsilon(P_i) < \varepsilon(P_j)$ whenever $i < j$. In particular, all nodes have pairwise distinct evaluations.*

It turns out to be crucial that the nodes in $\Gamma$ are sorted by their values. Still, sometimes it is convenient to treat $\Gamma$ as a set—we write $P \in \Gamma$ or $S \subseteq \Gamma$ with the obvious meaning. Whenever convenient we assume that $\varepsilon(\Lambda_{P_i}) = \infty$ for $i \geq n$.

Notice that Diekert, Laun & Ushakov (2013) use a bit-vector to store which nodes have successor markings differing by one for the data structure of a reduced power circuit—we will compute this information on-the-fly whenever needed. For more details on power circuits, see (Diekert *et al.* 2013; Myasnikov *et al.* 2012).

Remark 2.10. *If $(\Gamma, \delta)$ is a reduced power circuit such that $\Gamma = (P_0, \ldots, P_{n-1})$, we have $\delta(P_i, P_j) = 0$ for $j \geq i$. Thus, the order on $\Gamma$ by evaluations is also a topological order on the dag $(\Gamma, \sigma(\delta))$.*

## 3. Compact signed-digit representations

In this section we will show that for every binary number we can efficiently calculate a so-called unique compact representation. This will be a crucial tool for the power circuit reduction process.

Definition 3.1. *(i) A sequence $B = (b_0, \ldots, b_{m-1})$ with $b_i \in \{-1, 0, +1\}$ for $i \in [m]$ is called a* signed-digit representation *of $\mathrm{val}(B) = \sum_{i=0}^{m-1} b_i \cdot 2^i \in \mathbb{Z}$.*

*(ii) The digit length of $B = (b_0, \ldots, b_{m-1})$ is the maximal $i$ with $b_{i-1} \neq 0$.*

*(iii) The sequence $B = (b_0, \ldots, b_{m-1})$ is called* compact *if $b_i b_{i-1} = 0$ for all $i \in [1\,..\,m-1]$ (i.e., no two successive digits are nonzero).*

A nonnegative binary number is the special case of a signed-digit representation where all $b_i$ are 0 or 1 (note that, in general, they are not compact). Also negative binary numbers can be seen as special cases of signed-digit representations—though the precise form depends on the representation: A negative number given as two's complement is a signed-digit representation where the most-significant digit is a –1 and the other nonzero digits are 1s; a negative signed magnitude representation can be viewed as signed-digit representation where all nonzero digits are –1s. In particular, every integer can be represented as a signed-digit representation. While, in general, a signed-digit representation for an integer is not unique, each integer has a unique *compact* signed-digit representation: previously to Myasnikov *et al.* (2012), signed-digit representations have been introduced and investigated by Güntzer & Paul (1987); Jedwab & Mitchell (1989); Reitwiesner (1960); Shallit (1993) under different names. These papers already showed that each integer can be represented by a unique compact signed-digit representation and gave polynomial time algorithms for computing

them. We improve upon this by showing that they can actually be computed in $\mathsf{AC}^0$. For an alternative proof of Theorem 3.2, we could use that compact signed-digit representations can be computed using a finite state transducer (Shallit 1993).

Note that by setting $b_i = 0$ for $i \geq m$, one can extend every signed-digit representation $B = (b_0, \ldots, b_{m-1})$ to an arbitrarily long or infinite sequence. By doing so, $\mathrm{val}(B)$ and the digit length of $B$ do not change.

**Computing compact signed-digit representations.** In the following we will, among other things, show that for every binary number $A$ there exists such a compact signed-digit representation $B$ of $A$ and that $B$ is unique with this property. We start with the existence and the complexity of calculating $B$. While Myasnikov *et al.* (2012, Section 2.1) described a linear-time algorithm for calculating $B$, we aim for optimizing the parallel complexity.

THEOREM 3.2. *The following is in* $\mathsf{AC}^0$*:*

| | |
|---|---|
| Input: | *A binary number* $A = (a_0, \ldots, a_{m-1})$. |
| Output: | *A compact signed-digit representation of* $A$. |

Notice that Theorem 3.2 implies that every integer has a compact signed-digit representation. Moreover, be aware that, clearly, the theorem is only true if we choose suitable encodings—in particular, we assume that the three values $-1, 0, 1$ are all encoded using two bits.

PROOF. Let $A = (a_0, \ldots, a_{m-1})$ be a binary number. For $i \geq m$ we set $a_i = 0$. We view the $a_i$ as Boolean variables and aim for constructing (almost) Boolean formulas for the compact representation. Since the digits of a compact representation are from the set $\{-1, 0, 1\}$, we treat the Boolean values $0, 1$ as a subset of the integers and we will mix Boolean operations ($\wedge$, $\vee$, $\oplus$) with arithmetic operations ($+$, $\cdot$). Here $\oplus$ denotes the *exclusive or*, which is addition modulo two.

For $i \geq 0$ we define

$$c_i = \bigvee_{1 \leq j \leq i} \Big( (a_j \wedge a_{j-1}) \wedge \bigwedge_{j < k \leq i} (a_k \vee a_{k-1}) \Big),$$

$$b_i = (a_i \oplus c_i) \cdot (-1)^{a_{i+1}}.$$

Moreover, we set $B = (b_0, \ldots, b_{m-1}, b_m)$. Observe that $c_0 = 0$ and, hence, $b_0 = a_0 \cdot (-1)^{a_1}$. Furthermore, $b_m = c_m$ and $b_i = c_i = 0$ for $i \geq m + 1$.

REMARK 3.3. *It is clear that the $c_i$ can be computed in $\mathsf{AC}^0$ and so the same holds for the $b_i$. This implies that on input of $A = (a_0, \ldots, a_{m-1})$, one can compute $B$ in $\mathsf{AC}^0$. By the very definition as a Boolean formula, it is clear that it is actually in uniform $\mathsf{AC}^0$ (see Remark 2.3).*

Thus, in order to prove Theorem 3.2, it remains to show that $B = (b_0, \ldots, b_m)$ is compact and that $\mathrm{val}(B) = \mathrm{val}(A)$.

CLAIM 3.4. *The $c_i$ satisfy the following recurrence:*

- $c_0 = 0$
- $c_i = (a_i \wedge a_{i-1}) \vee \big(c_{i-1} \wedge (a_i \vee a_{i-1})\big)$ *for* $i \geq 1$.

PROOF. For $i = 0$, the claim holds because the empty disjunction is equal to 0. Now we assume that $i \geq 1$ and that the recurrence holds for $i - 1$. We set $X_j = a_j \wedge a_{j-1}$ and $Y_j = a_j \vee a_{j-1}$. Then we obtain

$$c_i = \bigvee_{1 \leq j \leq i} \left( X_j \wedge \left( \bigwedge_{j < k \leq i} Y_k \right) \right) = X_i \vee \left( \bigvee_{1 \leq j \leq i-1} X_j \wedge \left( \bigwedge_{j < k \leq i} Y_k \right) \right)$$

$$= X_i \vee \left( \left( \bigvee_{1 \leq j \leq i-1} X_j \wedge \left( \bigwedge_{j < k \leq i-1} Y_k \right) \right) \wedge Y_i \right)$$

$$= X_i \vee (c_{i-1} \wedge Y_i)$$

This proves the claim. □

CLAIM 3.5. *Let $a_i, b_i$ and $c_i$ be as above. Then for all $k \geq 0$ we have*

$$a_k + c_k = b_k + 2c_{k+1}.$$

PROOF. Claim 3.4 implies $c_{k+1} = (a_{k+1} \wedge a_k) \vee (c_k \wedge (a_{k+1} \vee a_k))$. Thus, we can express both $b_k$ and $c_{k+1}$ in terms of $a_k$, $a_{k+1}$ and $c_k$. This leads us to the following table:

| $a_k$ | $a_{k+1}$ | $c_k$ | $b_k$ | $c_{k+1}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | –1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | –1 | 1 |
| 1 | 1 | 1 | 0 | 1 |

If we now take the values in the table as integer values and put them into the above equation, we see that the equation holds in all cases. □

CLAIM 3.6. *Let $a_i, b_i$ and $c_i$ be as above. Then for all $k \geq 0$ we have*

$$\sum_{i=0}^{k} 2^i a_i = 2^{k+1} c_{k+1} + \sum_{i=0}^{k} 2^i b_i.$$

PROOF. We use induction on $k$. Since $c_0 = 0$ we have $a_0 = 2 \cdot c_1 + b_0$ by Claim 3.5. Therefore, the equation holds for $k = 0$.

Now let $k \geq 0$. Then we obtain

$$\begin{aligned}
\sum_{i=0}^{k+1} 2^i a_i &= 2^{k+1} a_{k+1} + \sum_{i=0}^{k} 2^i a_i \\
&= 2^{k+1} a_{k+1} + 2^{k+1} c_{k+1} + \sum_{i=0}^{k} 2^i b_i \qquad \text{(by induction)} \\
&= 2^{k+1} (a_{k+1} + c_{k+1}) + \sum_{i=0}^{k} 2^i b_i
\end{aligned}$$

$$= 2^{k+1}\,(b_{k+1} + 2c_{k+2}) + \sum_{i=0}^{k} 2^i b_i \qquad \text{(by Claim 3.5)}$$

$$= 2^{k+2} c_{k+2} + \sum_{i=0}^{k+1} 2^i b_i$$

This proves the claim. □

CLAIM 3.7. *Let $B = (b_0, \ldots, b_{m-1}, b_m)$ be as defined above. Then $B$ is compact.*

PROOF. We have to make sure that there is no $i \in [m]$ such that $b_i \neq 0$ and $b_{i+1} \neq 0$. In order to do so, we express $b_i$ and $b_{i+1}$ in terms of $a_i$, $a_{i+1}$ and $c_i$. Notice that $b_{i+1}$ is not fully determined by $a_i$, $a_{i+1}$ and $c_i$. Still these three values tell us whether $b_{i+1}$ is zero or not. This leads us to the following table, which shows that $B$ is, indeed, compact:

| $a_i$ | $a_{i+1}$ | $c_i$ | $c_{i+1}$ | $b_i$ | $b_{i+1}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | ±1 |
| 0 | 1 | 1 | 1 | –1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | ±1 |
| 1 | 1 | 0 | 1 | –1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

□

Now we are ready to finish the proof of Theorem 3.2. Let $A = (a_0, \ldots, a_{m-1})$ and $B = (b_0, \ldots, b_{m-1}, b_m)$ as above. By Claim 3.7, $B$ is compact. Moreover, we have

$$\mathrm{val}(B) = \sum_{i=0}^{m} 2^i b_i = 2^m c_m + \sum_{i=0}^{m-1} 2^i b_i \qquad \text{(since } c_m = b_m\text{)}$$

$$= \sum_{i=0}^{m-1} 2^i a_i = \mathrm{val}(A) \qquad \text{(by Claim 3.6)}$$

Therefore, $B$ is a compact signed-digit representation for $A$ as claimed in Theorem 3.2. By Remark 3.3, it can be computed in $\mathsf{AC}^0$. □

**Uniqueness of compact signed-digit representations.** The following lemmas are crucial tools both for proving uniqueness of compact representations and for the power circuit reduction process, which we describe later. In Myasnikov *et al.* (2012, Section 2.1) similar statements can be found.

LEMMA 3.8. *Let $A$ be a compact signed-digit representation and let $B = (b_0, \ldots, b_{n-1})$ be a compact signed-digit representation of digit length $n$ such that $b_i = n - i \bmod 2$ (i.e., $b_{n-1} = 1$ and then $B$ alternates between 0 and 1). Then we have*

*(i)* $\mathrm{val}(B) = \left\lfloor \frac{2^{n+1}}{3} \right\rfloor$,

*(ii)* $\mathrm{val}(A) \leq \mathrm{val}(B)$ *if and only if* $\mathrm{val}(A) \leq 0$ *or the digit length of $A$ is at most $n$.*

PROOF. First, we want to calculate $\mathrm{val}(B)$. If $n$ is even, then

$$\mathrm{val}(B) = \sum_{i=0}^{\frac{n}{2}-1} 2^{2i+1} = 2 \sum_{i=0}^{\frac{n}{2}-1} 4^i = 2 \cdot \frac{1 - 4^{\frac{n}{2}}}{1-4} = \frac{2}{3} \cdot (2^n - 1).$$

If $n$ is odd, then

$$\mathrm{val}(B) = \sum_{i=0}^{\frac{n-1}{2}} 2^{2i} = \sum_{i=0}^{\frac{n-1}{2}} 4^i = \frac{1 - 4^{\frac{n-1}{2}+1}}{1-4} = \frac{2^{n+1} - 1}{3}$$

showing that in any case $\mathrm{val}(B) = \left\lfloor \frac{2^{n+1}}{3} \right\rfloor$. In order to see (ii), we denote $A = (a_0, \ldots, a_{n-1})$. If $\mathrm{val}(A) \leq 0$, then clearly $\mathrm{val}(A) \leq \mathrm{val}(B)$. Hence, assume that the digit length of $A$ is at most $n$ and consider the following operations:

1. If $a_i = -1$, then set $a_i = 0$.
2. If $a_{n-1} = 0$, then set $a_{n-1} = 1$ and set $a_{n-2} = 0$.

3. If $a_i = a_{i+1} = 0$ with $i \in [1\,..\,n-2]$, then set $a_i = 1$ and $a_{i-1} = 0$ (technically, this rule subsumes the previous rule).

4. If $a_0 = a_1 = 0$, set $a_0 = 1$.

Let $A'$ be the number we obtained after applying at least one of the above operations to $A$ (if this is possible). Then $A'$ is also a compact signed-digit representation, the digit length of $A'$ is at most $n$, and $\operatorname{val}(A) < \operatorname{val}(A')$. Moreover, if $A \neq B$, then we always can apply one of these rules. This shows that $\operatorname{val}(A) \leq \operatorname{val}(B)$.

On the other hand, assume that the digit length of $A$ is $m$ with $m \geq n+1$. First, assume that $a_{m-1} = 1$ and set $A' = (a_0, \ldots, a_{m-3})$. Then, since $A$ is compact, we have $a_{m-2} = 0$ and, hence, $\operatorname{val}(A) = 2^{m-1} + \operatorname{val}(A')$. By the previous implication and part (i), we know that $|\operatorname{val}(A')| \leq \left\lfloor \frac{2^{m-1}}{3} \right\rfloor$. Therefore, $\operatorname{val}(A) \geq 2^{m-1} - |\operatorname{val}(A')| \geq 2^{m-1} - \left\lfloor \frac{2^{m-1}}{3} \right\rfloor > \left\lfloor \frac{2^m}{3} \right\rfloor \geq \left\lfloor \frac{2^{n+1}}{3} \right\rfloor = \operatorname{val}(B)$. If $a_{m-1} = -1$, we obtain $\operatorname{val}(A) < 0$ with the same argument. □

Lemma 3.9 (see Myasnikov *et al.* 2012, Lemma 4). *Let $A$ and $B$ be compact signed-digit representations with $A = (a_0, \ldots, a_{m-1})$ and $B = (b_0, \ldots, b_{m-1})$. Then:*

*(i)* $\operatorname{val}(A) = \operatorname{val}(B)$ *if and only if* $a_i = b_i$ *for all* $i \in [m]$.

*(ii) If there is some* $i$ *with* $a_i \neq b_i$ *and* $i_0 = \max\{i \in [m] \mid a_i \neq b_i\}$, *then* $\operatorname{val}(A) < \operatorname{val}(B)$ *if and only if* $a_{i_0} < b_{i_0}$.

Proof. Notice that (i) is an immediate consequence of (ii). In order to see (ii), observe that it suffices to show only one implication. Let $A' = (a_0, \ldots, a_{i_0})$ and $B' = (b_0, \ldots, b_{i_0})$ and assume that $0 = a_{i_0} < b_{i_0} = 1$ (the cases involving the value –1 follow with the same argument). Now, $A'$ and $B'$ are compact signed-digit representations, so by Lemma 3.8, $\operatorname{val}(A') \leq \left\lfloor \frac{2^{i_0+1}}{3} \right\rfloor$ and $\operatorname{val}(B') > \left\lfloor \frac{2^{i_0+1}}{3} \right\rfloor$. Hence, $\operatorname{val}(A) < \operatorname{val}(B)$. □

From this lemma together with Theorem 3.2, it follows that each $k \in \mathbb{Z}$ can be uniquely represented by a compact signed-digit representation $\mathrm{CR}(k)$. Likewise for a signed-digit representation $A$, we write $\mathrm{CR}(A)$ for its compact signed-digit representation.

COROLLARY 3.10. *The following problems are in* $\mathsf{AC}^0$*:*

*(i)* Input: *A signed-digit representation* $A$.
Output: $\mathrm{CR}(A)$.

*(ii)* Input: *Signed-digit representations* $A$ *and* $B$.
Output: *The compact signed-digit repr. of* $\mathrm{val}(A) + \mathrm{val}(B)$.

*(iii)* Input: *Signed-digit representations* $A$ *and* $B$.
Question: *Is* $\mathrm{val}(A) < \mathrm{val}(B)$*?*

PROOF. Given a signed-digit representation $A = (a_0, \ldots, a_{m-1})$, we can split it into two nonnegative binary numbers $B, C$ such that $\mathrm{val}(A) = \mathrm{val}(B) - \mathrm{val}(C)$ (i.e., $b_i = \max\{0, a_i\}$ and $c_i = -\min\{0, a_i\}$). From these binary numbers we can compute the difference in $\mathsf{AC}^0$ and then make the result compact using Theorem 3.2. To see (ii), we proceed exactly the same way. For comparing two signed-digit representations, we compute their compact representations using part (i) and then compare them in $\mathsf{AC}^0$ by evaluating the condition in Lemma 3.9. □

## 4. Operations on power circuits

**4.1. Basic operations.** Before we consider the computation of reduced power circuits, which is our main result in this section, let us introduce some more notation on power circuits and recall the basic operations used by Diekert, Laun & Ushakov (2013); Myasnikov, Ushakov & Won (2012) under circuit complexity aspects.

**Markings and chains.**

Definition 4.1. *Let $(\Gamma, \delta)$ be a reduced power circuit with $\Gamma$ given as the sorted list $\Gamma = (P_0, \ldots, P_{n-1})$.*

*(i) A* chain *$C$ of length $|C| = \ell$ in $\Gamma$ starting at $P_i = \operatorname{start}(C)$ is a sequence $(P_i, \ldots, P_{i+\ell-1})$ such that $\varepsilon(P_{i+j+1}) = 2 \cdot \varepsilon(P_{i+j})$ for all $j \in [\ell - 1]$.*

*In particular, $\varepsilon(P_{i+j}) = 2^j \cdot \varepsilon(P_i)$ for all $j \in [\ell]$. As we do for $\Gamma$, we treat a chain both as a sorted list and as a set.*

*(ii) We call a chain $C$* maximal *if it cannot be extended in either direction. We denote the set of all maximal chains by $\mathcal{C}_\Gamma$.*

*As a set, a reduced power circuit is the disjoint union of its maximal chains.*

*(iii) Let $M$ be a marking in the reduced power circuit $(\Gamma, \delta)$ and let $C = (P_i, \ldots, P_{i+\ell-1}) \in \mathcal{C}_\Gamma$ and define $a_j = M(P_{i+j})$ for $i \in [\ell]$. Then we write $\operatorname{digit}_C(M) = (a_0, \ldots, a_{\ell-1})$.*

*(iv) There is a unique maximal chain $C_0$ containing the node $P_0$ of value 1. We call $C_0$ the* initial maximal chain *of $\Gamma$ and denote it by $C_0 = C_0(\Gamma)$.*

For an example of a power circuit with three maximal chains, see Figure 4.1.

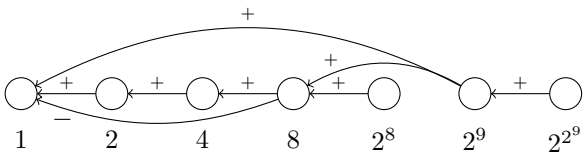



Figure 4.1: This power circuit is an example for a reduced power circuit with three maximal chains: The first one consists of the nodes of values $1, 2, 4, 8$, the next one is formed by the nodes of values $2^8$ and $2^9$, and the node of value $2^{2^9}$ is a maximal chain of length 1.

We will show how to computationally find the maximal chains in Corollary 4.7. The following facts are clear from the definition of maximal chains:

FACT 4.2. *Let $(\Gamma, \delta)$ be a reduced power circuit and let $M$ be a marking on $\Gamma$. Then the following holds:*

*(i)* $\varepsilon(M|_C) = \varepsilon(\mathrm{start}(C)) \cdot \mathrm{val}(\mathrm{digit}_C(M))$ *for every chain $C$ in $\Gamma$ (even if $C$ is not maximal).*

*(ii)* $\varepsilon(M) = \sum_{C \in \mathcal{C}_\Gamma} \varepsilon(\mathrm{start}(C)) \cdot \mathrm{val}(\mathrm{digit}_C(M))$.

*(iii) The marking $M$ is compact if and only if* $\mathrm{digit}_C(M)$ *is compact for all $C \in \mathcal{C}_\Gamma$.*

LEMMA 4.3. *Let $(\Gamma, \delta)$ be a reduced power circuit. Let $L$ and $M$ be compact markings in $\Gamma$ such that $\varepsilon(L) > \varepsilon(M)$ and let $0 \leq k \leq \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor$. Then $\varepsilon(L) \leq \varepsilon(M) + k$ if and only if the following holds:*

- $\varepsilon(M|_{\Gamma \setminus C_0}) = \varepsilon(L|_{\Gamma \setminus C_0})$ *and*
- $\varepsilon(L|_{C_0}) \leq \varepsilon(M|_{C_0}) + k$.

PROOF. We first assume that $\varepsilon(M|_{\Gamma \setminus C_0}) \neq \varepsilon(L|_{\Gamma \setminus C_0})$ and aim for showing that $\varepsilon(L) > \varepsilon(M) + k$: By Lemma 3.8, we have $|\varepsilon(L|_{C_0})|, |\varepsilon(M|_{C_0})| \leq \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor$. Hence,

$$\begin{aligned} \left|\varepsilon(M|_{C_0}) + k - \varepsilon(L|_{C_0})\right| &\leq |\varepsilon(M|_{C_0})| + k + |\varepsilon(L|_{C_0})| \\ &\leq 3 \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor \leq 2^{|C_0|+1} - 1. \end{aligned}$$

Furthermore, $\varepsilon(L|_{\Gamma \setminus C_0}) - \varepsilon(M|_{\Gamma \setminus C_0})$ is a multiple of $2^{|C_0|+1}$. Therefore, by the assumption $\varepsilon(L) > \varepsilon(M)$ and Lemma 3.9(ii), we obtain $\varepsilon(L|_{\Gamma \setminus C_0}) > \varepsilon(M|_{\Gamma \setminus C_0})$ and, thus, $\varepsilon(L|_{\Gamma \setminus C_0}) - \varepsilon(M|_{\Gamma \setminus C_0}) \geq 2^{|C_0|+1}$. It follows that

$$\varepsilon(L|_{\Gamma \setminus C_0}) - \varepsilon(M|_{\Gamma \setminus C_0}) + \varepsilon(L|_{C_0}) - \varepsilon(M|_{C_0}) - k \geq 1$$

and so $\varepsilon(L) > \varepsilon(M) + k$.

Now assume that $\varepsilon(M|_{\Gamma \setminus C_0}) = \varepsilon(L|_{\Gamma \setminus C_0})$. It remains to show that under this assumption we have $\varepsilon(L) \leq \varepsilon(M) + k$ if and only

if $\varepsilon(L|_{C_0}) \leq \varepsilon(M|_{C_0}) + k$. However, this follows immediately from the fact that

$$\varepsilon(L) = \varepsilon(L|_{\Gamma\setminus C_0}) + \varepsilon(L|_{C_0}) \quad \text{and} \quad \varepsilon(M) + k = \varepsilon(M|_{\Gamma\setminus C_0}) + \varepsilon(M|_{C_0}) + k.$$

This finishes the proof of the lemma. □

**Comparison of markings.**

LEMMA 4.4. *Given a reduced power circuit $(\Gamma, \delta)$ and a node $P \in \Gamma$, one can decide in $\mathsf{AC}^0$ whether $P \in C_0$.*

Be aware that in Lemma 4.4 we consider a promise problem: for $C_0$ to be defined and to decide in $\mathsf{AC}^0$ whether a given node $P$ is in $C_0$ we need the promise that the power circuit $(\Gamma, \delta)$ is reduced. In Remark 4.8 below we will see that we can actually test in $\mathsf{AC}^0$ if a given power circuit is reduced.

REMARK 4.5. *Since membership in $\mathsf{AC}^0$ often highly depends on the encoding of the input, in the following we always assume that power circuits are given in a suitable way.*

*In particular, we may assume that an $n$-node power circuit is given by the $n \times n$ matrix representing $\delta$ where each entry from $\{0, \pm 1\}$ is encoded using two bits. Moreover, in order to represent power circuits with fewer nodes within the same data structure, we can allow one* deleted *bit for every row and column of the matrix. Markings can be encoded the same way by a sequence of $n$ symbols from $\{0, \pm 1\}$. Moreover, if the power circuit is reduced, we also assume that the matrix representing $\delta$ is already in the sorted order (in particular, the ordering is not given by some separate data structure).*

*In the following, we do not further consider these encoding issues. Moreover, as soon as we are dealing with $\mathsf{TC}^0$ circuits, there is a lot of freedom how to encode inputs.*

PROOF (of Lemma 4.4). Let $\Gamma = (P_0, \ldots, P_{n-1})$. For each $i$ we define a signed-digit representation $A_i = (a_{i,0}, \ldots, a_{i,n-1})$ by $a_{i,j} =$

$\Lambda_{P_i}(P_j)$. These signed-digit representations might not be compact, but, if $P_i \in C_0$, then $A_i$ is compact (this is because, by Remark 2.10, $P_i$ has only successors in $C_0$). Using Corollary 3.10, we can compute the maximal $i_{max}$ such that $A_{i_{max}}$ is compact and for all $i < i_{max}$ also $A_i$ is compact and $\mathrm{val}(A_{i+1}) = \mathrm{val}(A_i) + 1 = i + 1$ (checking whether $A_i$ is compact, clearly, can be done in $\mathsf{AC}^0$).

By a straightforward induction, we obtain that for all $i \leq i_{max}$ we have $\mathrm{val}(A_i) = \varepsilon(\Lambda_{P_i})$ and $P_i \in C_0$. On the other hand, clearly, $P_{i_{max}+1} \notin C_0$. Hence, we have computed $C_0$. Thus, the lemma follows. □

PROPOSITION 4.6. *Let* $\triangle \in \{=, \neq, <, \leq, >, \geq\}$. *The following problems are in* $\mathsf{AC}^0$*:*

*(a)* Input: *A reduced power circuit* $(\Gamma, \delta)$ *and compact markings* $L$ *and* $M$ *on* $\Gamma$.
Question: *Is* $\varepsilon(L) \triangle \varepsilon(M)$*?*

*(b)* Input: *A reduced power circuit* $(\Gamma, \delta)$ *with compact markings* $L, M$ *and* $k \in [0 \,..\, \lfloor \frac{2^{|C_0|+1}}{3} \rfloor]$ *given in binary.*
Question: *Is* $\varepsilon(L) \triangle \varepsilon(M) + k$*?*

PROOF. Let us choose $\leq$ as $\triangle$ (the other cases follow from this case in a straightforward way).

Let $\Gamma = (P_0, \ldots, P_{n-1})$. By Lemma 3.9(i) we can check in $\mathsf{AC}^0$ if $\varepsilon(L) = \varepsilon(M)$. If this is not the case, then by Lemma 3.9(ii) we have $\varepsilon(M) < \varepsilon(L)$ if and only if $M(P_{i_0}) < L(P_{i_0})$ for $i_0 = \max\{i \in [n] \mid M(P_i) \neq L(P_i)\}$. Now, $i_0$ can be found in $\mathsf{AC}^0$ and, hence, the whole check is in $\mathsf{AC}^0$. This proves part (a).

For part (b) we first check whether $\varepsilon(L) \leq \varepsilon(M)$. If yes, then $\varepsilon(L) \leq \varepsilon(M) + k$. According to part (a), this check is possible in $\mathsf{AC}^0$. Now assume that $\varepsilon(L) > \varepsilon(M)$. By Lemma 4.4, we can compute $C_0$ in $\mathsf{AC}^0$. By Lemma 4.3 we know that $\varepsilon(L) \leq \varepsilon(M) + k$ if and only if $\varepsilon(M|_{\Gamma \setminus C_0}) = \varepsilon(L|_{\Gamma \setminus C_0})$ and $\varepsilon(L|_{C_0}) \leq \varepsilon(M|_{C_0}) + k$. The markings $M|_{\Gamma \setminus C_0}$ and $L|_{\Gamma \setminus C_0}$ are still compact markings in a reduced power circuit, and so we are able to decide in $\mathsf{AC}^0$ if that equality holds by part (a). So it remains to check if $\varepsilon(L|_{C_0}) \leq$

$\varepsilon(M|_{C_0}) + k$. This amounts to an addition and a comparison of signed-digit representations of digit length at most $|C_0| + 1$ (according to Lemma 3.8), which both can be done in $\mathsf{AC}^0$ (see Corollary 3.10). Thus, $\varepsilon(L) \triangle \varepsilon(M) + k$ can be checked in $\mathsf{AC}^0$. □

Corollary 4.7. *We can decide in* $\mathsf{AC}^0$*, given a reduced power circuit* $(\Gamma, \delta)$ *and nodes* $P, Q \in \Gamma$*, whether* $P$ *and* $Q$ *belong to the same maximal chain of* $\Gamma$*.*

Proof. Let $P = P_i$ and $Q = P_j$ with $i < j$. Then $P$ and $Q$ belong to the same maximal chain if and only if $\varepsilon(\Lambda_{P_{\ell+1}}) = \varepsilon(\Lambda_{P_\ell}) + 1$ for all $\ell \in [i \,..\, j-1]$. The latter can be checked in $\mathsf{AC}^0$ using Proposition 4.6. □

Remark 4.8. *Let us remark that we actually can decide in* $\mathsf{AC}^0$ *whether a given power circuit* $(\Gamma, \delta)$ *is reduced (thus, the promise in Lemma 4.4 can be checked in* $\mathsf{AC}^0$*). Indeed, assume that* $(\Gamma, \delta)$ *is not reduced and* $\Gamma = (P_0, \ldots, P_n)$*. Then there is a smallest index* $i$ *such that* $\Lambda_{P_i}$ *is not compact or* $\varepsilon(P_i) \leq \varepsilon(P_{i-1})$ *(see Definition 2.9). For a given power circuit* $(\Gamma, \delta)$*, we can check in* $\mathsf{AC}^0$ *if such a node* $P_i$ *exists: Assume that the nodes* $P_1, \ldots, P_{i-1}$ *have pairwise distinct evaluations, are sorted by their values, and all their successor markings are compact. First, check if* $\sigma(\Lambda_{P_i}) \subseteq \{P_1, \ldots, P_{i-1}\}$*. If not,* $(\Gamma, \delta)$ *is not reduced, see Remark 2.10. Otherwise, we can check in* $\mathsf{AC}^0$ *if* $\Lambda_{P_i}$ *is compact: using Proposition 4.6(b) we can check for two nodes* $P_{j-1}$ *and* $P_j$ *with* $j \leq i - 1$ *whether* $\varepsilon(P_j) = 2 \cdot \varepsilon(P_{j-1})$*. If* $\Lambda_{P_i}$ *is compact, we can check in* $\mathsf{AC}^0$ *if* $\varepsilon(P_i) \leq \varepsilon(P_{i-1})$ *using Proposition 4.6. We can apply the above procedure for each* $1 \leq i \leq n$ *independently in parallel. Then* $(\Gamma, \delta)$ *is not reduced if and only if we find such a first node* $P_i$ *as a witness that* $(\Gamma, \delta)$ *is not reduced.*

**Calculations with markings.**

Lemma 4.9. *The following problems are all in* $\mathsf{TC}^0$*:*

(a) Input: *A power circuit* $(\Pi, \delta_\Pi)$ *together with markings* $K$ *and* $L$.
Output: *A power circuit* $(\Pi', \delta_{\Pi'})$ *with a marking* $M$ *such that* $\varepsilon(M) = \varepsilon(K) + \varepsilon(L)$ *and* $(\Pi, \delta_\Pi) \leq (\Pi', \delta_{\Pi'})$, $|\Pi'| \leq 2 \cdot |\Pi|$ *and* $\operatorname{depth}(\Pi') = \operatorname{depth}(\Pi)$.

*If* $K$ *and* $L$ *have disjoint supports, then we can assume that* $(\Pi, \delta_\Pi) = (\Pi', \delta_{\Pi'})$.

(b) Input: *A power circuit* $(\Pi, \delta_\Pi)$ *together with a marking* $L$.
Output: *A marking* $M$ *in the power circuit* $(\Pi, \delta_\Pi)$ *such that* $\varepsilon(M) = -\varepsilon(L)$.

(c) Input: *A power circuit* $(\Pi, \delta_\Pi)$ *together with markings* $K$ *and* $L$ *such that* $\varepsilon(L) \geq 0$.
Output: *A power circuit* $(\Pi', \delta_{\Pi'})$ *with a marking* $M$ *such that* $\varepsilon(M) = \varepsilon(K) \cdot 2^{\varepsilon(L)}$ *and* $(\Pi, \delta_\Pi) \leq (\Pi', \delta_{\Pi'})$, $|\Pi'| \leq 3 \cdot |\Pi|$ *and* $\operatorname{depth}(\Pi') \leq \operatorname{depth}(\Pi) + 1$.

The proof of this lemma uses the following construction (see also Diekert *et al.* 2013):

Definition 4.10. *Let* $(\Pi, \delta)$ *be a power circuit and let* $M$ *be a marking on* $\Pi$.

(a) *Let* $P \in \Pi$. *We define a new power circuit* $\Pi \cup \{\textsc{Clone}(P)\}$ *where* $\textsc{Clone}(P)$ *is a new node with* $\Lambda_{\textsc{Clone}(P)} = \Lambda_P$.

(b) *We define a marking* $\textsc{Clone}(M)$ *as follows: First we clone all the nodes in* $\sigma(M)$. *Then we set* $\textsc{Clone}(M)(\textsc{Clone}(P)) = M(P)$ *for* $P \in \sigma(M)$ *and* $\textsc{Clone}(M)(P) = 0$ *otherwise.*

It is clear that the problem, given a power circuit $(\Pi, \delta)$ and a marking $M$, compute a new power circuit $(\Pi', \delta')$ containing $\textsc{Clone}(M)$ is in $\mathsf{TC}^0$—and even in $\mathsf{AC}^0$ when defining the underlying data structure properly. Notice that $|\Pi'| \leq 2 \cdot |\Pi|$ and $\operatorname{depth}(\Pi') = \operatorname{depth}(\Pi)$.

Proof. We apply the constructions described by Myasnikov *et al.* (2012, Section 7) and Diekert *et al.* (2013, Section 2).

Part (a): First, we clone the marking $K$ leading to a power circuit $(\Pi', \delta_{\Pi'})$ of size at most $2 \cdot |\Pi|$. Now $\textsc{Clone}(K)$ and $L$ certainly have disjoint supports. Then we define

$$M(P) = \begin{cases} \textsc{Clone}(K)(P), & P \in \sigma(\textsc{Clone}(K)), \\ L(P), & P \in \sigma(L), \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $\varepsilon(M) = \varepsilon(K) + \varepsilon(L)$, and $M$ can be output in $\mathsf{TC}^0$. If $\sigma(K) \cap \sigma(L) = \emptyset$, then no cloning is necessary. To show (b), we set

$$M(P) = \begin{cases} -L(P), & P \in \sigma(L), \\ 0, & \text{otherwise.} \end{cases}$$

As for (a), to define $M(P)$ we only have to look up $L(P)$ and change the sign and we do not have to create any new nodes or edges—so this can be done even in $\mathsf{AC}^0$.

To (c): To obtain $M$, we follow a similar approach as described in Diekert *et al.* (2013, Section 2). We first clone the markings $K$ and $L$ and so obtain markings $\textsc{Clone}(K)$ and $\textsc{Clone}(L)$. At this point, the size of $\Pi$ increased by a factor of at most three.

Next we create a new edge from each node $P \in \sigma(\textsc{Clone}(K))$ to each node $Q \in \sigma(\textsc{Clone}(L))$ with $\delta(P, Q) = \textsc{Clone}(L)(Q)$. This operation does not change the size of the power circuit, but it increases the depth by at most 1 since there are no incoming edges to nodes in $\sigma(\textsc{Clone}(K))$. Then the marking $\textsc{Clone}(K)$ is the marking we search for. □

Notice that the construction in (c) also yields $\varepsilon(M) = \varepsilon(K) \cdot 2^{\varepsilon(L)}$ in the case that $\varepsilon(L) < 0$. However, then the resulting graph might not be a power circuit anymore since it might have nodes of non-integral evaluation. Note that Diekert, Laun & Ushakov (2013) are not very precise here: it is actually not sufficient that $\varepsilon(K) \cdot 2^{\varepsilon(L)} \in \mathbb{Z}$ in order to assure that there are no nodes of non-integral evaluation.

**4.2. Relation to arithmetic circuits with + and $2^x$ gates.** Before we proceed to the power circuit reduction, our main result on power circuits, let us elaborate on the relation of power circuits to more general arithmetic circuits. A (constant) $(0, +, -, 2^x)$-circuit is a dag where each node is either a 0- (i.e., a constant-), +-, −- or a $2^x$-gate. 0-gates have zero inputs, +-gates two and −- and $2^x$-gates have one input. There is one designated output gate. The evaluation $\mathrm{eval}(\mathcal{C})$ of such a circuit $\mathcal{C}$ is defined in a straightforward way (as a real number—in general, it might not be an integer). The $2^x$-depth of a circuit $\mathcal{C}$ denoted by $\mathrm{depth}_{2^x}(\mathcal{C})$ is the maximal number of $2^x$-gates on any path in the circuit.

PROPOSITION 4.11. *For every power circuit $(\Pi, \delta)$ with a marking $M$, there is a $(0, +, -, 2^x)$-circuit $\mathcal{C}$ with $\mathrm{eval}(\mathcal{C}) = \varepsilon(M)$ such that*

- $|\mathcal{C}| \leq 2\,|\sigma(\delta)| + 3\,|\Pi| + 1$,
- $\mathrm{depth}(\mathcal{C}) \leq (\mathrm{depth}(\Pi) + 2) \cdot (\lceil \log(|\Pi|) \rceil + 2)$ *and*
- $\mathrm{depth}_{2^x}(\mathcal{C}) = \mathrm{depth}(\Pi) + 1$.

*Moreover, $\mathcal{C}$ can be computed in $\mathsf{TC}^0$.*

*Conversely, for every $(0, +, -, 2^x)$-circuit $\mathcal{C}$ there is a power circuit $(\Pi, \delta)$ (with possibly non-integral node values) with a marking $M$ such that $\mathrm{eval}(\mathcal{C}) = \varepsilon(M)$ and*

- $\mathrm{depth}(\Pi) \leq \mathrm{depth}_{2^x}(\mathcal{C})$ *and*
- $|\Pi| \leq |\mathcal{C}|^2 + |\mathcal{C}|$.

*Moreover, $(\Pi, \delta)$ and $M$ can be computed in $\mathsf{NC}^2$.*

*If the input of every $2^x$-gate of $\mathcal{C}$ is nonnegative, then $(\Pi, \delta)$ is, indeed, a power circuit—i.e., all nodes evaluate to positive integers.*

Note that technically speaking in the second part of this proposition $(\Pi, \delta)$ is not a power circuit as defined in Definition 2.6 since it might have nodes not evaluating to integers. Since we have not introduced a terminology for circuits without this integrality condition, we use the term "power circuit" here.

Proof. In order to transform $(\Pi, \delta)$ with a marking $M$ into a $(0, +, -, 2^x)$-circuit $\mathcal{C}$, we proceed as follows: we create one 0-gate; for every leaf (node of out-degree zero) of $\Pi$ we create a $2^x$-gate with input coming from the 0-gate, for every other node of $\Pi$, we create a $2^x$-gate whose input we describe next. For every marking (both $M$ and the successor markings $\Lambda_P$) we create a tree of $+$-gates (possibly with some $-$-gates) of logarithmic depth where the leaves correspond to some of the (already created) $2^x$- or 0-gates and the last $+$-gate (i.e., the root) evaluates to $\varepsilon(M)$ (resp. $\varepsilon(\Lambda_p)$). Now, the $2^x$-gate corresponding to a node $P \in \Pi$ receives its input from the $+$-gate corresponding to $\Lambda_P$. It is straightforward that this construction can be done in $\mathsf{TC}^0$.

Clearly, this process introduces at most one $+$-gate and one $-$-gate for every pair in the support of $\delta$ and every node in the support of $M$. So we have at most $2\,|\sigma(\delta)| + 2\,|\Pi|$ many $+$ and $-$-gates. Since there is one 0-gate and $|\Pi|$ many $2^x$-gates, the total number of gates is at most $2\,|\sigma(\delta)| + 3\,|\Pi| + 1$. It is clear that $\mathrm{depth}_{2^x}(\mathcal{C}) = \mathrm{depth}(\Pi) + 1$ (note that the depth increases by one because leaves of $\Pi$ are replaced by $2^x$-gates with input from the 0-gate). Moreover, the depth of each $+$-tree is bounded by $\lceil \log(|\Pi|) \rceil$; introducing the $-$-gates and connecting to the $2^x$-gates increases the depth further by 2 (note that for the $2^x$-gates with single input from a 0-gate this is a huge over-estimate). Since $\mathrm{depth}_{2^x}(\mathcal{C}) = \mathrm{depth}(\Pi) + 1$ and we have one additional $+$-tree for the marking $M$, the total depth is at most $(\mathrm{depth}(\Pi) + 2) \cdot (\lceil \log(|\Pi|) \rceil + 2)$.

Now consider a $(0, +, -, 2^x)$-circuit $\mathcal{C}$ with $n$ gates including its $2^x$-gates $h_1, \ldots, h_k$. As a first step, we replace each $2^x$-gate $h_i$ by an "input" gate $X_i$ and cut its incoming wire. Thus, we obtain an arithmetic circuit over $\mathbb{Z}$ with $+$- and $-$-gates. Each $+$- or $-$-gate $g$ computes a linear combination $\sum_{i=1}^{k} a_{g,i} X_i$ with $a_{g,i} \in \mathbb{Z}$. By Theorem 21 of Travers (2006), the $a_{g,i}$ can be computed in $\mathsf{GapL}$, and hence, in $\mathsf{NC}^2$ (Àlvarez & Jenner 1993, Theorem 4.1). Notice that $|a_{g,i}| < 2^n$ for all $g$ and $i$.

Now, to construct the power circuit $(\Pi, \delta)$, we proceed as follows: we start with $n$ singleton nodes. For each $2^x$-gate $h_j$ in $\mathcal{C}$ we construct nodes $Q_{j,0}, \ldots, Q_{j,n-1}$. The aim is to define $\Lambda_{Q_{j,\ell}}$ such that $\varepsilon(Q_{j,\ell}) = \mathrm{eval}(h_j) \cdot 2^\ell$; in particular, $\varepsilon(Q_{j,0}) = \mathrm{eval}(h_j)$ and

$C_j = (Q_{j,0}, \ldots, Q_{j,n-1})$ is a chain (by a slight abuse of the notation of Definition 4.1 since now the power circuit is not reduced).

Let $h_j$ be some $2^x$-gate and $g$ the gate from where $h_j$ receives its input. If $g$ is a 0-gate, we define $a_{j,i} = 0$ for all $i \in [1 \,..\, k]$; if $g$ is a $2^x$-gate $h_m$, we set $a_{j,m} = 1$ and $a_{j,i} = 0$ for all $i \neq m$. Otherwise, $g$ is a $+$ or $-$-gate. In this case we define $a_{j,i} = a_{g,i}$ where $a_{g,i} \in \mathbb{Z}$ is as above. Then for all $\ell \in [n]$ we define $\Lambda_{Q_{j,\ell}}$ on each chain $C_i$ such that $\mathrm{digit}_{C_i}(\Lambda_{Q_{j,\ell}})$ is the binary representation of $a_{j,i}$ (notice that $a_{j,i}$ requires only $n$ bits and all the chains $C_i$ for different $i$ are disjoint, so this is well-defined). Moreover, we add a $+$ edge from $Q_{j,\ell}$ to $\ell$ many of the singleton nodes. We do this for all $2^x$-gates in parallel. By induction we see that, indeed, $\varepsilon(Q_{j,\ell}) = \mathrm{eval}(h_j) \cdot 2^\ell$.

If the output gate of $\mathcal{C}$ is a $2^x$-gate $h_j$, we obtain a marking evaluating to the same value by simply marking $Q_{j,0}$ with one; if the output gate is a $+$- or $-$-gate, we obtain a corresponding marking in the same fashion as for the $\Lambda_{Q_{j,0}}$ described above.

Clearly, the whole computation also can be done in $\mathsf{NC}^2$. The bound $|\Pi| \leq n^2 + n$ is straightforward: we introduced at most $n$ singleton nodes and then for every of the at most $n$ $2^x$-gates we introduced $n$ additional nodes. The bound on the depth is because we can have an edge from $Q_{j,\ell}$ to $Q_{j',\ell'}$ only if there is a path from $h_j$ to $h_{j'}$ in $\mathcal{C}$. Adding the edges from $Q_{j,\ell}$ to the singleton nodes only increases its depth if the depth without these edges was zero, i.e., if $h_j$ is a $2^x$-gate whose input is a sum of 0-gates. However, we counted the depth of such $2^x$-gates already as one—so also in this case the depth does not increase. □

**4.3. Power circuit reduction.** While compact markings on a reduced power circuit yield unique representations of integers, in an arbitrary power circuit $(\Pi, \delta_\Pi)$ we can have two markings $L$ and $M$ such that $L \neq M$ but $\varepsilon(L) = \varepsilon(M)$. Therefore, given an arbitrary power circuit, we wish to produce a reduced power circuit for comparing markings. This is done by the following theorem, which is our main technical result on power circuits.

THEOREM 4.12. *The following is in* $\mathsf{DepParaTC}^0$ *parametrized by* $\mathrm{depth}(\Pi)$*:*

Input: *A power circuit* $(\Pi, \delta_\Pi)$ *together with a marking* $M$ *on* $\Pi$.
Output: *A reduced power circuit* $(\Gamma, \delta)$ *together with a compact marking* $\widetilde{M}$ *on* $\Gamma$ *such that* $\varepsilon(\widetilde{M}) = \varepsilon(M)$.

For a power circuit $(\Pi, \delta_\Pi)$ with a marking $M$, we call the power circuit $(\Gamma, \delta)$ together with the marking $\widetilde{M}$ obtained by Theorem 4.12 the *reduced form* of $\Pi$.

The proof of Theorem 4.12 consists of several steps, which we introduce on the next pages. The high-level idea is as follows: Like Diekert *et al.* (2013); Myasnikov *et al.* (2012), we keep the invariant that there is an already reduced part and a non-reduced part (initially the non-reduced part is $\Pi$). The main difference is that in one iteration we insert *all* the nodes of the non-reduced part that have only successors in the reduced part into the reduced part. Each iteration can be done in $\mathsf{TC}^0$; after $\operatorname{depth}(\Pi) + 1$ iterations we obtain a reduced power circuit.

**Insertion of new nodes.** The following procedure, which we will call InsertNodes, is a basic tool for the reduction process. Let $(\Gamma, \delta)$ be a reduced power circuit and $I$ be a set of nodes with $\Gamma \cap I = \emptyset$. Assume that for every $P \in I$ there exists a marking $\Lambda_P \colon \Gamma \to \{-1, 0, 1\}$ satisfying:

- $\varepsilon(\Lambda_P) \geq 0$ for all $P \in I$,
- $\Lambda_P$ is compact for all $P \in I$, and
- $\varepsilon(\Lambda_P) \neq \varepsilon(\Lambda_Q)$ for all $P, Q \in I \cup \Gamma$, $P \neq Q$.

We wish to add $I$ to the reduced power circuit $(\Gamma, \delta)$. For this, we set $\Gamma' = \Gamma \cup I$ and define $\delta' \colon \Gamma' \times \Gamma' \to \{-1, 0, 1\}$ in the obvious way: $\delta'|_{\Gamma\times\Gamma} = \delta$, $\delta'|_{\Gamma'\times I} = 0$ and $\delta'(P, Q) = \Lambda_P(Q)$ for $(P, Q) \in I \times \Gamma$. Now, $(\Gamma', \delta')$ is a power circuit with $(\Gamma, \delta) \leq (\Gamma', \delta')$ and for every $P \in I$ the map $\Lambda_P$ is the successor marking of $P$. Moreover, each node of $\Gamma'$ has a unique value. In order to obtain a reduced power circuit, we need to sort the nodes in $\Gamma'$ according to their values: Since for every node $P \in \Gamma'$ the marking $\Lambda_P$ is a compact marking on the reduced power circuit $\Gamma$, by Proposition 4.6, for $P, Q \in \Gamma'$ we are able to decide in $\mathsf{AC}^0$ whether $\varepsilon(\Lambda_Q) \leq \varepsilon(\Lambda_P)$.

Therefore, by Example 2.2 we can sort $\Gamma'$ according to the values of the nodes in $\mathsf{TC}^0$ and, hence, assume that $\Gamma' = (P_0, \ldots, P_{|\Gamma'|-1})$ is in increasing order.

Observe that $|\Gamma'| = |\Gamma \cup I| = |\Gamma| + |I|$. In addition, inserting a new node either extends an already existing maximal chain, joins two existing maximal chains, or increases the number of maximal chains by one. Therefore, $|\mathcal{C}_{\Gamma'}| \leq |\mathcal{C}_\Gamma| + |I|$. So we have proven the following:

LEMMA 4.13 (INSERTNODES). *The following problem is in* $\mathsf{TC}^0$*:*

Input: *A power circuit* $(\Gamma, \delta)$ *and a set* $I$ *with the properties described above.*

Output: *A reduced power circuit* $(\Gamma', \delta')$ *such that* $(\Gamma, \delta) \leq (\Gamma', \delta')$ *and such that for every* $P \in I$ *there is a node* $Q$ *in* $\Gamma'$ *with* $\Lambda_Q = \Lambda_P$*. In addition,*

- $|\Gamma'| = |\Gamma| + |I|$*, and*
- $|\mathcal{C}_{\Gamma'}| \leq |\mathcal{C}_\Gamma| + |I|$.

**The three steps of the reduction process.** The reduction process for a power circuit $(\Pi, \delta_\Pi)$ with a marking $M$ consists of several iterations. Each iteration starts with a power circuit $(\Gamma_i \cup \Xi_i, \delta_i)$ such that $\Gamma_i$ is a reduced sub-power circuit and a marking $M_i$ with $\varepsilon(M_i) = \varepsilon(M)$. The aim of one iteration is to integrate the vertices $\mathrm{Min}(\Xi_i) \subseteq \Xi_i$ into $\Gamma_i$ where $\mathrm{Min}(\Xi_i)$ is defined by

$$\mathrm{Min}(\Xi_i) = \{P \in \Xi_i \mid \sigma(\Lambda_P) \subseteq \Gamma_i\}$$

and to update the marking $M_i$ accordingly. Each iteration consists of the three steps called UPDATENODES, EXTENDCHAINS, and UPDATEMARKINGS, which can be done in $\mathsf{TC}^0$. We have $\Xi_{i+1} = \Xi_i \backslash \mathrm{Min}(\Xi_i)$. Thus, the full reduction process consists of $\mathrm{depth}(\Pi)+1$ many $\mathsf{TC}^0$ computations. Let us now describe these three steps in detail and also show that they can be done in $\mathsf{TC}^0$. After that we present the full algorithm for power circuit reduction.

We write $(\Gamma \cup \Xi, \delta) = (\Gamma_i \cup \Xi_i, \delta_i)$ for the power circuit at the start of one iteration(for simplicity we do not write the indices). Let us fix its precise properties: $\Gamma \cap \Xi = \emptyset$, $(\Gamma, \delta|_{\Gamma\times\Gamma}) \leq (\Gamma \cup \Xi, \delta)$

is a reduced power circuit and $\Lambda_P|_\Gamma$ is a compact marking for every $P \in \Xi$. Moreover, we assume that $|C_0(\Gamma)| \geq \lceil \log(|\Xi|) \rceil + 1$.

LEMMA 4.14 (UPDATENODES). *The following problem is in* $\mathsf{TC}^0$*:*

Input: *A power circuit* $(\Gamma \cup \Xi, \delta)$ *as above.*
Output: *A reduced power circuit* $(\Gamma', \delta')$ *such that*

- $(\Gamma, \delta|_{\Gamma\times\Gamma}) \leq (\Gamma', \delta')$,
- *for every node* $Q \in \mathrm{Min}(\Xi)$ *there exists a node* $P \in \Gamma'$ *with* $\varepsilon(P) = \varepsilon(Q)$,
- $|\Gamma'| \leq |\Gamma| + |\mathrm{Min}(\Xi)|$, *and*
- $|\mathcal{C}_{\Gamma'}| \leq |\mathcal{C}_\Gamma| + |\mathrm{Min}(\Xi)|$.

For the proof, we define the following equivalence relation $\sim_\varepsilon$ on $\Gamma \cup \mathrm{Min}(\Xi)$:

$$P \sim_\varepsilon Q \text{ if and only if } \varepsilon(P) = \varepsilon(Q).$$

For $P \in \Gamma \cup \mathrm{Min}(\Xi)$ we write $[P]_\varepsilon$ for the equivalence class containing $P$.

PROOF. Consider the equivalence relation $\sim_\varepsilon$ as defined above on $\Gamma \cup \mathrm{Min}(\Xi)$. Define a set $I \subseteq \mathrm{Min}(\Xi)$ by taking one representative of each $\sim_\varepsilon$-class not containing a node of $\Gamma$. Such a set $I$ can be computed in $\mathsf{TC}^0$: Clearly, $\mathrm{Min}(\Xi)$ can be computed in $\mathsf{TC}^0$. The $\sim_\varepsilon$-classes can be computed in $\mathsf{AC}^0$ by Proposition 4.6. Finally, for defining $I$ one has to pick representatives. For example, for every $\sim_\varepsilon$-class which does not contain a node of $\Gamma$ one can pick the first node in the input which belongs to this class. These representatives also can be found in $\mathsf{TC}^0$. Now, we can apply Lemma 4.13 to insert $I$ into $\Gamma$ in $\mathsf{TC}^0$. This yields our power circuit $(\Gamma', \delta')$. The size bounds follow now immediately from those in Lemma 4.13 (notice that $|I| \leq |\mathrm{Min}(\Xi)|$). □

LEMMA 4.15 (EXTENDCHAINS). *The following is in* $\mathsf{TC}^0$*:*

Input: *A reduced power circuit* $(\Gamma', \delta')$ *and* $\mu \in \mathbb{N}$ *such that* $\mu \leq \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor$ *(where, as before,* $C_0 = C_0(\Gamma')$ *is the initial maximal chain of* $\Gamma'$*)*

Output: *A reduced power circuit* $(\Gamma'', \delta'')$ *such that*

- $(\Gamma', \delta') \leq (\Gamma'', \delta'')$,
- *for each* $P \in \Gamma'$ *and each* $i \in [0\,..\,\mu]$ *there is a node* $Q \in \Gamma''$ *with* $\varepsilon(\Lambda_Q) = \varepsilon(\Lambda_P) + i$,
- $|\Gamma''| \leq |\Gamma'| + |\mathcal{C}_{\Gamma'}| \cdot \mu$, *and*
- $|\mathcal{C}_{\Gamma''}| \leq |\mathcal{C}_{\Gamma'}|$.

PROOF. First, consider the case that $|C_0| = 1$ and assume for contradiction that $|\Gamma'| \geq 2$. Let $P \in \Gamma' \setminus C_0$ be minimal (w.r.t $\varepsilon$). Then, $\varepsilon(P) = 2$ because its only successor is the single vertex in $C_0$. Thus, $|C_0| > 1$. So if $|C_0| = 1$, then $|\Gamma'| = 1$ and $\mu \leq 1$. If $\mu = 1$, then just one node has to be created, namely the one of value 2 and we are done. Thus, in the following we can assume that $|C_0| \geq 2$. Now, the proof of Lemma 4.15 consists of two steps: first, we extend only the chain $C_0$ to some longer (and long enough) chain in order to make sure that the values of the (compact) successor markings of the nodes we wish to introduce can be represented within the power circuit; only afterward, we add the new nodes as described in the lemma.

**Step 1:** We first want to extend the chain $C_0$ to the chain $\tilde{C}_0$ of minimal length such that $\tilde{C}_0$ is a maximal chain, $C_0 \subseteq \tilde{C}_0$, and the last node of $\tilde{C}_0$ is not already present in $\Gamma'$. The resulting power circuit will be denoted by $\tilde{\Gamma}$. We define

$$i_0 = \min\left\{ i \in [|\Gamma'|] \;\middle|\; \varepsilon(\Lambda_{P_{i+1}}) - \varepsilon(\Lambda_{P_i}) > 2 \right\}.$$

Here, we use the convention that $P_{|\Gamma'|}$ has value infinity, so $i_0$ indeed exists. Furthermore, we define

$$I = \left\{ i \in [0\,..\,i_0] \;\middle|\; \varepsilon(\Lambda_{P_{i+1}}) - \varepsilon(\Lambda_{P_i}) \geq 2 \right\}.$$

Thus, in order to obtain $\tilde{\Gamma}$, we need to insert a new node between $P_i$ and $P_{i+1}$ into $\Gamma'$ for each $i \in I$ (resp. one node above $P_{i_0}$). Since the successor markings of these new nodes might point to some of the other new nodes, we cannot apply Lemma 4.13 as a black-box. Instead, we need to take some more care: the rough idea is that, first, we compute all positions $I$ where new nodes need to be introduced ($I$ is as defined above), then we compute compact signed-digit representations for the respective successor markings, and, finally, we introduce these new nodes all at once knowing that all nodes where the successor markings point to are also introduced at the same time. In order to map the positions of nodes in $\Gamma'$ to positions of nodes in $\tilde{\Gamma}$, we introduce a function $\lambda\colon [|\Gamma'|] \to \mathbb{N}$ with

$$\lambda(i) = i + |I \cap [0\,..\,i-1]|\,.$$

Observe that $\lambda(i) = i$ for $i \in [|C_0|]$, and $\lambda(i+1) = \lambda(i) + 2$ for $i \in I$, and $\lambda(j) = j + |I|$ for $j \geq i_0 + 1$.

For each $i \in I$ we introduce a node $Q_i$ whose successor marking we will specify later such that $\varepsilon(Q_i) = 2\varepsilon(P_i)$. We define the new power circuit $\tilde{\Gamma} = (\tilde{P}_0, \ldots, \tilde{P}_{|\Gamma'|+|I|-1})$ by

$$\tilde{P}_j = \begin{cases} P_i & \text{if } j = \lambda(i) \\ Q_i & \text{if } j = \lambda(i)+1 \text{ and } i \in I. \end{cases}$$

Notice that, if $j = \lambda(i) + 1$ for some $i \in I$, then $j \neq \lambda(i)$ for any $i$—hence, $\tilde{P}_j$ is well-defined in any case.

The nodes $\tilde{P}_0, \ldots, \tilde{P}_{\lambda(i_0)+1}$ will form the chain $\tilde{C}_0$ as claimed above. Moreover, we have $\Gamma' \subseteq \tilde{\Gamma}$ and $\tilde{\Gamma}$ is sorted increasingly. The successor markings of nodes from $\Gamma'$ remain unchanged (i.e., $\Lambda_{\tilde{P}_{\lambda(i)}}(\tilde{P}_{\lambda(j)}) = \Lambda_{P_i}(P_j)$ for $i, j \in [|\Gamma'|]$ and $\Lambda_{\tilde{P}_{\lambda(i)}}(Q_j) = 0$ for $j \in I$).

For every $i \in I$ we define the successor marking of the node $Q_i$ by

$$\operatorname{digit}_{\tilde{C}_0}(\Lambda_{Q_i}) = \operatorname{CR}(\varepsilon(\Lambda_{P_i}) + 1) \quad \text{and} \quad \Lambda_{Q_i}|_{\tilde{\Gamma}\setminus\tilde{C}_0} = 0.$$

Be aware that, since $Q_i \in \tilde{C}_0$, also the successor marking of $Q_i$ (of value $\varepsilon(\Lambda_{P_i}) + 1$) can be represented using only the nodes from $\tilde{C}_0$ (see Remark 2.10), so this is, indeed, a meaningful definition

(be aware that to represent $\varepsilon(\Lambda_{P_i})+1$, we might need some of the additional nodes $Q_i$, but never a node that is not part of the chain $\tilde{C}_0$). Clearly, this yields $\varepsilon(\Lambda_{Q_i}) = \varepsilon(\Lambda_{P_i})+1$ as desired.

We obtain a reduced power circuit $(\tilde{\Gamma}, \tilde{\delta})$ with $(\Gamma', \delta') \leq (\tilde{\Gamma}, \tilde{\delta})$ where the map $\tilde{\delta}\colon \tilde{\Gamma} \to \{-1, 0, 1\}$ is defined by the successor markings. Moreover, $\tilde{C}_0 \subseteq \tilde{\Gamma}$ has the required properties.

It remains to show that $\tilde{\Gamma}$ can be computed in $\mathsf{TC}^0$: As $|C_0| \geq 2$, according to Proposition 4.6, we are able to decide in $\mathsf{AC}^0$ whether the markings $\Lambda_{P_i}$ and $\Lambda_{P_{i+1}}$ differ by 1, 2, or more than 2—for all $i \in [|\Gamma'|]$ in parallel. Now, $i_0$ can be determined in $\mathsf{TC}^0$ via its definition as above. Likewise $I$ and the function $\lambda$ can be computed in $\mathsf{TC}^0$. By Corollary 3.10, $\mathrm{CR}\,(\varepsilon(\Lambda_{P_i})+1)$ for $i \in I$ can be computed in $\mathsf{AC}^0$ (since $|\tilde{C}_0| \leq 2 \cdot |\Gamma'|$) showing that altogether $\tilde{\Gamma}$ can be computed in $\mathsf{TC}^0$.

**Step 2:** The second step is to add nodes above each chain of $\tilde{\Gamma}$ as required in the Lemma. The outcome will be denoted by $(\Gamma'', \delta'')$. We start by defining

$$\begin{aligned} d_i &= \min\{\varepsilon(\Lambda_{\tilde{P}_{i+1}}) - \varepsilon(\Lambda_{\tilde{P}_i}) - 1, \mu\} && \text{for } i \in [|\tilde{\Gamma}|] \setminus \{|\tilde{C}_0| - 1\}, \\ d_i &= \min\{\varepsilon(\Lambda_{\tilde{P}_{i+1}}) - \varepsilon(\Lambda_{\tilde{P}_i}) - 1, \mu - 1\} && \text{for } i = |\tilde{C}_0| - 1. \end{aligned}$$

Note that because the last node of $\tilde{C}_0$ is not in $\Gamma'$, $\mu - 1$ new nodes suffice in the latter case. In order to obtain $(\Gamma'', \delta'')$ from $(\tilde{\Gamma}, \tilde{\delta})$, for every $i \in [|\tilde{\Gamma}|]$ and every $h \in [1\,..\,d_i]$ we have to insert a node $R^{(i,h)}$ such that

$$\varepsilon(\Lambda_{R^{(i,h)}}) = \varepsilon(\Lambda_{\tilde{P}_i}) + h.$$

Observe that the numbers $d_i$ can be computed in $\mathsf{TC}^0$: since

$$\mu + 1 \leq \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor + 1 \leq \left\lfloor \frac{2^{|\tilde{C}_0|}}{3} \right\rfloor + 1 \leq \left\lfloor \frac{2^{|\tilde{C}_0|+1}}{3} \right\rfloor,$$

by Proposition 4.6, we can check in $\mathsf{AC}^0$ whether $\varepsilon(\Lambda_{\tilde{P}_{i+1}}) \leq \varepsilon(\Lambda_{\tilde{P}_i}) + k$ with $k \leq \mu + 1$. If $i = |\tilde{C}_0| - 1$, we choose $k = \mu$, otherwise $k = \mu + 1$. If the respective inequality holds, we obtain by Lemma 4.3 that $\varepsilon(\Lambda_{\tilde{P}_{i+1}}) - \varepsilon(\Lambda_{\tilde{P}_i}) - 1 = \varepsilon(\Lambda_{\tilde{P}_{i+1}}|_{\tilde{C}_0}) - \varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) - 1$.

For the latter we have signed-digit representations of digit length at most $|\tilde{C}_0|$. Hence, this difference can be computed in $\mathsf{TC}^0$.

Since $\tilde{P}_{|\tilde{C}_0|-1} \notin \Gamma'$ and in Step 1 we have not introduced any vertex above $\tilde{P}_{|\tilde{C}_0|-1}$, we know that $\tilde{P}_{|\tilde{C}_0|-1}$ is not marked by $\Lambda_{\tilde{P}}$ for any $\tilde{P} \in \tilde{\Gamma}$. Therefore, for all $i \in [|\tilde{\Gamma}|]$ we have $\varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) + \mu \leq \left\lfloor \frac{2^{|\tilde{C}_0|}}{3} \right\rfloor + \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor \leq 2 \left\lfloor \frac{2^{|\tilde{C}_0|}}{3} \right\rfloor$ and, hence, by Lemma 3.8, $\varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) + h$ can be represented as a compact marking using only nodes from $\tilde{C}_0$ for every $h \in [1 \,..\, d_i]$. Thus, for every $d_i \neq 0$ and every $h \in [1 \,..\, d_i]$ we define a successor marking of $R^{(i,h)}$ by

$$\operatorname{digit}_{\tilde{C}_0}(\Lambda_{R^{(i,h)}}) = \operatorname{CR}(\varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) + h) \quad \text{and} \quad \Lambda_{R^{(i,h)}}|_{\tilde{\Gamma}\setminus\tilde{C}_0} = \Lambda_{\tilde{P}_i}|_{\tilde{\Gamma}\setminus\tilde{C}_0}.$$

Again, we know that $|\tilde{C}_0| \leq 2\,|\Gamma'|$. So, according to Corollary 3.10 we are able to calculate $\operatorname{CR}(\varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) + h)$ in $\mathsf{AC}^0$.

Now we set $I = \left\{ R^{(i,h)} \;\middle|\; d_i \neq 0, h \in [1 \,..\, d_i] \right\}$. According to Lemma 4.13 we are able to construct in $\mathsf{TC}^0$ a reduced power circuit $(\Gamma'', \delta'')$ such that $(\tilde{\Gamma}, \tilde{\delta}) \leq (\Gamma'', \delta'')$ and such that for each $R \in I$ there exists a node $Q \in \Gamma''$ with $\varepsilon(Q) = \varepsilon(R)$.

Considering the size of $\Gamma''$, observe that during the whole construction, for every node $P_i \in \Gamma'$ we create at most $\mu$ new nodes between $P_i$ and $P_{i+1}$.

Moreover, we only create new nodes between $P_i$ and $P_{i+1}$ if $P_i$ is the last node of a maximal chain of $\Gamma'$. Furthermore, notice that the only node of $\Gamma'$ above which we have introduced new nodes in both Step 1 and Step 2 is the second largest node of $\tilde{C}_0$: in Step 1 we have created one new node and in Step 2 we have created at most $\mu - 1$ new nodes above it. Thus, for every chain of $\Gamma'$ we have introduced at most $\mu$ new nodes. Thus, $|\Gamma''| \leq |\Gamma'| + |\mathcal{C}_{\Gamma'}| \cdot \mu$. Finally, the new nodes we create only prolongate the already existing chains, so we do not create any new chains. This finishes the proof of the lemma. □

In the following, $(\Gamma', \delta')$ denotes the power circuit obtained by UpdateNodes when starting with $(\Gamma \cup \Xi, \delta)$, and $(\Gamma'', \delta'')$ denotes the power circuit obtained by ExtendChains with $\mu = \lceil \log(|\operatorname{Min}(\Xi)|) \rceil + 1$ on input of the power circuit $(\Gamma', \delta')$ (observe that, by the assumption $|C_0(\Gamma)| \geq \lceil \log(|\Xi|) \rceil + 1$, the condition on

$\mu$ in Lemma 4.15 is satisfied). The value of $\mu$ is chosen to make sure that in the following lemma one can make the markings compact. Indeed, if $\mathrm{Min}(\Xi) = \{P_1, \ldots, P_k\}$ and all $P_i$ have the same evaluation and are marked with 1 by $M$, then we might need a node of value $2^\mu \cdot \varepsilon(P_1)$ in order to make $M$ compact.

LEMMA 4.16 (UPDATEMARKINGS). *The following problem is in* $\mathsf{TC}^0$*:*

| | |
|---|---|
| Input: | *The power circuit* $(\Gamma'', \delta'')$ *as a result of* EXTENDCHAINS *with* $\mu = \lceil \log(\lvert\mathrm{Min}(\Xi)\rvert) \rceil + 1$ *and a marking* $M$ *on* $\Gamma \cup \Xi$*.* |
| Output: | *A marking* $\widetilde{M}$ *on* $\Gamma'' \cup (\Xi \setminus \mathrm{Min}(\Xi))$ *such that* $\varepsilon(\widetilde{M}) = \varepsilon(M)$ *and* $\widetilde{M}\lvert_{\Gamma''}$ *is compact.* |

PROOF. Consider again the equivalence relation $\sim_\varepsilon$ as defined above on $\Gamma'' \cup \mathrm{Min}(\Xi)$. For the equivalence class of a node $P \in \Gamma'' \cup \mathrm{Min}(\Xi)$, we write $[P]_\varepsilon$. We will define the marking $\widetilde{M}$ on $\Gamma''$ by defining it on each maximal chain. Recall that we can view $M$ as a marking on $\Gamma'' \cup \Xi$ by defining $M(P) = 0$ if $P \notin \Gamma \cup \Xi$.

Let $C = (P_i, \ldots, P_{i+h-1}) \in \mathcal{C}_{\Gamma''}$ be a maximal chain of length $h$ and let

$$S = \bigcup_{P \in C} [P]_\varepsilon = \bigcup_{P \in C} \{Q \in \Gamma'' \cup \mathrm{Min}(\Xi) \mid \varepsilon(Q) = \varepsilon(P)\}.$$

Note that $S \subseteq \Gamma'' \cup \mathrm{Min}(\Xi)$. We wish to find a compact marking $\tilde{M}_C$ with support contained in $C \subseteq \Gamma''$ and evaluation $\varepsilon(\tilde{M}_C) = \varepsilon(M|_S)$. First, define the integer

$$Z_{M,C} = \sum_{r=0}^{h-1} \left( \sum_{Q \in [P_{i+r}]_\varepsilon} M(Q) \right) 2^r.$$

Then we have

$$\begin{aligned} Z_{M,C} \cdot \varepsilon(\mathrm{start}(C)) &= \sum_{r=0}^{h-1} \sum_{Q \in [P_{i+r}]_\varepsilon} M(Q) 2^r \cdot \varepsilon(\mathrm{start}(C)) \\ &= \sum_{Q \in S} M(Q) \varepsilon(Q) \\ &= \varepsilon(M|_S). \end{aligned}$$

Thus, defining $\tilde{M}_C$ by $\mathrm{digit}_C(\tilde{M}_C) = \mathrm{CR}(Z_{M,C})$ gives our desired marking.

However, be aware that, for this, we have to show that the digit length of $\mathrm{CR}(Z_{M,C})$ is at most $|C| = h$. Let $k$ be maximal such that $P_{i+k} \in \Gamma'$. Then, in particular, no node in $S$ with higher evaluation than $P_{i+k}$ is marked by $M$. Moreover, by the properties of ExtendChains($\lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 1$), we have $h - 1 - k \geq \lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 1$. Therefore,

$$\begin{aligned} Z_{M,C} &\leq \mathrm{val}(\mathrm{digit}_C(M)) + |\mathrm{Min}(\Xi)| \cdot 2^k \\ &\leq \frac{1}{3} \cdot 2^{k+2} + 2^{k+\log(|\mathrm{Min}(\Xi)|)} && \text{(by Lemma 3.8)} \\ &\leq \frac{4}{3} \cdot \left(2^k + 2^{k+\log(|\mathrm{Min}(\Xi)|)}\right) \\ &\leq \frac{2}{3} \cdot 2^{k+\lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 2}. \end{aligned}$$

Thus, by Lemma 3.8, the digit length of $\mathrm{CR}(Z_{M,C})$ is at most $k + \lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 2 \leq h$.

By Corollary 4.7, the maximal chains can be determined in $\mathsf{TC}^0$. Now, for every maximal chain $C$ the (binary) number $Z_{M,C}$ can be computed in $\mathsf{TC}^0$ using Proposition 4.6 (to obtain the equivalence classes) and iterated addition. Moreover, the numbers $Z_{M,C}$ can be made compact in $\mathsf{AC}^0$ using Theorem 3.2. Thus, the marking $\tilde{M}_C$ can be computed in $\mathsf{TC}^0$. The marking $\widetilde{M}$ as desired in the lemma is simply defined by $\widetilde{M}|_{\Xi \setminus \mathrm{Min}(\Xi)} = M|_{\Xi \setminus \mathrm{Min}(\Xi)}$ and $\widetilde{M}|_C = \tilde{M}_C|_C$ for $C \in \mathcal{C}_{\Gamma''}$—all the markings $\tilde{M}_C$ can be computed in parallel. □

Proof (of Theorem 4.12). Now we are ready to describe the full reduction process based on the three steps described above. We

aim for a $\mathsf{DepParaTC}^0$ circuit where the input is parametrized by the depth of the power circuit. The input is some arbitrary power circuit $(\Pi, \delta_\Pi)$ together with a marking $M$ on $\Pi$. We start with some initial reduced power circuit $(\Gamma_0, \delta_0)$ and some non-reduced part $\Xi_0 = \Pi$ and successively apply the three steps to obtain power circuits $(\Gamma_i \cup \Xi_i, \delta_i)$ and markings $M_i$ for $i = 0, 1 \ldots$ while keeping the following invariants:

- $(\Gamma_i, \delta_i|_{\Gamma_i \times \Gamma_i}) \leq (\Gamma_i \cup \Xi_i, \delta_i)$ (i.e., there are no edges from $\Gamma_i$ to $\Xi_i$),
- $\Gamma_i$ is reduced,
- $\Gamma_{i-1} \leq \Gamma_i$ and $\Xi_i \subseteq \Xi_{i-1}$,
- $\varepsilon(M_i) = \varepsilon(M)$,
- $M_i|_{\Gamma_i}$ is compact.

Also, as long as $\Xi_{i-1} \neq \emptyset$ we assure that $\mathrm{depth}(\Xi_i) < \mathrm{depth}(\Xi_{i-1})$.

We first construct the initial reduced power circuit $(\Gamma_0, \tilde{\delta}_0)$ which consists exactly of a chain of length $\ell = \lceil \log(|\Pi|) \rceil + 1$. This can be done as follows: Let $\Gamma_0 = (P_0, \ldots, P_{\ell-1}) = C_0$ and define successor markings by $\mathrm{digit}_{C_0}(\Lambda_{P_i}) = \mathrm{CR}(i)$ for $i \in [\ell]$. This defines $\tilde{\delta}_0$. Now we set $\Xi_0 = \Pi$ and we define $\delta_0 \colon (\Gamma_0 \cup \Xi_0) \times (\Gamma_0 \cup \Xi_0) \to \{-1, 0, 1\}$ by $\delta_0|_{\Gamma_0 \times \Gamma_0} = \tilde{\delta}_0$, $\delta_0|_{\Xi_0 \times \Xi_0} = \delta_\Pi$ and $\delta = 0$ otherwise. We extend the marking $M$ to $\Gamma_0$ by setting $M(P) = 0$ for all $P \in \Gamma_0$. So we obtain a power circuit of the form $(\Gamma_0 \cup \Xi_0, \delta_0)$ with the properties described above.

Now let the power circuit $(\Gamma_i \cup \Xi_i, \delta_i)$ together with the marking $M_i$ be the input for the $i+1$-th iteration meeting the above described invariants. We write $\tilde{\delta}_i = \delta_i|_{\Gamma_i \times \Gamma_i}$. Now we apply the three steps from above:

1. Using UpdateNodes (Lemma 4.14) we compute a reduced power circuit $(\Gamma_i', \delta_i')$ with $(\Gamma_i, \tilde{\delta}_i) \leq (\Gamma_i', \delta_i')$ such that for every $P \in \mathrm{Min}(\Xi_i)$ there is some $Q \in \Gamma_i'$ with $\varepsilon(Q) = \varepsilon(P)$.
2. We apply the procedure ExtendChains (Lemma 4.15) with $\mu = \lceil \log(|\mathrm{Min}(\Xi_i)|) \rceil + 1$ in order to extend each maximal

chain in $(\Gamma_i', \delta_i')$ by at most $\lceil \log(|\mathrm{Min}(\Xi_i)|) \rceil + 1$ nodes. Notice that $\lceil \log(|\mathrm{Min}(\Xi_i)|) \rceil + 1 \leq \lceil \log(|\Pi|) \rceil + 1$ and so, as $\Gamma_0 \leq \Gamma_i'$, the condition $\mu \leq \left\lfloor \frac{2^{|C_0(\Gamma_i')|+1}}{3} \right\rfloor$ in Lemma 4.15 is satisfied. The result of this step is denoted by $(\Gamma_i'', \delta_i'')$.

3. We apply UpdateMarkings (Lemma 4.16) to obtain markings $\widetilde{M}_i$ and $\tilde{\Lambda}_P$ for $P \in \Xi_i \setminus \mathrm{Min}(\Xi_i)$ on $\Gamma_i'' \cup (\Xi_i \setminus \mathrm{Min}(\Xi_i))$ such that $\varepsilon(\widetilde{M}_i) = \varepsilon(M_i)$ and $\varepsilon(\tilde{\Lambda}_P) = \varepsilon(\Lambda_P)$. Observe that these markings restricted to $\Gamma_i''$ are compact.

4. At the end of each iteration, we set $\Gamma_{i+1} = \Gamma_i''$ and $\Xi_{i+1} = \Xi_i \setminus \mathrm{Min}(\Xi_i)$ and $M_{i+1} = \widetilde{M}_i$. Finally, $\delta_{i+1}$ is defined as $\delta_i''$ on $\Gamma_{i+1}$ and via the successor markings $\tilde{\Lambda}_P$ for $P \in \Xi_{i+1}$.

After $\mathrm{depth}(\Pi)+1$ iterations, we reach $\Xi_{d+1} = \Xi_d \setminus \mathrm{Min}(\Xi_d) = \emptyset$ where $d = \mathrm{depth}(\Pi)$. In this case we do not change the resulting power circuit any further. It is clear from Lemma 4.14, Lemma 4.15 and Lemma 4.16 that throughout the above-mentioned invariants are maintained. Thus, $(\Gamma, \delta) = (\Gamma_{d+1}, \delta_{d+1})$ is a reduced power circuit and for every node $P \in \Pi$ there exists a node $Q \in \Gamma_{d+1}$ such that $\varepsilon(Q) = \varepsilon(P)$ and $M_{d+1}$ is a compact marking on $\Gamma_{d+1}$ with $\varepsilon(M_{d+1}) = \varepsilon(M)$.

Claim 4.17. *Let $d = \mathrm{depth}(\Pi)$ and $\Gamma_0, \ldots, \Gamma_{d+1}$ be as constructed above. Then for all $i$ we have*

- $|\mathcal{C}_{\Gamma_i}| \leq |\Pi| + 1$,
- $|\Gamma_i| \leq (|\Pi| + 1)^2 \cdot (\log(|\Pi|) + 2)$.

Proof. According to Lemma 4.14 and Lemma 4.15, we have $\left|\mathcal{C}_{\Gamma_{i+1}}\right| \leq |\mathcal{C}_{\Gamma_i}| + |\mathrm{Min}(\Xi_i)|$. Further observe that $\Pi$ is the disjoint union of the $\mathrm{Min}(\Xi_j)$ for $j \in [0\,..\,d]$. Since $|\mathcal{C}_{\Gamma_0}| = 1$, we obtain for all $i \in [0\,..\,d]$ that

$$\begin{aligned} \left|\mathcal{C}_{\Gamma_{i+1}}\right| &\leq |\mathcal{C}_{\Gamma_i}| + |\mathrm{Min}(\Xi_i)| \\ &\leq 1 + \sum_{0 \leq j \leq i} |\mathrm{Min}(\Xi_j)| \leq |\Pi| + 1. \end{aligned} \tag{4.18}$$

Again by Lemma 4.15 and Lemma 4.14, we have

$$\begin{aligned}|\Gamma_{i+1}| &\leq |\Gamma'_i| + \left|C_{\Gamma'_i}\right| \cdot (\lceil \log(|\mathrm{Min}(\Xi_i)|)\rceil + 1)\\ &\leq |\Gamma_i| + |\mathrm{Min}(\Xi_i)| + (|\mathcal{C}_{\Gamma_i}| + |\mathrm{Min}(\Xi_i)|) \cdot (\lceil \log(|\mathrm{Min}(\Xi_i)|)\rceil + 1)\\ &\leq |\Gamma_i| + |\mathrm{Min}(\Xi_i)| + (|\Pi| + 1) \cdot (\lceil \log(|\Pi|)\rceil + 1)\,.\end{aligned}$$

where the last line is due to (4.18). Since $|\Gamma_0| = \lceil \log(|\Pi|)\rceil + 1$, we obtain by induction that

$$\begin{aligned}|\Gamma_i| &\leq |\Gamma_0| + \sum_{0\leq j\leq i-1} |\mathrm{Min}(\Xi_j)| + i \cdot (|\Pi| + 1) \cdot (\log(|\Pi|) + 2)\\ &\leq (\lceil \log(|\Pi|)\rceil + 1) + |\Pi| + i \cdot (|\Pi| + 1) \cdot (\log(|\Pi|) + 2)\\ &\leq (i+1) \cdot (|\Pi| + 1) \cdot (\log(|\Pi|) + 2)\end{aligned}$$

for all $i \in [1\,..\,d+1]$. The last inequality is due to the fact that $|\Pi| + 1 \geq 2$ and $\log(|\Pi|) + 2 \geq 2$. Since $d + 1 \leq |\Pi|$, we obtain $|\Gamma_i| \leq (|\Pi| + 1)^2 \cdot (\log(|\Pi|) + 2)$. □

Let $D \in \mathbb{N}$ and assume that $\mathrm{depth}(\Pi) \leq D$. By Lemma 4.14, Lemma 4.15 and Lemma 4.16, each iteration of the three steps above can be done in $\mathsf{TC}^0$. Notice here that the construction of the markings $\widetilde{M}_i$ and $\tilde{\Lambda}_P$ during UpdateMarkings can be done in parallel—so it is in $\mathsf{TC}^0$, although Lemma 4.16 is stated only for a single marking. Now, the crucial observation is that, due to Claim 4.17, the input size for each iteration is polynomial in the original input size of $(\Pi, \delta_\Pi)$. Therefore, we can compose the individual iterations and obtain a circuit of polynomial size and depth bounded by $\mathcal{O}(D)$ as described in Lemma 2.5. Thus, we have described a $\mathsf{DepParaTC}^0$ circuit (parametrized by $\mathrm{depth}(\Pi)$) for the problem of computing a reduced form for $(\Pi, \delta_\Pi)$. This completes the proof of Theorem 4.12. □

Remark 4.19. *(1) While Theorem 4.12 is only stated for one input marking, the construction works within the same complexity bounds for any number of markings on* $(\Pi, \delta_\Pi)$ *since during* UpdateMarkings *these all can be updated in parallel.*

*(2) Moreover, note that for every maximal chain* $C \in \mathcal{C}_\Gamma$ *there exists a node* $Q \in \Pi$ *(i.e., in the original power circuit) such*

*that* $\varepsilon(Q) = \varepsilon(\text{start}(C))$. *This is because new chains are only created during* UpdateNodes, *the other steps only extend already existing chains.*

*(3) Further observe that* $|\sigma(\widetilde{M})| \leq |\sigma(M)|$. *Looking at the construction of* $\widetilde{M}$ *we see that we first make sure that* $M$ *does not mark two nodes of the same value, then we make the marking compact. Both operations do not increase the number of nodes in the support of the marking.*

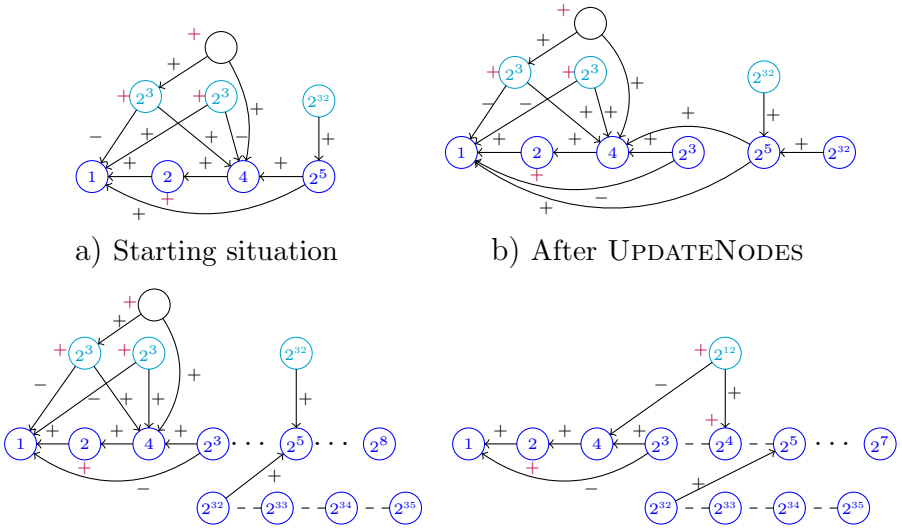



Figure 4.2: The three steps of power circuit reduction. The already reduced part consist of blue nodes and $\text{Min}(\Xi_i)$ is colored in cyan. The red signs indicate a marking. Three dots $\cdots$ in between two nodes mean that we omitted some nodes. A dashed edge - - means that we actually omitted the outgoing edges of the right node (color figure online).

Example 4.20. In Figure 4.2 we illustrate what happens in the steps UpdateNodes, ExtendChains and UpdateMarkings during the reduction process. Picture a) shows our starting situation. In b) we already inserted the nodes of value $2^3$ and $2^{32}$ into the reduced part. Now the reduced part consists of three chains:

one starting at the node of value 1 and the nodes $2^5$ and $2^{32}$ as chains of length 1. Because $|\mathrm{Min}(\Xi)| = 3$, we have to extend each chain by three nodes or until two chains merge. So in c) we obtain two chains, one from 1 to $2^8$ and the one from $2^{32}$ to $2^{35}$. In d) we then updated the markings and discarded the nodes from $\mathrm{Min}(\Xi)$. ◊

Example 4.21. In Figure 4.3 we give an example of the complete power circuit reduction process by showing the result after each iteration. We start with a non-reduced power circuit of depth 2 in a). This power circuit has size 5, so we first construct the starting chain of length 4 in b). Part c) and d) show the result after inserting layer 0 and layer 1, respectively. In e) we finally inserted all layers and thus have constructed the reduced power circuit. ◊

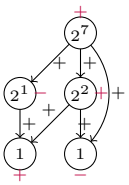
For comparing two markings $L$ and $M$ on an arbitrary power circuit, we can proceed as follows: first compute the difference (Lemma 4.9), then reduce the power circuit (Theorem 4.12) and, finally, compare the resulting compact marking with zero (Proposition 4.6). This shows:

Corollary 4.22. *Let* $\triangle \in \{=, \neq, <, \leq, >, \geq\}$. *The following is in* $\mathsf{DepParaTC}^0$ *parametrized by* $\mathrm{depth}(\Pi)$*:*

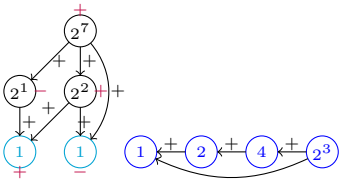
Input: *A power circuit* $(\Pi, \delta_\Pi)$ *together with markings* $L, M$ *on* $\Pi$.
Question: *Is* $\varepsilon(L) \triangle \varepsilon(M)$*?*

Proof (of Proposition A). When assuming $\mathrm{depth}(\Pi) \leq C \cdot \log |\Pi|$, by Lemma 2.4, we obtain Proposition A as an immediate consequence of Corollary 4.22. □

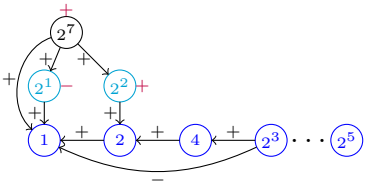
Remark 4.23. *By Corollary 4.22 comparing two numbers* $m_1$ *and* $m_2$ *represented by a* $(0, +, -, 2^x)$*-circuit* $\mathcal{C}$ *is in* $\mathsf{DepParaTC}^0$ *parametrized by* $\mathrm{depth}_{2^x}(\mathcal{C}) + \log^2 |\mathcal{C}|$ *if the input of every* $2^x$*-gate of* $\mathcal{C}$ *is nonnegative: by Proposition 4.11 we can find a power circuit* $(\Pi, \delta)$ *with* $\mathrm{depth}(\Pi) \leq \mathrm{depth}_{2^x}(\mathcal{C})$ *and markings* $M_1$ *and* $M_2$ *evaluating to* $m_1$ *and* $m_2$ *in* $\mathsf{NC}^2 \subseteq \mathsf{TC}^2$*. It remains to compare* $\varepsilon(M_1)$ *and*



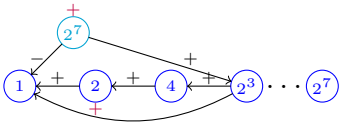
a) Non-reduced circuit



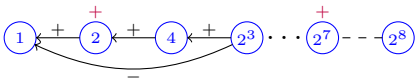
b) With initial chain $\Gamma_0$.



c) After inserting layer 0



d) After inserting layer 1



e) After inserting layer 2

Figure 4.3: The full process of power circuit reduction—inserting layer after layer. For an explanation of the colors, see Figure 4.2.

$\varepsilon(M_2)$. *Note that by a slight abuse of our notation we can write* $\mathsf{TC}^2$ *as* $\mathsf{DepParaTC}^0$ *parametrized by* $\log^2|\mathcal{C}|$. *This explains the additional* $\log^2|\mathcal{C}|$ *above.*

**4.4. Operations with floating point numbers.** In the following, we want to represent a number $r \in \mathbb{Z}[1/2]$ using markings in a power circuit. For this, we use a floating point representation. Observe that for each such $r \in \mathbb{Z}[1/2] \setminus \{0\}$ there exist unique $u, e \in \mathbb{Z}$ with $u$ odd such that $r = u \cdot 2^e$.

LEMMA 4.24. *The following problem is in* $\mathsf{DepParaTC}^0$ *parametrized by* $\operatorname{depth}(\Pi)$*:*

Input: *A power circuit* $(\Pi, \delta_\Pi)$ *with a marking* $M$ *on* $\Pi$.
Output: *A power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ *with markings* $E, U$ *on* $\tilde{\Pi}$ *such that* $\varepsilon(M) = \varepsilon(U) \cdot 2^{\varepsilon(E)}$ *with* $\varepsilon(U)$ *odd.*

*In addition,* $(\Pi, \delta_\Pi) \leq (\tilde{\Pi}, \delta_{\tilde{\Pi}})$, $\operatorname{depth}(\tilde{\Pi}) \leq \max(\operatorname{depth}(\Pi), 2)$ *and* $|\tilde{\Pi}| \in \mathcal{O}(|\Pi|)$.

PROOF. First, note that we are searching for a marking representing the maximal $e \in \mathbb{Z}$ with $2^e \mid \varepsilon(M)$. For finding $e$, we need the compact representation of $M$. Therefore, we construct the reduced form $(\Gamma, \delta)$ of $\Pi$ and a compact marking $\widetilde{M}$ on $\Gamma$ such that $\varepsilon(\widetilde{M}) = \varepsilon(M)$. According to Theorem 4.12 this is possible in $\mathsf{DepParaTC}^0$. Now we have $\varepsilon(M) = \sum_{i=1}^{k} \widetilde{M}(Q_i) \cdot 2^{\varepsilon(\Lambda_{Q_i})}$ where $\sigma(\widetilde{M}) = \{Q_1, \ldots, Q_k\} \subseteq \Gamma$. We assume that the $Q_i$ are ordered according to their value, i.e., $\varepsilon(Q_i) < \varepsilon(Q_j)$ for $i < j$. Hence, $e = \varepsilon(\Lambda_{Q_1})$.

Before we can define the markings $U$ and $E$, we have to introduce some new nodes. First, we add $\lfloor \log(|\Gamma|) \rfloor$ new nodes to $\Pi$ each of value 1 (i.e., with empty successor marking). Then for each $j \in [0 \,..\, \lfloor \log(|\Gamma|) \rfloor]$ we create a node of value $2^j$ and depth 1 in the following way: the successor marking of such a node marks exactly $j$ nodes of value 1 with $+1$ and all the other nodes with 0.

In order to define $U$, we aim for adding a node $S_i$ to $\Pi$ with $\varepsilon(\Lambda_{S_i}) = \varepsilon(\Lambda_{Q_i}) - e$ for each $i \in [1 \,..\, k]$. We proceed as follows: For each $i \in [1 \,..\, k]$, let $C_i \in \mathcal{C}_\Gamma$ denote the maximal chain to which $Q_i$ belongs. Note that for different $i$ these chains could be equal. By Remark 4.19, we know that there exist nodes $R_1, \ldots, R_k \in \Pi$ such that $\varepsilon(R_i) = \varepsilon(\mathrm{start}(C_i))$ for $i \in [1 \,..\, k]$. To find the nodes $R_i$, we can for example remember the equivalence classes we obtain during the reduction process. Now there exist $m_i \in \mathbb{N}$ with $m_i \in [0 \,..\, |\Gamma|]$ such that $\varepsilon(\Lambda_{Q_i}) = \varepsilon(\Lambda_{R_i}) + m_i$. We can find $m_i$ as the difference of the indices of $Q_i$ and $\mathrm{start}(C_i)$ in the sorted order of $\Gamma$, and so we can find all the $m_i$ in $\mathsf{AC}^0$. Note that the binary representation of $m_i$ uses at most $\lfloor \log(|\Gamma|) \rfloor + 1$ bits. We define markings $M_i$ on the newly defined nodes of depth 1 using the binary representation of $m_i$ such that $\varepsilon(M_i) = m_i$ for $i \in [1 \,..\, k]$. Now we are ready to construct the marking $E$ with $\varepsilon(E) = \varepsilon(\Lambda_{R_1}) + \varepsilon(M_1)$ using Lemma 4.9. Observe that no cloning is necessary here because the markings have disjoint support.

We now want to define a marking $U$, with $\varepsilon(U) = \varepsilon(M) \cdot 2^{-\varepsilon(E)}$. For every $i \in [1 \,..\, k]$ we create a node $S_i$ with $\varepsilon(\Lambda_{S_i}) = \varepsilon(\Lambda_{R_i}) + \varepsilon(M_i) - \varepsilon(E)$ (again using Lemma 4.9), so in particular, $\varepsilon(S_i) = \varepsilon(Q_i) \cdot 2^{-\varepsilon(E)}$ (notice that $\varepsilon(S_1) = 1$). Because $E$ and $M_i$ could have supports with non-trivial intersection (as well as $E$ and $\Lambda_{R_i}$),

we have to clone the nodes in $\sigma(E)$ for the addition. However, for all $i$ together, a single clone of $\sigma(E)$ is sufficient. Then the marking $U$ with $U(S_i) = \widetilde{M}(Q_i)$ for $i \in [1 \,..\, k]$ is the marking $U$ we searched for.

Regarding the size of $\tilde{\Pi}$, observe that to define the markings $M_i$, we insert $2 \cdot \lfloor \log(|\Gamma|) \rfloor + 1$ nodes. By cloning the nodes in $\sigma(E)$, we add at most $\lfloor \log(|\Gamma|) \rfloor + 1 + |\Pi|$ additional nodes. By Remark 4.19(3), we know that $|\sigma(\widetilde{M})| \leq |\sigma(M)| \leq |\Pi|$, so we insert at most $|\Pi|$ nodes when inserting the nodes $S_i$. According to Claim 4.17, we have $\log(|\Gamma|) \in \mathcal{O}(\log(|\Pi|))$. Hence, $|\tilde{\Pi}| \in \mathcal{O}(|\Pi|)$.

Considering the depth, when inserting the new nodes of depth 1, the depth only increases if $\operatorname{depth}(\Pi) = 0$. When inserting a node $S_i$, the depth increases only if $\operatorname{depth}(\Pi) \leq 1$. □

Definition 4.25. *A power circuit representation of $r \in \mathbb{Z}[1/2]$ consists of a power circuit $(\Pi, \delta_\Pi)$ together with a pair of markings $(U, E)$ on $\Pi$ such that $\varepsilon(U)$ is either zero or odd and $r = \varepsilon(U) \cdot 2^{\varepsilon(E)}$.*

Lemma 4.26. *(a) The following problems are in $\mathsf{TC}^0$:*

| | |
|---|---|
| Input: | *A power circuit representation for $r \in \mathbb{Z}[1/2]$ over a power circuit $(\Pi, \delta_\Pi)$ and a marking $M$ on $\Pi$.* |
| Output: | *A power circuit representation of $r \cdot 2^{\varepsilon(M)}$ over a power circuit $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$.* |

| | |
|---|---|
| Input: | *A power circuit representation for $r \in \mathbb{Z}[1/2]$ over a power circuit $(\Pi, \delta_\Pi)$.* |
| Output: | *A power circuit representation of $-r$ over $(\Pi, \delta_\Pi)$.* |

| | |
|---|---|
| Input: | *Power circuit representations for $r, s \in \mathbb{Z}[1/2]$ over a power circuit $(\Pi, \delta_\Pi)$ such that $\frac{r}{s}$ is a power of two.* |
| Output: | *A marking $L$ in a power circuit $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ such that $\varepsilon(L) = \log(\frac{r}{s})$.* |

*(b) The following problems are in* $\mathsf{DepParaTC}^0$ *parametrized by* $\mathrm{depth}(\Pi)$*:*

| | |
|---|---|
| Input: | *A power circuit* $(\Pi, \delta_\Pi)$ *and a marking* $M$ *on* $\Pi$. |
| Output: | *A power circuit representation of* $\varepsilon(M) \in \mathbb{Z}[1/2]$ *over a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$. |

| | |
|---|---|
| Input: | $r, s \in \mathbb{Z}[1/2]$ *given as power circuit representations over a power circuit* $(\Pi, \delta_\Pi)$. |
| Output: | *A power circuit representation of* $r + s$ *over a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$. |

| | |
|---|---|
| Input: | *A power circuit representation for* $r \in \mathbb{Z}[1/2]$ *over a power circuit* $(\Pi, \delta_\Pi)$ |
| Question: | *Is* $r \triangle 0$ *for* $\triangle \in \{=, \neq, <, \leq, >, \geq\}$*?* |

| | |
|---|---|
| Input: | *A power circuit representation for* $r \in \mathbb{Z}[1/2]$ *over a power circuit* $(\Pi, \delta_\Pi)$. |
| Output: | *Is* $r \in \mathbb{Z}$*? If yes, a marking* $M$ *in a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ *such that* $\varepsilon(M) = r$. |

*In all cases we have the following bounds:* $(\Pi, \delta_\Pi) \leq (\tilde{\Pi}, \delta_{\tilde{\Pi}})$, $|\tilde{\Pi}| \in \mathcal{O}(|\Pi|)$, *and* $\mathrm{depth}(\tilde{\Pi}) = \mathrm{depth}(\Pi) + \mathcal{O}(1)$.

Proof. During the whole proof, let $U, V, E, F$ be markings in $\Pi$ such that $\varepsilon(U)$, $\varepsilon(V)$ are odd, $r = \varepsilon(U) \cdot 2^{\varepsilon(E)}$ and $s = \varepsilon(V) \cdot 2^{\varepsilon(F)}$.

**Part (a):** We have $r \cdot 2^{\varepsilon(M)} = \varepsilon(U) \cdot 2^{\varepsilon(E)+\varepsilon(M)}$. According to Lemma 4.9(a), the marking $E + M$ can be obtained in $\mathsf{TC}^0$ as a marking in a power circuit $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ that satisfies all the required properties. The computation of $-r$ is clear by Lemma 4.9.

If $\frac{r}{s}$ is a power of two, we know that $\varepsilon(U) = \varepsilon(V)$, and so $\frac{r}{s} = 2^{\varepsilon(E)-\varepsilon(F)}$. Again Lemma 4.9 finishes the proof of part (a).

**Part (b):** The first point is due to Lemma 4.24. For the addition, first observe that

$$r + s = \varepsilon(U) \cdot 2^{\varepsilon(E)} + \varepsilon(V) \cdot 2^{\varepsilon(F)} = 2^{\varepsilon(E)} \cdot (\varepsilon(U) + \varepsilon(V) \cdot 2^{\varepsilon(F)-\varepsilon(E)})$$

We can decide in $\mathsf{DepParaTC}^0$ whether $\varepsilon(E) \leq \varepsilon(F)$ using Corollary 4.22. W. l. o. g. let $\varepsilon(E) \leq \varepsilon(F)$ (otherwise we switch the roles of $r$ and $s$). Next, we construct a marking $K$ in a power circuit $(\Pi', \delta_{\Pi'})$ such that $\varepsilon(K) = \varepsilon(U) + \varepsilon(V) \cdot 2^{\varepsilon(F)-\varepsilon(E)}$. According to Lemma 4.9 and because $\varepsilon(F) - \varepsilon(E) \geq 0$, this is possible in $\mathsf{TC}^0$ and such that $|\Pi'| \in \mathcal{O}(|\Pi|)$ and $\operatorname{depth}(\Pi') \leq \operatorname{depth}(\Pi) + 1$. Now, according to Lemma 4.24 we can construct markings $W$ and $G$ in a power circuit $(\Pi'', \delta_{\Pi''})$ such that $\varepsilon(W)$ is odd and $\varepsilon(K) = \varepsilon(W) \cdot 2^{\varepsilon(G)}$ in $\mathsf{DepParaTC}^0$. In addition, $|\Pi''| \in \mathcal{O}(|\Pi'|)$ and $\operatorname{depth}(\Pi'') \leq \max(\operatorname{depth}(\Pi'), 2)$. Now according to Lemma 4.9 a marking $H$ with $\varepsilon(H) = \varepsilon(E) + \varepsilon(G)$ can be obtained as marking in a power circuit $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ with $|\tilde{\Pi}| \in \mathcal{O}(|\Pi|)$ and $\operatorname{depth}(\tilde{\Pi}) = \operatorname{depth}(\Pi) + \mathcal{O}(1)$. Then the power circuit $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ together with the markings $W$ and $H$ is the power circuit representation for $r + s$, satisfying the required properties.

To decide if $r \triangle 0$, we just have to check if $\varepsilon(U) \triangle 0$. According to Corollary 4.22, this is possible in $\mathsf{DepParaTC}^0$. To decide if $r \in \mathbb{Z}$, since $\varepsilon(U)$ is odd, we just need to decide if $\varepsilon(E) \geq 0$. Again, this can be done using Corollary 4.22. In the affirmative case, we just have to apply Lemma 4.9(c) to produce the desired output. □

## 5. The word problem of the Baumslag group

Before we start solving the word problem of the Baumslag group, let us fix our notation from group theory.

**Group presentations.** A group $G$ is *finitely generated* if there is some finite set $\Sigma$ and a surjective monoid homomorphism $\eta\colon \Sigma^* \to G$ (called a presentation). Usually, we do not write the homomorphism $\eta$ and treat words over $\Sigma$ both as words and as their images under $\eta$. We write $v =_G w$ with the meaning that $\eta(v) = \eta(w)$. If $\Sigma = S \cup S^{-1}$ where $S^{-1}$ is some disjoint set of formal inverses and $R \subseteq \Sigma^* \times \Sigma^*$ is some set of relations, we write $\langle \Sigma \mid R \rangle$ for the group $\Sigma^*/C(R)$ where $C(R)$ is the congruence generated by $R$ together with the relations $aa^{-1} = a^{-1}a = 1$ for $a \in \Sigma$. If $R$ is finite, $G$ is called *finitely presented*.

The word problem for a fixed group $G$ with presentation $\eta\colon \Sigma^* \to G$ is as follows:

Input: A word $w \in \Sigma^*$
Question: Is $w =_G 1$?

For further background on group theory, we refer to Lyndon & Schupp (2001).

**The Baumslag–Solitar group.** Recall that the set of dyadic fractions with addition as group operation is denoted by $\mathbb{Z}[1/2] = \{p/2^q \in \mathbb{Q} \mid p, q \in \mathbb{Z}\}$. The Baumslag–Solitar group is defined by

$$\mathbf{BS}_{1,2} = \langle a, t \mid tat^{-1} = a^2 \rangle .$$

We have $\mathbf{BS}_{1,2} \cong \mathbb{Z}[1/2] \rtimes \mathbb{Z}$ via the isomorphism $a \mapsto (1, 0)$ and $t \mapsto (0, 1)$. Here, $\rtimes$ denotes the so-called *semi-direct product*. The elements of $\mathbb{Z}[1/2] \rtimes \mathbb{Z}$ are pairs $(r, m)$ with $r \in \mathbb{Z}[1/2]$ and $m \in \mathbb{Z}$. The multiplication is defined by $(r, m)\cdot(s, n) = (r+2^m s, m+n)$. Inverses can be computed by the formula $(r, m)^{-1} = (-r \cdot 2^{-m}, -m)$. In the following we use $\mathbf{BS}_{1,2}$ and $\mathbb{Z}[1/2] \rtimes \mathbb{Z}$ as synonyms.

**The Baumslag group.** A convenient way to understand the Baumslag group $\mathbf{G}_{1,2}$ is as an HNN extension (named after Graham Higman, Bernhard H. Neumann and Hanna Neumann, for a precise definition, see Lyndon & Schupp (2001)) of the Baumslag–Solitar group:

$$\begin{aligned}\mathbf{G}_{1,2} &= \langle \mathbf{BS}_{1,2}, b \mid bab^{-1} = t \rangle \\ &= \langle a, t, b \mid tat^{-1} = a^2, bab^{-1} = t \rangle .\end{aligned}$$

Indeed, due to $bab^{-1} = t$, we can remove $t$ and we obtain exactly the presentation $\langle a, b \mid bab^{-1}a = a^2bab^{-1} \rangle$. Moreover, $\mathbf{BS}_{1,2}$ is a subgroup of $\mathbf{G}_{1,2}$ via the canonical embedding and we have $b(q, 0)b^{-1} = (0, q)$ if $q \in \mathbb{Z}$; so a conjugation by $b$ "flips" the two components of the semi-direct product if possible. Henceforth, we will use the alphabet $\Sigma = \{1, a, a^{-1}, t, t^{-1}, b, b^{-1}\}$ to represent elements of $\mathbf{G}_{1,2}$ (the letter 1 represents the group identity; it is there for padding reasons). Note that this way of writing $\mathbf{G}_{1,2}$ as an HNN extension is also the way how the Magnus breakdown procedure works.

**Britton reductions.** Britton reductions are a standard way to solve the word problem in HNN extensions. Here we define them for the special case of $\mathbf{G}_{1,2}$. Let

$$\Delta = \mathbf{BS}_{1,2} \cup \{b, b^{-1}\}$$

be an infinite alphabet (note that $\Sigma \subseteq \Delta$). A word $w \in \Delta^*$ is called *Britton-reduced* if it is of the form

$$w = (s_0, n_0)\beta_1(s_1, n_1) \cdots \beta_\ell(s_\ell, n_\ell)$$

with $\beta_i \in \{b, b^{-1}\}$ and $(s_i, n_i) \in \mathbf{BS}_{1,2}$ for all $i$ (i.e., $w$ does not have two successive letters from $\mathbf{BS}_{1,2}$) and there is no factor of the form $b(q, 0)b^{-1}$ or $b^{-1}(0, k)b$ with $q, k \in \mathbb{Z}$. If $w$ is not Britton-reduced, one can apply one of the rules

$$\begin{aligned} (r, m)(s, n) &\to (r + 2^m s, m + n) \\ b(q, 0)b^{-1} &\to (0, q) \\ b^{-1}(0, k)b &\to (k, 0) \end{aligned}$$

in order to obtain a shorter word representing the same group element. The following lemma is well-known (see also Lyndon & Schupp 2001, Section IV.2).

Lemma 5.1 (Britton's Lemma, 1963, special case for $\mathbf{G}_{1,2}$). *Let $w \in \Delta^*$ be Britton-reduced. Then $w \in \mathbf{BS}_{1,2}$ as a group element if and only if $w$ does not contain any letter $b$ or $b^{-1}$. In particular, $w =_{\mathbf{G}_{1,2}} 1$ if and only if $w = (0, 0)$ or $w = 1$ as a word.*

Example 5.2. Define words $w_0 = t$ and $w_{n+1} = b\, w_n\, a\, w_n^{-1}\, b^{-1}$ for $n \geq 0$. Then we have $|w_n| = 2^{n+2} - 3$ but $w_n =_{\mathbf{G}_{1,2}} t^{\tau(n)}$. While the length of the word $w_n$ is only exponential in $n$, the length of its Britton-reduced form is $\tau(n)$. ◊

**5.1. Conditions for Britton reductions.** The idea to obtain a parallel algorithm for the word problem is to compute a Britton reduction of $uv$ given that both $u$ and $v$ are Britton-reduced. For this, we have to find a maximal suffix of $u$ which cancels with a prefix of $v$. The following lemma is our main tool for finding the longest canceling suffix. It is important to note that for all suffixes the conditions can be checked in parallel.

LEMMA 5.3. *Let* $w = \beta_1(r,m)\beta_2\, x\, \beta_2^{-1}(s,n)\beta_1^{-1} \in \Delta^*$ *with* $\beta_1, \beta_2 \in \{b, b^{-1}\}$ *such that* $\beta_1(r,m)\beta_2$ *and* $\beta_2^{-1}(s,n)\beta_1^{-1}$ *both are Britton-reduced and* $\beta_2 x \beta_2^{-1} =_{\mathbf{G}_{1,2}} (q,k) \in \mathbf{BS}_{1,2}$ *(in particular,* $k = 0$ *and* $q \in \mathbb{Z}$*, or* $q = 0$*).*

*Then* $w \in \mathbf{BS}_{1,2}$ *if and only if the respective condition in the following table is satisfied. Moreover, if* $w \in \mathbf{BS}_{1,2}$*, then* $w =_{\mathbf{G}_{1,2}} \hat{w}$ *according to the last column of the table.*

| $\beta_1$ | $\beta_2$ | *Condition* | $\hat{w}$ |
|---|---|---|---|
| $b$ | $b$ | $r + 2^{m+k}s \in \mathbb{Z}, \quad m+n+k = 0$ | $(0,\ r + 2^{-n}s)$ |
| $b$ | $b^{-1}$ | $r + 2^{m}(q+s) \in \mathbb{Z}, \quad m+n = 0$ | $(0,\ r + 2^{m}(q+s))$ |
| $b^{-1}$ | $b$ | $r + 2^{m+k}s = 0$ | $\left(n + \log(\frac{-r}{s}),\ 0\right)$ |
| $b^{-1}$ | $b^{-1}$ | $r + 2^{m}(q+s) = 0$ | $(m+n,\ 0)$ |

*Notice that in the case* $\beta_1 = b^{-1}$ *and* $\beta_2 = b$*, we have* $r \neq 0$ *and* $s \neq 0$*.*

EXAMPLE 5.4. Let us illustrate with two examples how to read Lemma 5.3. For this, let $w = \beta_1(r_1,m_1)\beta_2\, x\, \beta_2^{-1}(s_1,n_1)\beta_1^{-1} \in \Delta^*$ with the same properties as in Lemma 5.3, in particular, we have $\beta_2\, x\, \beta_2^{-1} =_{\mathbf{G}_{1,2}} (q,k) \in \mathbf{BS}_{1,2}$.

We first consider the case that $\beta_1 = \beta_2 = b$. To check if $w \in \mathbf{BS}_{1,2}$ we have to check if $m_1 + n_1 + k = 0$ and if $r_1 + 2^{m_1+k} \cdot s_1 \in \mathbb{Z}$ according to Lemma 5.3. In order to obtain a formula for $k$, we apply Lemma 5.3 to $\beta_2\, x\, \beta_2^{-1}$ using the rightmost column. We write

$$(q,k) =_{\mathbf{G}_{1,2}} \beta_2\, x\, \beta_2^{-1} = \beta_2(r_2,m_2)\beta_3\, x'\, \beta_3^{-1}(s_2,n_2)\beta_2^{-1}. \tag{5.5}$$

For our example let us assume that $\beta_3 = b$. Then according to Lemma 5.3, $(q,k) = (0, r_2 + 2^{-n_2} \cdot s_2)$. Hence, $w \in \mathbf{BS}_{1,2}$ if and only if $m_1 + n_1 + (r_2 + 2^{-n_2} \cdot s_2) = 0$ and $r_1 + 2^{m_1+r_2+2^{-n_2}\cdot s_2} \cdot s_1 \in \mathbb{Z}$. If both conditions are satisfied, we know that $w =_{\mathbf{G}_{1,2}} (0,\ r_1 + 2^{-n_1}s_1)$.

Now let us consider a more difficult case. Assume that $\beta_1 = b^{-1}$ and $\beta_2 = b$. According to Lemma 5.3, $w \in \mathbf{BS}_{1,2}$ if and only if $r_1 + 2^{m_1+k}s_1 = 0$. To obtain a formula for $k$, we apply again Lemma 5.3 using the rightmost column. We again write $(q,k) =_{\mathbf{G}_{1,2}} \beta_2\, x\, \beta_2^{-1}$ as in (5.5), but here we assume that $\beta_3 = b^{-1}$.

As we assumed $\beta_2 = b$, we have by the last column in Lemma 5.3 that

$$\begin{aligned}(q,k) =_{\mathbf{G}_{1,2}} \beta_2\, x\, \beta_2^{-1} &= \beta_2(r_2,m_2)\beta_3\, x'\, \beta_3^{-1}(s_2,n_2)\beta_2^{-1}\\ &=_{\mathbf{G}_{1,2}} (0, r_2 + 2^{m_2}(q' + s_2))\end{aligned}$$

if $\beta_3 x' \beta_3^{-1} =_{\mathbf{G}_{1,2}} (q',k') \in \mathbf{BS}_{1,2}$. This means that $k = r_2 + 2^{m_2}(q' + s_2)$. Now we have to find a formula for $q'$. We write

$$(q',k') =_{\mathbf{G}_{1,2}} \beta_3\, x'\, \beta_3^{-1} = \beta_3(r_3,m_3)\beta_4\, x''\, \beta_4^{-1}(s_3,n_3)\beta_3^{-1}.$$

We assume $\beta_4 = b$. Then according to Lemma 5.3 we obtain that $q' = n_3 + \log\left(\frac{-r_3}{s_3}\right)$. This implies that $k = r_2 + 2^{m_2} \cdot (n_3 + \log(\frac{-r_3}{s_3}) + s_2)$. So, to check whether $w \in \mathbf{BS}_{1,2}$, we have to check if

$$r_1 + 2^{m_1 + r_2 + 2^{m_2} \cdot (n_3 + \log(\frac{-r_3}{s_3}) + s_2)} \cdot s_1 = 0.$$

If this is the case, then $w =_{\mathbf{G}_{1,2}} \left(n_1 + \log\left(\frac{-r_1}{s_1}\right), 0\right)$. ◊

Proof (of Lemma 5.3). We distinguish the two cases $\beta_2 = b$ and $\beta_2 = b^{-1}$. Each case consists of two sub-cases depending on $\beta_1$.

**Case $\beta_2 = b$:** Since $\beta_2 x \beta_2^{-1} \in \mathbf{BS}_{1,2}$, we have $\beta_2 x \beta_2^{-1} =_{\mathbf{G}_{1,2}} (0,k)$ for some $k \in \mathbb{Z}$. Therefore, we obtain

$$\begin{aligned}(r,m)\beta_2\, x\, \beta_2^{-1}(s,n) &=_{\mathbf{G}_{1,2}} (r,m)(0,k)(s,n)\\ &=_{\mathbf{G}_{1,2}} (r, m+k)(s,n)\\ &=_{\mathbf{G}_{1,2}} (r + 2^{m+k}s,\ m+k+n).\end{aligned}$$

Thus, if $\beta_1 = b$, we have $w \in \mathbf{BS}_{1,2}$ if and only if $r + 2^{m+k}s \in \mathbb{Z}$ and $m + n + k = 0$. Moreover, if the latter conditions are satisfied, we have $w =_{\mathbf{G}_{1,2}} b(r + 2^{m+k}s,\, 0)b^{-1} = b(r + 2^{-n}s,\, 0)b^{-1} =_{\mathbf{G}_{1,2}} (0,\, r + 2^{-n}s)$.

On the other hand, if $\beta_1 = b^{-1}$, it follows that $w \in \mathbf{BS}_{1,2}$ if and only if $r + 2^{m+k}s = 0$. Notice that in this case, by the assumption that $\beta_1(r,m)\beta_2$ and $\beta_2^{-1}(s,n)\beta_1^{-1}$ are Britton-reduced, we have $r \neq 0$ and $s \neq 0$. Therefore, if the condition $r + 2^{m+k}s = 0$ is satisfied, we have $k = \log(\frac{-r}{2^m s})$. Hence, under this condition, we have $w =_{\mathbf{G}_{1,2}} b^{-1}(0,\, m+k+n)b = b^{-1}(0,\, m + \log(\frac{-r}{2^m s}) + n)b =_{\mathbf{G}_{1,2}} (n + \log(\frac{-r}{s}), 0)$ (because $\log(\frac{-r}{2^m s}) = \log(\frac{-r}{s}) - m$).

**Case $\beta_2 = b^{-1}$:** In this case, we do a similar computation:

$$\begin{aligned}(r,m)\beta_2\, x\, \beta_2^{-1}(s,n) &=_{\mathbf{G}_{1,2}} (r,m)(q,0)(s,n)\\ &=_{\mathbf{G}_{1,2}} (r,m)(q+s,n)\\ &=_{\mathbf{G}_{1,2}} (r+2^m(q+s),\, m+n)\end{aligned}$$

with $q \in \mathbb{Z}$. Again, let us consider the case $\beta_1 = b$ first. In this case we have $w \in \mathbf{BS}_{1,2}$ if and only if $r + 2^m(q+s) \in \mathbb{Z}$ and $m + n = 0$. If these conditions are satisfied, we have $w =_{\mathbf{G}_{1,2}} b(r+2^m(q+s),\, 0)b^{-1} =_{\mathbf{G}_{1,2}} (0,\, r+2^m(q+s))$.

Finally, if $\beta_1 = b^{-1}$, it follows that $w \in \mathbf{BS}_{1,2}$ if and only if $r + 2^m(q+s) = 0$. If this applies, we have $w =_{\mathbf{G}_{1,2}} b^{-1}(r+2^m(q+s),\, m+n)b =_{\mathbf{G}_{1,2}} (m+n, 0)$. □

Let us fix the following notation for elements $u, v \in \mathbf{G}_{1,2}$ written as words over $\Delta$:

$$\begin{aligned}u &= (r_h, m_h)\beta_h \cdots (r_1, m_1)\beta_1(r_0, m_0),\\ v &= (s_0, n_0)\tilde{\beta}_1(s_1, n_1)\cdots\tilde{\beta}_\ell(s_\ell, n_\ell)\end{aligned} \tag{5.6}$$

with $(r_j, m_j), (s_j, n_j) \in \mathbb{Z}[1/2] \rtimes \mathbb{Z}$ and $\beta_j, \tilde{\beta}_j \in \{b, b^{-1}\}$. We define

$$uv[i,j] = \beta_i(r_{i-1}, m_{i-1})\cdots\beta_1(r_0, m_0)\ (s_0, n_0)\tilde{\beta}_1\cdots(s_{j-1}, n_{j-1})\tilde{\beta}_j.$$

Notice that as an immediate consequence of Britton's Lemma we obtain that, if $u$ and $v$ as in (5.6) are Britton-reduced and $uv[i,i] \in \mathbf{BS}_{1,2}$ for some $i$, then also $uv[j,j] \in \mathbf{BS}_{1,2}$ for all $j \le i$. Moreover, $uv$ is Britton-reduced if and only if $\beta_1(r_0, m_0)(s_0, n_0)\tilde{\beta}_1 \notin \mathbf{BS}_{1,2}$.

For $\ell \in \mathbb{N}$ let $\mathcal{X}_\ell$ denote some set of $\ell$ variables. Denote by $\mathrm{PowExp}(\mathcal{X}_\ell)$ the set of expressions which can be made up from the variables $\mathcal{X}_\ell$ using the operations $+$, $-$, $(r,s) \mapsto r \cdot 2^s$ if $s \in \mathbb{Z}$ (and undefined otherwise), and $(r,s) \mapsto \log(r/s)$ if $\log(r/s) \in \mathbb{Z}$ (and undefined otherwise).

Lemma 5.7. *For every $\vec{\beta} \in \{b, b^{-1}\}^4$ there are expressions $\theta_{\vec{\beta}}$, $\xi_{\vec{\beta}}$, $\varphi_{\vec{\beta}}$, $\psi_{\vec{\beta}} \in \mathrm{PowExp}(\mathcal{X}_{12})$ such that the following holds: Let $i \ge 4$ and let $u, v \in \mathbf{G}_{1,2}$ as in (5.6) be Britton-reduced and assume that $uv[i-1, i-1] \in \mathbf{BS}_{1,2}$ and $\beta_i = \tilde{\beta}_i^{-1}$ and let $V_i = \{r_j, s_j, m_j, n_j \mid j \in \{i-1, i-2, i-3\}\}$. If $\vec{\beta} = (\beta_i, \beta_{i-1}, \beta_{i-2}, \beta_{i-3})$, then*

*(i)* $uv[i,i] \in \mathbf{BS}_{1,2}$ *if and only if* $\theta_{\vec{\beta}}(V_i) \in \mathbb{Z}$ *and* $\xi_{\vec{\beta}}(V_i) = 0$,

*(ii) if* $uv[i,i] \in \mathbf{BS}_{1,2}$, *then* $uv[i,i] =_{\mathbf{G}_{1,2}} \big(\varphi_{\vec{\beta}}(V_i), \psi_{\vec{\beta}}(V_i)\big)$.

Be aware that here we have to read the set $V_i$ of cardinality 12 as an assignment to the variables $\mathcal{X}_{12}$. In particular, given that $uv[i-1,i-1] \in \mathbf{BS}_{1,2}$, one can decide whether $uv[i,i] \in \mathbf{BS}_{1,2}$ by looking at only constantly many letters of $uv$—this is the crucial observation we shall be using for describing an NC algorithm for the word problem of $\mathbf{G}_{1,2}$ (see Lemma 5.9 below).

Proof. We follow the approach of Example 5.4. By assumption we know that there exist $q, k \in \mathbb{Z}$ such that $uv[i-1,i-1] =_{\mathbf{G}_{1,2}} (q,k) \in \mathbf{BS}_{1,2}$. According to the conditions in Lemma 5.3, to show Lemma 5.7 it suffices to find expressions $\varphi_{\vec{\beta}}(V_i)$, $\psi_{\vec{\beta}}(V_i)$ for $q$ and $k$ respectively. If $(\beta_{i-1}, \beta_{i-2}) \neq (b, b^{-1})$, this follows directly from the rightmost column in Lemma 5.3. Otherwise, we know that $(\beta_{i-2}, \beta_{i-3}) \neq (b, b^{-1})$ and so we obtain the expressions for $q$ and $k$ by applying Lemma 5.3 to $uv[i-2,i-2]$ (note that $uv[i-2,i-2] \in \mathbf{BS}_{1,2}$ because $uv[i-1,i-1] \in \mathbf{BS}_{1,2}$). This proves the lemma. □

Remark 5.8. *Observe that we can easily calculate similar expressions as in Lemma 5.7 (with* $\vec{\beta} \in \{b, b^{-1}\}^{\leq 3}$*) and check if* $uv[i,i] \in \mathbf{BS}_{1,2}$ *for* $i \leq 3$ *applying Lemma 5.3 at most three times (as for* $i \geq 4$*).*

**5.2. The algorithm.** A power circuit representation of $u \in \mathbf{G}_{1,2}$ written as in (5.6) consists of the sequence $\mathcal{B} = (\beta_h, \ldots, \beta_1)$ and a power circuit $(\Pi, \delta_\Pi)$ with markings $U_i, E_i, M_i$ for $i \in [0\,..\,h]$ such that $(U_i, E_i)$ is a power circuit representation of $r_i$ (see Definition 4.25) and $m_i = \varepsilon(M_i)$.

Lemma 5.9. *The following problem is in* $\mathsf{DepParaTC}^0$ *parametrized by* $\max_i \operatorname{depth}(\Pi_i)$*:*

Input: *Britton-reduced power circuit representations of* $u, v \in \mathbf{G}_{1,2}$ *over power circuits* $\Pi_1$, $\Pi_2$.

Output: *A Britton-reduced power circuit representation of* $w \in \mathbf{G}_{1,2}$ *over a power circuit* $\Pi'$ *such that* $w =_{\mathbf{G}_{1,2}} uv$ *and* $\operatorname{depth}(\Pi') = \max_i \operatorname{depth}(\Pi_i) + \mathcal{O}(1)$ *and* $|\Pi'| \in \mathcal{O}(|\Pi_1| + |\Pi_2|)$.

Proof. Let $\Pi$ be the disjoint union of $\Pi_1$ and $\Pi_2$. We need to find the maximal $i$ such that $uv[i,i] \in \mathbf{BS}_{1,2}$. This can be done as follows: By Lemma 4.26 one can evaluate the expressions $\theta_{\vec{\beta}}(V_i)$ and $\xi_{\vec{\beta}}(V_i)$ of Lemma 5.7 (and Remark 5.8) and test the conditions $\theta_{\vec{\beta}}(V_i) \in \mathbb{Z}$ and $\xi_{\vec{\beta}}(V_i) = 0$ in $\mathsf{DepParaTC}^0$. For every $i$ this can be done independently in parallel giving us Boolean values indicating whether $uv[i-1,i-1] \in \mathbf{BS}_{1,2}$ implies $uv[i,i] \in \mathbf{BS}_{1,2}$. Now, we have to find only the maximal $i_0$ such that for all $j \leq i_0$ this implication is true. Since $uv[0,0] = 1 \in \mathbf{BS}_{1,2}$, it follows inductively that $uv[i,i] \in \mathbf{BS}_{1,2}$ for all $i \leq i_0$. Moreover, as the implication $uv[i_0,i_0] \in \mathbf{BS}_{1,2} \implies uv[i_0+1,i_0+1] \in \mathbf{BS}_{1,2}$ fails, we have $uv[j,j] \notin \mathbf{BS}_{1,2}$ for $j \geq i_0 + 1$.

Now, using the expressions $\varphi_{\vec{\beta}}, \psi_{\vec{\beta}}$ from Lemma 5.7 (and Remark 5.8) we compute $(q,k) = (\varphi_{\vec{\beta}}(V_{i_0}), \psi_{\vec{\beta}}(V_{i_0})) =_{\mathbf{G}_{1,2}} uv[i_0,i_0]$ in $\mathsf{DepParaTC}^0$ by Lemma 4.26. Again, using Lemma 4.26, we can compute in $\mathsf{DepParaTC}^0$ $(s,m) = (r_{i_0}, m_{i_0})(q,k)(s_{i_0}, n_{i_0})$ as a power circuit representation over a power circuit $(\Pi', \delta_{\Pi'})$ with $(\Pi, \delta_\Pi) \leq (\Pi', \delta_{\Pi'})$, $|\Pi'| \in \mathcal{O}(\Pi)$ and $\operatorname{depth}(\Pi') \in \operatorname{depth}(\Pi) + \mathcal{O}(1)$. Now, the output is

$$(r_h, m_h)\beta_h \cdots (r_{i_0+1}, m_{i_0+1})\beta_{i_0+1}\ (s,m)\ \tilde{\beta}_{i_0+1}(s_{i_0+1}, n_{i_0+1}) \cdots \tilde{\beta}_\ell(s_\ell, n_\ell).$$

□

Before showing Theorem B, we prove the following slightly more general result. Recall that $\Sigma = \{1, a, a^{-1}, t, t^{-1}, b, b^{-1}\}$.

Theorem 5.10. *The following problem is in* $\mathsf{TC}^2$*:*

Input: *A word* $w \in \Sigma^*$.

Output: *A power circuit representation for a Britton-reduced word* $w_{red} \in \Delta^*$ *such that* $w =_{\mathbf{G}_{1,2}} w_{red}$ *and the underlying power circuit has depth* $\mathcal{O}(\log |w|)$.

Proof. Let $w = w_1 \cdots w_n$ with $w_j \in \Sigma$ be some input. Since we can pad with the letter 1, we can assume $n = 2^m$ for $m \in \mathbb{N}$. The idea for the proof is simple: First, we transform each letter $w_j$ into a power circuit representation. After that, the first layer computes the Britton reduction of two-letter words using Lemma 5.9, the next layer takes always two of these Britton-reduced words and joins them to a new Britton-reduced word and so on. After $m = \log n$ layers we have obtained a single Britton-reduced word. By the bound in Lemma 5.9, the size of the resulting power circuits stays polynomial in $n$ and their depth in $\mathcal{O}(\log n)$. In particular, each application of Lemma 5.9 is in $\mathsf{TC}^1$ and, hence, the whole computation is in $\mathsf{TC}^2$.

Let us detail this high-level description a bit further: For $j \in [1\,..\,n]$ we set $w_j = w_j^{(1)}$. Now for each word $w_j^{(1)}$ we construct its power circuit representation as follows: Let $(\Pi_j^{(1)}, \delta_j^{(1)})$ be a power circuit such that $|\Pi_j^{(1)}| = 1$ for $j \in [1\,..\,n]$. We define markings $U_i$, $E_i$ and $M_i$ as follows: If $w_j^{(1)} = a^\alpha$ for $\alpha \in \{-1, 1\}$ then $\varepsilon(U_i) = \alpha$ and $\varepsilon(E_i) = \varepsilon(M_i) = 0$. If $w_j^{(1)} = t^\alpha$, then $\varepsilon(U_i) = \varepsilon(E_i) = 0$ and $\varepsilon(M_i) = \alpha$. If $w_j^{(1)} = \beta$ with $\beta \in \{b, b^{-1}\}$, then all markings evaluate to 0 and we set $\mathcal{B}_j^{(1)} = (\beta)$ (in all other cases we define $\mathcal{B}_j^{(1)}$ to be the empty sequence). If $w_j^{(1)} = 1$, then all markings evaluate to 0 and $\mathcal{B}_j^{(1)}$ is the empty sequence.

Now let $k \in [2\,..\,m+1]$ and $j \in [1\,..\,\frac{n}{2^{k-1}}]$ and assume that the words $w_{2j-1}^{(k-1)}$ and $w_{2j}^{(k-1)}$ are Britton-reduced with power circuit representations over $(\Pi_{2j-1}^{(k-1)}, \delta_{2j-1}^{(k-1)})$ and $(\Pi_{2j}^{(k-1)}, \delta_{2j}^{(k-1)})$, respectively. By Lemma 5.9 we can construct a power circuit representation for a Britton-reduced word $w_j^{(k)}$ over a power circuit $(\Pi_j^{(k)}, \delta_j^{(k)})$ such that $w_j^{(k)} =_{\mathbf{G}_{1,2}} w_{2j-1}^{(k-1)} w_{2j}^{(k-1)}$ and

$$|\Pi_j^{(k)}| \leq c_s \cdot \left(|\Pi_{2j-1}^{(k-1)}| + |\Pi_{2j}^{(k-1)}|\right) \text{ and}$$
$$\operatorname{depth}(\Pi_j^{(k)}) \leq c_d + \max(\operatorname{depth}(\Pi_{2j-1}^{(k-1)}), \operatorname{depth}(\Pi_{2j}^{(k-1)}))$$

for constants $c_s$ and $c_d$. In order to bound the size and depth of these power circuits inductively, define $\Pi^{(k)}$ to be the disjoint union

of the $\Pi_j^{(k)}$ for $j \in [1 \,..\, \frac{n}{2^{k-1}}]$. It follows that

$$|\Pi^{(k)}| \leq c_s \cdot |\Pi^{(k-1)}| \qquad \text{and} \qquad \operatorname{depth}(\Pi^{(k)}) \leq \operatorname{depth}(\Pi^{(k-1)}) + c_d.$$

Let $\nu \in \mathbb{N}$ with $c_s \leq 2^\nu$. With $\left|\Pi^{(1)}\right| = n$ and $\operatorname{depth}(\Pi^{(1)}) = 1$ we obtain that

$$\begin{aligned} \operatorname{depth}(\Pi^{(k)}) &\leq \operatorname{depth}(\Pi^{(1)}) + (k-1) \cdot c_d \\ &\leq 1 + m \cdot c_d = 1 + \log(n) \cdot c_d, \end{aligned} \tag{5.11}$$

$$|\Pi^{(k)}| \leq c_s^{k-1} \cdot |\Pi^{(1)}| \leq c_s^m \cdot n \leq (2^\nu)^{\log(n)} \cdot n = n^{\nu+1}.$$

Therefore, the size of each $\Pi^{(k)}$ is polynomial in the input size $n$ and its depth is logarithmic in $n$. In particular, the same applies to the power circuits $\Pi_j^{(k)}$. Therefore, by Lemma 2.4, Lemma 5.9 yields a $\mathsf{TC}^1$ circuit for computing $w_j^{(k)}$ from $w_{2j-1}^{(k-1)}$ and $w_{2j}^{(k-1)}$. For each $k$, all the power circuit representations of the $w_j^{(k)}$ for $j \in [1 \,..\, \frac{n}{2^{k-1}}]$ can be computed in parallel with the bound on the depth given by (5.11). Since we have $\mathcal{O}(\log n)$ of these stages, the overall complexity is $\mathsf{TC}^2$. □

Proof (of Theorem B). In order to decide whether $w =_{\mathbf{G}_{1,2}} 1$, we first compute its Britton reduction $\hat{w}$ using Theorem 5.10. If $\hat{w}$ still contains some $b$ or $b^{-1}$, by Britton's Lemma, we know that $w \neq_{\mathbf{G}_{1,2}} 1$. Otherwise, $\hat{w} = (r, m) \in \mathbb{Z}[1/2] \rtimes \mathbb{Z}$ where $r, m$ are given as there power circuit representations over a power circuit $\Pi$ of depth $\mathcal{O}(\log |w|)$. According to Lemma 4.26, we can check in $\mathsf{TC}^1$ whether $r = m = 0$. □

**The compressed word problem.** The *compressed word problem* of a group is similar to the ordinary word problem. However, the input element is not given directly but as a straight-line program. A *straight-line program* is a context-free grammar which generates exactly one word. The compressed word problem for a group $G$ with presentation $\eta\colon \Sigma^* \to G$ is as follows:

Input: A straight-line program generating a word $w \in \Sigma^*$
Question: Is $w =_G 1$?

Corollary 5.12. *The compressed word problem of the Baumslag group is in* PSPACE.

Proof. It is an easy exercise that all the circuit families we described are, indeed, LOGSPACE-uniform. In particular, the word problem of $\mathbf{G}_{1,2}$ is in LOGSPACE-uniform $\mathsf{TC}^2$. Since LOGSPACE-uniform $\mathsf{TC}^2$ is contained in $\mathsf{DSPACE}(\log^3 n)$, we can apply Lemma 7.4 of Bartholdi *et al.* (2023) in order to obtain the corollary. □

## 6. Hardness of comparison in power circuits

The main result of this section is to show how Boolean circuits can be simulated by power circuits. This leads to P-completeness of the comparison problem for power circuits (Theorem C). In this section we consider functions computable in DLOGTIME-uniform $\mathsf{AC}^0$. The reader unfamiliar with the precise definitions might simply think of LOGSPACE-computable. We start by introducing some normalization steps for Boolean circuits.

For a Boolean circuit $\mathcal{C}$ with input gates $x_1, \ldots, x_n$ and some $\vec{a} \in \{0,1\}^n$, we write $\mathrm{eval}_{\vec{a}}(\mathcal{C})$ for the evaluation of the output gate of $\mathcal{C}$ when assigning $\vec{a}$ to the inputs.

**Elimination of And-gates.** By de Morgan's rule we can simulate each And-gate by a circuit of depth 3 using an Or-gate and Not-gates. So for each AC-circuit $\mathcal{C}$ of depth $D$ there is an equivalent Boolean circuit $\mathcal{C}'$ of depth at most $3D$ using only Or and Not-gates such that $\mathrm{eval}_{\vec{a}}(\mathcal{C}) = \mathrm{eval}_{\vec{a}}(\mathcal{C}')$ for all $\vec{a} \in \{0,1\}$. Moreover, the circuit $\mathcal{C}'$ clearly can be computed in DLOGTIME-uniform $\mathsf{AC}^0$.

**Layered circuits.** A circuit is called layered if we can assign a level number to each gate such that input gates are on level 0 and gates on level $k$ only receive inputs from level $k-1$.

Given an arbitrary AC-circuit $\mathcal{C}$ of depth $D$, we can construct a layered AC-circuit $\mathcal{C}'$ of depth $D$ as follows: We first make $D+1$ copies of all gates of $\mathcal{C}$ numbered from 0 to $D$. The input gates of $\mathcal{C}'$ are the input gates in copy 0. The other gates in copy 0 (of type And, Or, Not) get the constant 0 as input (indeed, their input

does not have any effect). Then we introduce wires between the gates as in the original circuit, but only between copy $i$ and copy $i+1$. Moreover, for $k \geq 1$ we replace an input gate in copy $k$ by a fan-in one OR-gate which receives its input from the corresponding input gate in copy $k-1$. The output gate of $\mathcal{C}'$ is the output gate in copy $D$. So, we obtain a layered AC-circuit of depth $D$ and size $(D+1)\cdot|\mathcal{C}|$. Because the paths that connect input gates with output gates are the same in both circuits, the new circuit evaluates to 1 if and only if this is the case for $\mathcal{C}$.

Notice that we also can perform this construction if $D$ is not the exact depth but only an upper bound. Moreover, if $D$ is given in the input, the construction can be computed in DLOGTIME-uniform $\mathsf{AC}^0$. Also note that if $\mathcal{C}$ uses only OR and NOT-gates, then also the layered circuit will only use those gates.

THEOREM 6.1. *Let $\mathcal{C}$ be a layered* AC*-circuit made of unbounded fan-in* OR*-gates and* NOT*-gates of size $L$ and depth $D$ and input gates $x_1, \ldots, x_n$. There exists a power circuit $(\Gamma, \delta)$ with special vertices $V_1, \ldots, V_n$ and $\top$, $A$, and $B$ satisfying the following properties:*

*For $\vec{a} \in \{0,1\}^n$ we define a graph $(\Gamma, \delta_{\vec{a}})$ where $\delta_{\vec{a}}(V_i, \top) = a_i$ and on all other nodes $\delta_{\vec{a}}$ agrees with $\delta$. Then for all $\vec{a} \in \{0,1\}^n$ we have*

- *$(\Gamma, \delta_{\vec{a}})$ is a power circuit,*
- *$\mathrm{depth}(\Gamma, \delta_{\vec{a}}) = 2D + \log^*(L) + 4$ and $|\Gamma| \leq 3(L+D) + \log^*(L) + 6$,*
- *$P \neq Q \implies \varepsilon_{\vec{a}}(P) \neq \varepsilon_{\vec{a}}(Q)$ for all nodes $P, Q \in \Gamma$,*
- *$\varepsilon_{\vec{a}}(A) \leq \varepsilon_{\vec{a}}(B)$ if and only if $\mathrm{eval}_{\vec{a}}(\mathcal{C}) = 1$,*
- *each (successor) marking in $(\Gamma, \delta_{\vec{a}})$ is compact,*

*where $\varepsilon_{\vec{a}}$ denotes the evaluation in $(\Gamma, \delta_{\vec{a}})$. Moreover, given $\mathcal{C}$, the power circuit $(\Gamma, \delta)$ can be computed in* LOGSPACE.

Intuitively, Theorem 6.1 states that the family of power circuits $((\Gamma, \delta_{\vec{a}}))_{\vec{a} \in \{0,1\}^n}$ simulates the circuit $C$.

Corollary 6.2. *Let $k \geq 1$. The following problem is in $\mathsf{TC}^k$ and it is hard for $\mathsf{AC}^k$ under $\mathsf{AC}^0$-Turing reductions:*

Input: *A power circuit $(\Pi, \delta_\Pi)$ with $\operatorname{depth}(\Pi) \leq \log^k |\Pi|$ and nodes $A, B \in \Pi$ such that for all $P \in \Pi$ the marking $\Lambda_P$ is compact and for all $P \neq Q$ we have $\varepsilon(P) \neq \varepsilon(Q)$.*

Question: *Is $\varepsilon(A) \leq \varepsilon(B)$?*

Corollary 6.2 shows that Theorem 4.12 is almost optimal—only the space between $\mathsf{AC}^k$ and $\mathsf{TC}^k$ is possibly for the "true" complexity of comparison in $\log^k$-depth power circuits.

Proof. Let $\mathrm{PCComp}_k$ denote the problem from Corollary 6.2. Membership of $\mathrm{PCComp}_k$ in $\mathsf{TC}^k$ is by Corollary 4.22. Like in the proof of Theorem 4.22 in Vollmer (1999), we see that $\mathsf{AC}^k = \mathsf{AC}^0(\mathsf{AC}^k)$ where for a fixed circuit family the oracle gates for $\mathsf{AC}^k$ can be assumed to come from a fixed language $L$ in $\mathsf{AC}^k$ with layered circuits $(C_n)_{n\in\mathbb{N}}$ of depth $\varepsilon \log^k n$ for some small enough $\varepsilon > 0$.

Thus, starting with an $\mathsf{AC}^0(L)$ circuit, we have to construct an $\mathsf{AC}^0(\mathrm{PCComp}_k)$ circuit computing the same function. In order to do so, we proceed as follows: When choosing $\varepsilon$ small enough, by Theorem 6.1, for each $n$ there is a power circuit $(\Gamma_n, \delta_n)$ of depth $\log^k n$ together with the nodes $A$ and $B$ "simulating" the circuit $C_n$ as described in Theorem 6.1. Therefore, we can replace each oracle gate $g$ for $L$ with fan-in $n$ by an oracle gate for $\mathrm{PCComp}_k$ where the actual input power circuit $(\Gamma_n, \delta_n)$ and the nodes $A$ and $B$ are hardwired (as bit-strings) and only $\delta(V_i, \top) \in \{0, 1\}$ is set to the $i$-th input of the gate $g$. By Theorem 6.1, the $\mathrm{PCComp}_k$ oracle gate with this input evaluates to 1 if and only if the oracle gate for $L$ evaluates to 1. □

Note that it is an easy exercise to prove a $\mathsf{LOGSPACE}$-uniform variant of Corollary 6.2 using the statement on $\mathsf{LOGSPACE}$-computability in Theorem 6.1.

Proof (of Theorem 6.1). We start with a layered $\mathsf{AC}$-circuit $\mathcal{C}$ of size $L$ and depth $D$ consisting of *input gates* $x_i$ for $i = 1, \ldots, n$,

NOT-gates, OR-gates (of fan-in at most $L$); one of these gates is marked as *output gate*. Each gate is on a unique level: input gates on level 0, and gates on level $k$ only receive inputs from level $k-1$; the output gate is on level $D$. We assume that the gates are numbered from 1 to $L$ with gates 1 to $n$ being the input gates (i.e., $g_i = x_i$ for $i \in [1\,..\,n]$).

We write $\ell = \log^*(L) + 3$. We assume that $L \geq 3$. Then the following inequalities hold for all $k \geq 0$:

$$\begin{aligned} 2L &\leq 2^L \\ 2^{L+2} &\leq \tau(\ell + k) \\ \tau(k - 1 + \ell) &\leq \tau(k + \ell)/2 - 2^{L+1} \end{aligned} \tag{6.3}$$

**Invariants of $(\Gamma, \delta_{\vec{a}})$:** Starting with the circuit $\mathcal{C}$, we design a power circuit $(\Gamma, \delta)$ with designated nodes $V_1, \ldots, V_n$ such that for every gate $g$ on level $k$, there is some node $P_g$ such that for all $\vec{a} \in \{0,1\}^n$ the following inequalities hold:

$$\begin{aligned} \tau(\ell - 1) < \varepsilon_{\vec{a}}(\Lambda_{P_g}) &\leq \tau(2k + 1 + \ell) - L, \\ \varepsilon_{\vec{a}}(\Lambda_{P_g}) &\leq \tau(2k + \ell) - L \qquad \text{if } \operatorname{eval}_{\vec{a}}(g) = 0, \\ \varepsilon_{\vec{a}}(\Lambda_{P_g}) &\geq \tau(2k + \ell) \qquad \text{if } \operatorname{eval}_{\vec{a}}(g) = 1. \end{aligned} \tag{6.4}$$

Recall that $\varepsilon_{\vec{a}}$ here denotes the evaluation in $(\Gamma, \delta_{\vec{a}})$.

Note that, in particular, if $g$ is the output gate (which is on level $D$), then $\varepsilon_{\vec{a}}(P_g) \geq \tau(2D + 1 + \ell)$ if and only if $\operatorname{eval}_{\vec{a}}(g) = 1$.

**The construction of $(\Gamma, \delta)$:**

- We first create a reduced power circuit consisting of nodes $X_0, \ldots, X_L$ such that $\varepsilon_{\vec{a}}(\Lambda_{X_i}) = i$ (i.e., $\varepsilon_{\vec{a}}(X_i) = 2^i$) independently of $\vec{a}$ and the successor markings $\Lambda_{X_i}$ are compact.

- We create nodes $T_0, \ldots, T_{2D+1+\ell}$ such that $\varepsilon_{\vec{a}}(T_0) = 1$ and $\delta(T_i, T_{i-1}) = 1$ for $i \in [1\,..\,2D + 1 + \ell]$ and $\delta(T_i, Q) = 0$ otherwise. Then $\varepsilon_{\vec{a}}(T_i) = \tau(i)$ independently of $\vec{a}$. If there exists a node $X_i$ such that $\tau(j) = \varepsilon_{\vec{a}}(X_i)$, then we set $T_j = X_i$.

- As an abbreviation, we will denote a node $T_{2k+\ell}$ by $R_k$ for $k \in [1 \,..\, D]$. Note that then $\varepsilon_{\vec{a}}(\Lambda_{R_k}) = \tau(2k-1+\ell)$ independently of $\vec{a}$. Moreover, we write $\top$ for $T_\ell$.

- For every $k \in [1 \,..\, D]$ we create a node $S_k$ with $\delta(S_k, R_{k-1}) = 1$ and $\delta(S_k, X_0) = -1$. So $\varepsilon_{\vec{a}}(\Lambda_{S_k}) = \tau(2k-2+\ell) - 1$ independently of $\vec{a}$. Note that $\Lambda_{S_k}$ is a compact marking and $\varepsilon_{\vec{a}}(\Lambda_{S_k}) > \tau(\ell-1)$ because $k \geq 1$ and $\ell \geq 3$.

- For every input gate $x_i$, we create a node $V_i$ with $\delta(V_i, X_i) = \delta(V_i, T_{\ell-1}) = 1$. Notice that $2^L \leq 2^{\tau(\log^* L)} = \tau(\ell-2)$, so $X_i$ and $T_{\ell-1}$ are guaranteed to be distinct nodes and, thus, $\delta$ is well-defined. We write $P_{g_i}$ as an alias for $V_i$. Note that by the definition of $\delta_{\vec{a}}$ in the theorem we have $\delta_{\vec{a}}(V_i, \top) = a_i$.

- For every OR-gate $g_i$ with incoming edges from other gates $u_1, \ldots, u_h$ we create nodes $P_{g_i}, Q_{g_i} \in \Gamma$ and set $\delta(P_{g_i}, Q_{g_i}) = 1$ and $\delta(Q_{g_i}, P_{u_j}) = 1$ for $1 \leq j \leq h$. In addition, we set $\delta(Q_{g_i}, X_i) = \delta(P_{g_i}, X_i) = 1$. The construction is shown in Figure 6.1.

- For every NOT-gate $g_i$ on level $k$ with incoming edge from gate $u$, we create a node $P_{g_i} \in \Gamma$ and set $\delta(P_{g_i}, R_k) = \delta(P_{g_i}, S_k) = 1$, and $\delta(P_{g_i}, P_u) = -1$. In addition, we set $\delta(P_{g_i}, X_i) = 1$. The construction is depicted in Figure 6.2.

Finally, we set $A = T_{2D+1+\ell}$ and $B = P_g$ where $g$ is the output gate. Thus, once we have proved (6.4), we know that $\varepsilon_{\vec{a}}(A) \leq \varepsilon_{\vec{a}}(B)$ if and only if $\mathrm{eval}_{\vec{a}}(\mathcal{C}) = 1$. The compactness of all the markings will be shown at the end of the proof.

**Size and depth bounds:** Observe that for every gate $g_i$ of $\mathcal{C}$ we introduce at most two nodes $P_{g_i}$ and $Q_{g_i}$ plus the node $X_i$ in $\Gamma$. So, by adding the number of nodes $T_i$ and $S_i$ plus $X_0$, we obtain that $|\Gamma| \leq 3(L+D) + \ell + 3$.

In the following we define the depth of a node as the length of a longest path starting from it (i.e., the depth of $\Gamma$ is the maximal depth of its nodes). Each node $X_i$ for $i \in [0 \,..\, L]$ has depth at most $\ell$ (see also Example 2.8). Each of the nodes $T_k$ has depth $k$, so here

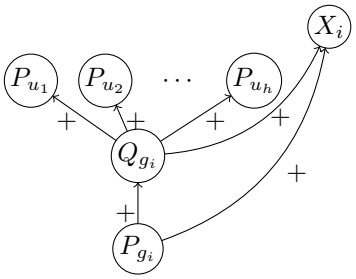



Figure 6.1: Power circuit for OR-gates.

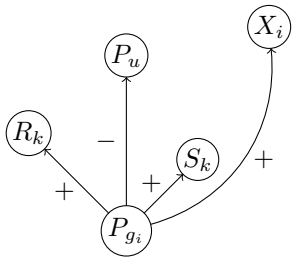



Figure 6.2: Power circuit for NOT-gates.

we obtain nodes of depth at most $2D + \ell + 1$. The node $S_k$ only points to the node $R_{k-1}$ and to $X_0$, so it also has depth $2k - 1 + \ell$ with $k \leq D$.

By induction we see that if $g$ is a gate on level $k$, then the depth of $P_g$ is at most $2k+\ell+1$ Therefore, for each $\vec{a} \in \{0,1\}^n$ the depth of $(\Gamma, \delta_{\vec{a}})$ is exactly $2D + \ell + 1$.

**Complexity of the construction:** The whole construction can be done in LOGSPACE: We can compute the level of each gate and the depth of the circuit in LOGSPACE just by following any path from the gate to the input gates. Since the circuit is layered, this always gives the same result.

The actual construction of $(\Gamma, \delta)$ is straightforward: We start by creating only the nodes, without the edges. First create the nodes $X_i$ for $i \in [0\,..\,L]$. Then we add the nodes $T_0, \ldots, T_{2D+1+\ell}$, and $S_1, \ldots, S_D$. Here we need to be careful and identify each node $T_i$ with $X_{\tau(i-1)}$ if it exists. Since $\log^*$ can be computed in LOGSPACE, both $\ell$ and this identification can be computed in LOGSPACE. Now it remains to create the nodes $P_g$ and (only for OR-gates) $Q_g$ corresponding to the gates, which also clearly can be done in LOGSPACE. For every node the outgoing edges can be determined in a straightforward way from the definition.

**Proof of (6.4) (correctness):** Let $\vec{a} \in \{0,1\}^n$ be some input to $\mathcal{C}$. Let us show (6.4) for every power circuit $(\Gamma, \delta_{\vec{a}})$ by induction starting from the input gates (notice that (6.4) shows that $(\Gamma, \delta_{\vec{a}})$

is a power circuit, indeed). Let $g_i = x_i$ be an input gate and first assume that $a_i = 0$. Then using (6.3) we obtain

$$\varepsilon_{\vec{a}}(\Lambda_{V_i}) = \varepsilon_{\vec{a}}(T_{\ell-1}) + \varepsilon_{\vec{a}}(X_i) = \tau(\ell-1) + 2^i < \tau(\ell) - L$$

and $\varepsilon_{\vec{a}}(\Lambda_{V_i}) > \tau(\ell-1)$. Now we assume that $a_i = 1$. Then again by (6.3) we have

$$\varepsilon_{\vec{a}}(\Lambda_{V_i}) = \varepsilon_{\vec{a}}(T_{\ell-1}) + \varepsilon_{\vec{a}}(T_\ell) + \varepsilon_{\vec{a}}(X_i) = \tau(\ell-1) + \tau(\ell) + 2^i > \tau(\ell)$$

and

$$\varepsilon_{\vec{a}}(\Lambda_{V_i}) \le 2 \cdot \tau(\ell) + 2^L \le \tau(\ell+1) - L.$$

This shows (6.4) for nodes $V_i = P_{g_i}$ corresponding to input gates.

Now let $g_i$ be some OR-gate on level $k \ge 1$ with inputs from gates $u_1, \ldots, u_h$. First assume that $\mathrm{eval}_{\vec{a}}(u_j) = 0$ for all $j \in [1\,..\,h]$. By induction, $\varepsilon_{\vec{a}}(\Lambda_{P_{u_j}}) > \tau(\ell-1)$ for all $j \in [1\,..\,h]$; hence, also $\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) > \tau(\ell-1)$. Moreover, by induction, we have $\varepsilon_{\vec{a}}(\Lambda_{P_{u_j}}) \le \tau(2k-2+\ell) - L$ for all $j \in [1\,..\,h]$ meaning that

$$\begin{aligned}\varepsilon_{\vec{a}}(P_{u_j}) = 2^{\varepsilon_{\vec{a}}(\Lambda_{P_{u_j}})} &\le 2^{\tau(2k-2+\ell)-L} = \tau(2k-1+\ell)/2^L \\ &\le \tau(2k-1+\ell)/(2L).\end{aligned}$$

The last inequality is by (6.3). We know that $h \le L$, so

$$\begin{aligned}\varepsilon_{\vec{a}}(\Lambda_{Q_{g_i}}) &= \left(\sum_{j=1}^{h} \varepsilon_{\vec{a}}(P_{u_j})\right) + \varepsilon_{\vec{a}}(X_i) \\ &\le \left(\sum_{j=1}^{h} \frac{1}{2\,L} \cdot \tau(2k-1+\ell)\right) + 2^i \\ &\le \frac{1}{2} \cdot \tau(2k-1+\ell) + 2^L \\ &\le \tau(2k-1+\ell) - L \qquad \text{(by (6.3))}\end{aligned}$$

and

$$\begin{aligned}\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) &= 2^{\varepsilon_{\vec{a}}(\Lambda_{Q_{g_i}})} + \varepsilon_{\vec{a}}(X_i) \le 2^{\tau(2k-1+\ell)-L} + 2^i \\ &= \tau(2k+\ell)/2^L + 2^i \le \tau(2k+\ell) - L. \qquad \text{(by (6.3))}\end{aligned}$$

Now assume that $\mathrm{eval}_{\vec{a}}(u_j) = 1$ for some $j \in [1\,..\,h]$. The same argument as above also shows the bound $\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) \leq \tau(2k+1+\ell)-L$ in this case. Furthermore, by induction, we have $\varepsilon_{\vec{a}}(\Lambda_{P_{u_j}}) \geq \tau(2k-2+\ell)$. Hence,

$$\varepsilon_{\vec{a}}(\Lambda_{Q_{g_i}}) = \sum_{j=1}^{h} \varepsilon_{\vec{a}}(P_{u_j}) + \varepsilon_{\vec{a}}(X_i) > 2^{\tau(2k-2+\ell)} = \tau(2k-1+\ell)$$

and

$$\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) = \varepsilon_{\vec{a}}(Q_{g_i}) + \varepsilon_{\vec{a}}(X_i) \geq 2^{\tau(2k-1+\ell)} = \tau(2k+\ell).$$

Next, let $g_i$ be a NOT-gate on level $k \geq 1$ with an incoming edge from gate $u$. Assume that $\mathrm{eval}_{\vec{a}}(u) = 0$. Then, $\varepsilon_{\vec{a}}(\Lambda_{P_u}) \leq \tau(2k-2+\ell) - L$. So $\varepsilon_{\vec{a}}(P_u) \leq \tau(2k-1+\ell)/2^L$ and

$$\begin{aligned}
\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) &= \varepsilon_{\vec{a}}(R_k) + \varepsilon_{\vec{a}}(S_k) - \varepsilon_{\vec{a}}(P_u) + \varepsilon_{\vec{a}}(X_i) \\
&\geq \tau(2k+\ell) + \tau(2k-1+\ell)/2 - \tau(2k-1+\ell)/2^L \\
&\geq \tau(2k+\ell).
\end{aligned}$$

In addition, we have

$$\begin{aligned}
\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) &= \varepsilon_{\vec{a}}(R_k) + \varepsilon_{\vec{a}}(S_k) - \varepsilon_{\vec{a}}(P_u) + \varepsilon_{\vec{a}}(X_i) \\
&\leq \varepsilon_{\vec{a}}(R_k) + \varepsilon_{\vec{a}}(S_k) + \varepsilon_{\vec{a}}(X_i) \\
&\leq \tau(2k+\ell) + \tau(2k-1+\ell)/2 + 2^L \\
&\leq \tau(2k+1+\ell) - L. && \text{(by (6.3))}
\end{aligned}$$

Now, assume that $\mathrm{eval}_{\vec{a}}(u) = 1$. Then, by induction, we have $\varepsilon_{\vec{a}}(\Lambda_{P_u}) \geq \tau(2k-2+\ell)$. Hence,

$$\begin{aligned}
\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) &= \varepsilon_{\vec{a}}(R_k) + \varepsilon_{\vec{a}}(S_k) - \varepsilon_{\vec{a}}(P_u) + \varepsilon_{\vec{a}}(X_i) \\
&\leq \tau(2k+\ell) + \tau(2k-1+\ell)/2 - \tau(2k-1+\ell) + 2^L \\
&\leq \tau(2k+\ell) - \frac{1}{2} \cdot \tau(2k-1+\ell) + 2^L \\
&\leq \tau(2k+\ell) - L. && \text{(by (6.3))}
\end{aligned}$$

Finally, we know that $\varepsilon_{\vec{a}}(\Lambda_{P_u}) \leq \tau(2k-1+\ell) - L$. So,

$$\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) = \varepsilon_{\vec{a}}(R_k) + \varepsilon_{\vec{a}}(S_k) - \varepsilon_{\vec{a}}(P_u) + \varepsilon_{\vec{a}}(X_i)$$

$$\geq \tau(2k+\ell) + \tau(2k-1+\ell)/2 - \tau(2k+\ell)/2^L$$
$$\geq \frac{2^L-1}{2^L}\,\tau(2k+\ell) > \tau(\ell-1).$$

**Uniqueness of evaluations.** It remains to show that no two nodes have the same evaluation and that each successor marking is compact. Let $P, Q \in \Gamma$. Now we have to show that $\varepsilon_{\vec{a}}(P) \neq \varepsilon_{\vec{a}}(Q)$ if $P \neq Q$. First, we set $\mathcal{X} = \{X_i, T_j \mid i \in [0\,..\,L], j \in [0\,..\,\ell]\}$. By construction, $\varepsilon_{\vec{a}}(P) \neq \varepsilon_{\vec{a}}(Q)$ if $P \neq Q$ is clear for $P, Q \in \mathcal{X}$ (independently of $\vec{a} \in \{0,1\}^n$). We further know that $\varepsilon_{\vec{a}}(P) > \tau(\ell)$ for all nodes $P \in \Gamma\backslash\mathcal{X}$ and $\varepsilon_{\vec{a}}(P) \leq \tau(\ell)$ for $P \in \mathcal{X}$; so in particular, $\varepsilon_{\vec{a}}(P) \neq \varepsilon_{\vec{a}}(Q)$ for all nodes $P \in \Gamma\backslash\mathcal{X}$ and $Q \in \mathcal{X}$.

For $P \in \Gamma \setminus \mathcal{X}$ we can conclude that $\tau(\ell)$ divides $\varepsilon_{\vec{a}}(P)$ and so $\varepsilon_{\vec{a}}(P) \equiv 0 \mod \tau(\ell)$ (note that $\varepsilon_{\vec{a}}(P)$ is a power of two). Now let $u_i$ be an arbitrary gate and $v_j$ be an arbitrary OR-gate. Considering the successor markings of the nodes $P_{u_i}$ and $Q_{v_j}$, we obtain the following:

$$\begin{aligned}
\varepsilon_{\vec{a}}(\Lambda_{P_{u_i}}) &\equiv 2^i \mod \tau(\ell),\\
\varepsilon_{\vec{a}}(\Lambda_{Q_{v_j}}) &\equiv 2^j \mod \tau(\ell),\\
\varepsilon_{\vec{a}}(\Lambda_{T_r}) &\equiv 0 \mod \tau(\ell) \qquad \text{for } r \geq \ell+1,\\
\varepsilon_{\vec{a}}(\Lambda_{S_k}) &\equiv -1 \mod \tau(\ell).
\end{aligned} \tag{6.5}$$

By the choice of $\ell$ we know that $2^i \not\equiv \alpha \mod \tau(\ell)$ for $\alpha \in \{-1, 0\}$ and $i \in [1\,..\,L]$, and $2^i \not\equiv 2^j \mod \tau(\ell)$ for $i, j \in [1\,..\,L]$ and $i \neq j$. Since all nodes of $\Gamma$ are of the above form (remember that for an input gate $g_i$ we wrote $V_i = P_{g_i}$ and also $R_i$, $\top$, $A$ and $B$ were just aliases for other nodes), $\varepsilon_{\vec{a}}(P) \neq \varepsilon_{\vec{a}}(Q)$ whenever $P \neq Q$. Note that these observations hold independently of $\vec{a} \in \{0,1\}^n$.

The successor markings of the nodes $X_i$, $T_i$, $S_i$ are compact by construction. By (6.5) we have $|\varepsilon_{\vec{a}}(\Lambda_P) - \varepsilon_{\vec{a}}(\Lambda_Q)| \geq 2$ for all nodes $P, Q \in \Gamma\backslash(\mathcal{X} \cup \{S_1, \ldots, S_D\})$ with $P \neq Q$. Moreover, $|\varepsilon_{\vec{a}}(\Lambda_P) - \varepsilon_{\vec{a}}(\Lambda_{X_i})| \geq 2$ for all nodes $P \in \Gamma \setminus \mathcal{X}$ and all $i \in [1\,..\,L]$. This shows that $\Lambda_{P_i}$ and $\Lambda_{Q_i}$ are compact for OR-gates $g_i$. In order to see that the successor markings of nodes corresponding to NOT-gates are compact, observe that also $|\varepsilon_{\vec{a}}(\Lambda_{R_k}) - \varepsilon_{\vec{a}}(\Lambda_{S_k})| \geq 2$ and $\left|\varepsilon_{\vec{a}}(\Lambda_{P_{g_i}}) - \varepsilon_{\vec{a}}(\Lambda_{S_k})\right| \geq 2$ for all $k \in [1\,..\,D]$ and $i \in [1\,..\,L]$.

Finally, successor markings of nodes corresponding to input gates are compact because $\varepsilon_{\vec{a}}(\Lambda_{X_i}) + 2 \leq L + 2 < 2^L \leq \tau(\ell - 2) = \varepsilon_{\vec{a}}(\Lambda_{T_{\ell-1}}) < \tau(\ell - 1) - 2 = \varepsilon_{\vec{a}}(\Lambda_{T_\ell}) - 2$. Thus, all successor markings are compact. □

**6.1. P-hardness of power circuit comparison.** Finally, let us apply Theorem 6.1 in order to prove some P-hardness results on comparison in power circuits. Here, we use LOGSPACE-reductions.

COROLLARY 6.6. *The following problem is* P*-complete:*

Input: *A power circuit* $(\Pi, \delta_\Pi)$ *and nodes* $A, B \in \Pi$ *such that for all* $P \in \Pi$ *the marking* $\Lambda_P$ *is compact and for all* $P \neq Q$, $\varepsilon(P) \neq \varepsilon(Q)$.

Question: *Is* $\varepsilon(A) \leq \varepsilon(B)$*?*

Note that a weaker form of this result already has been stated in the second author's dissertation (Weiß 2015).

PROOF. By Proposition 49 of Myasnikov, Ushakov & Won (2012) the problem can be solved in P (this also follows from Corollary 4.22 together with the observation that the circuit family we construct is uniform). P-hardness is by Theorem 6.1 since the circuit value problem (with circuits of unrestricted depth) is P-complete. □

Notice that the only feature the power circuit in Corollary 6.6 lacks for being reduced is the sorting of the nodes. In particular, under the assumption NC ≠ P, it is not possible to sort the nodes of a given power circuit in NC.

REMARK 6.7. *(a) It is an immediate consequence of Corollary 6.6 that the comparison problem of two markings in a power circuit is* P*-complete. This is because for two nodes* $A$ *and* $B$ *in a power circuit* $(\Pi, \delta_\Pi)$ *we have* $\varepsilon(A) \leq \varepsilon(B)$ *if and only if* $\varepsilon(\Lambda_A) \leq \varepsilon(\Lambda_B)$.

*(b) If the input is given as in Corollary 6.6, we can check in* $\mathsf{AC}^0$ *whether* $\varepsilon(A) = \varepsilon(B)$ *because this is the case if and only if* $\Lambda_A(P) = \Lambda_B(P)$ *for all* $P \in \Gamma$ *(see Lemma 3.9). This can be viewed as a hint that also in an arbitrary power circuit testing for equality might be easier than comparing for less than.*

Corollary 6.8. *The following problem is* P*-complete:*

Input: *A dag* $\Gamma$ *such that each node has a successor marking.*
Question: *Is* $\Gamma$ *already a power circuit?*

Proof. The comparison problem for power circuits (see Corollary 6.6) can be reduced to this problem in a straightforward way: As input we have a power circuit $(\Pi, \delta_\Pi)$ with two nodes $A, B \in \Pi$. Then we construct a dag as follows: We take $(\Pi, \delta_\Pi)$ and add a node $R$ with $\delta(R, A) = 1$ and $\delta(R, B) = -1$. Then $\varepsilon(\Lambda_R) \geq 0$ if and only if $\varepsilon(A) \geq \varepsilon(B)$, and so the newly defined dag is a power circuit if and only if $\varepsilon(A) \geq \varepsilon(B)$. □

Corollary 6.9. *The following problem is* P*-complete:*

Input: *A power circuit representation of* $w \in \mathbf{G}_{1,2}$.
Question: *Is* $w \in \mathbf{BS}_{1,2}$*?*

Proof. The problem is in P since the algorithms for the word problem by Diekert *et al.* (2013); Myasnikov *et al.* (2011) also work with power circuit representations as input and they can be used to decide membership in $\mathbf{BS}_{1,2}$ (this is due to Britton's Lemma). Thus, it remains to show the hardness part.

We reduce from the comparison problem for power circuits (Corollary 6.6). So let $(\Pi, \delta_\Pi)$ be a power circuit and let $M$ be a marking on $\Pi$. Now we consider the word $w = b(2^{\varepsilon(M)}, 0)b^{-1} \in \Delta^*$. Then $(\Pi, \delta_\Pi)$ together with the marking $M$ is a power circuit representation of $w$. For $w$ to be in $\mathbf{BS}_{1,2}$ we need the $b$'s to cancel. This happens if and only if $\varepsilon(M) \geq 0$. So, $w \in \mathbf{BS}_{1,2}$ if and only if $\varepsilon(M) \geq 0$. □

As a last result, let us state a straightforward lower bound for the problem of checking two markings for equality. Note that here we even need an arbitrary power circuit as input and cannot restrict it as in Corollary 6.6.

PROPOSITION 6.10. *The following problem is* NL*-hard:*

Input: *Given a power circuit and markings $M, K$.*
Question: *Is $\varepsilon(M) = \varepsilon(K)$?*

PROOF. Reduce the $s$-$t$-connectivity problem for dags to this problem. Given a dag $G = (V, E)$ and vertices $s, t \in V$ make two copies of this graph and add a label $+1$ to every edge. In one copy of $G$ add an additional node $P$ and an edge $(t, P)$. Let $s_1, s_2$ denote the two copies of $s$. Then $\varepsilon(s_1) = \varepsilon(s_2)$ if and only if there is no path from $s$ to $t$ in $G$. □

## 7. Conclusion

We showed that the word problem of the Baumslag group can be solved in $\mathsf{TC}^2$. The proof relies on the fact that all power circuits used during the execution of the algorithm have logarithmic depth. The comparison problem for such power circuits is in $\mathsf{TC}^1$, although for arbitrary power circuits it is P-complete. We conclude with some open problems:

- Is it possible to reduce the complexity of the word problem of the Baumslag group any further—e.g., to find a LOGSPACE algorithm? Notice that in the time this manuscript has been under review, we succeeded to build upon the methods developed in this work and to improve our results by showing that the word problem of the Baumslag group is actually in $\mathsf{TC}^1$, see (Mattes & Weiß 2022). However, it remains an open problem whether it can be solved in LOGSPACE.

- Can we prove some non-trivial lower bounds (the word problem is $\mathsf{NC}^1$-hard as $\mathbf{G}_{1,2}$ contains a non-abelian free group (Robinson 1993))?

- The problem of comparing two markings on a power circuit for equality is NL-hard – is it also P-complete like comparison with less than?

- Is the word problem of the Baumslag group with power circuit representations as input P-complete? (By Corollary 6.9

this holds for the subgroup membership problem for $\mathbf{BS}_{1,2}$ in $\mathbf{G}_{1,2}$. Moreover, as a consequence of Proposition 6.10, this variant of the word problem is $\mathsf{NL}$-hard.)

- By Corollary 6.2 for every $k$ the comparison problem for power circuits of depth $\log^k n$ is in $\mathsf{TC}^k$ and hard for $\mathsf{AC}^k$ under $\mathsf{AC}^0$-Turing reductions. Thus, the question remains whether, indeed, this problem is complete for $\mathsf{TC}^k$ under $\mathsf{AC}^0$-Turing reductions.

## Acknowledgements

A conference version of this work has appeared at MFCS 2021 (Mattes & Weiß 2021). The second author (Armin Weiß) has been funded by DFG projects DI 435/7-1 and WE 6835/1-2.

**Funding**
Open Access funding enabled and organized by Projekt DEAL.

## References

Carme Àlvarez & Birgit Jenner (1993). A Very Hard log-Space Counting Class. *Theor. Comput. Sci.* **107**(1), 3–30. URL https://doi.org/10.1016/0304-3975(93)90252-O.

Sanjeev Arora & Boaz Barak (2009). *Computational Complexity - A Modern Approach*. Cambridge University Press. ISBN 0521424267.

Owen Baker (2020). The conjugacy problem for Higman's group. *Internat. J. Algebra Comput.* **30**(6), 1211–1235. URL https://doi.org/10.1142/S0218196720500393.

Laurent Bartholdi, Michael Figelius, Markus Lohrey & Armin Weiss (2023). Groups with ALOGTIME-Hard Word Problems and PSPACE-Complete Compressed Word Problems. *ACM Trans. Comput. Theory* **14**(3-4). ISSN 1942-3454. URL https://doi.org/10.1145/3569708.

G. Baumslag, A. G. Myasnikov & V. Shpilrain (2002). Open problems in combinatorial group theory. Second Edition. In *Combinatorial and geometric group theory*, volume 296 of *Contemporary Mathematics*, 1–38. American Mathematical Society. URL https://doi.org/10.1090/conm/296/05067.

Gilbert Baumslag (1969). A non-cyclic one-relator group all of whose finite quotients are cyclic. *Journal of the Australian Mathematical Society* **10**(3-4), 497–498. URL https://doi.org/10.1017/S1446788700007783.

W. W. Boone (1959). The Word Problem. *Ann. of Math.* **70**(2), 207–265. URL https://doi.org/10.2307/1970103.

John L. Britton (1963). The word problem. *Ann. of Math.* **77**, 16–32. URL https://doi.org/10.2307/1970200.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein (2009). *Introduction to Algorithms*. The MIT Press, 3rd edition. ISBN 0262033844.

Max Dehn (1911). Ueber unendliche diskontinuierliche Gruppen. *Math. Ann.* **71**, 116–144. URL https://doi.org/10.1007/BF01456932.

Volker Diekert, Jürn Laun & Alexander Ushakov (2013). Efficient algorithms for highly compressed data: The word problem in Higman's group is in P. *International Journal of Algebra and Computation* **22**(8). URL https://doi.org/10.1142/S0218196712400085.

Volker Diekert, Alexei G. Myasnikov & Armin Weiss (2016). Conjugacy in Baumslag's group, generic case complexity, and division in power circuits. *Algorithmica* **74**, 961–988. URL https://doi.org/10.1007/s00453-016-0117-z.

Will Dison, Eduard Einstein & Timothy R. Riley (2018). Taming the hydra: The word problem and extreme integer compression. *Int. J. Algebra Comput.* **28**(7), 1299–1381. URL https://doi.org/10.1142/S0218196718500583.

S. M. Gersten (1991). Isodiametric and isoperimetric inequalities in group extensions. Preprint.

U. Güntzer & M. Paul (1987). Jump interpolation search trees and symmetric binary numbers. *Information Processing Letters* **26**(4), 193–204. URL https://doi.org/10.1016/0020-0190(87)90005-6.

Graham Higman (1951). A finitely generated infinite simple group. *J. London Math. Soc.* **26**, 61–64. URL https://doi.org/10.1112/jlms/s1-26.1.61.

Jonathan Jedwab & Chris Mitchell (1989). Minimum weight modified signed-digit representations and fast exponentiation. *Electronics Letters* **25**, 1171–1172. URL https://doi.org/10.1049/el:19890785.

I. Kapovich, A. G. Miasnikov, P. Schupp & V. Shpilrain (2003). Generic-case complexity, decision problems in group theory and random walks. *Journal of Algebra* **264**, 665–694. URL https://doi.org/10.1016/S0021-8693(03)00167-4.

Jonathan Kausch (2017). *The parallel complexity of certain algorithmic problems in group theory*. Dissertation, Institut für Formale Methoden der Informatik, Universität Stuttgart. URL http://dx.doi.org/10.18419/opus-9152.

Daniel König & Markus Lohrey (2018). Evaluation of Circuits Over Nilpotent and Polycyclic Groups. *Algorithmica* **80**(5), 1459–1492. URL https://doi.org/10.1007/s00453-017-0343-z.

Jürn Laun (2014). Efficient Algorithms for Highly Compressed Data: The Word Problem in Generalized Higman Groups Is in P. *Theory Comput. Syst.* **55**(4), 742–770. URL https://doi.org/10.1007/s00224-013-9509-5.

J. Lehnert & P. Schweitzer (2007). The co-word problem for the Higman-Thompson group is context-free. *Bull. London Math. Soc.* **39**, 235–241. URL http://blms.oxfordjournals.org/content/39/2/235.abstract.

Richard J. Lipton & Yechezkel Zalcstein (1977). Word Problems Solvable in Logspace. *J. ACM* **24**, 522–526. URL https://doi.org/10.1145/322017.322031.

Markus Lohrey (2005). Decidability and complexity in automatic monoids. *International Journal of Foundations of Computer Science* **16**(4), 707–722. URL https://doi.org/10.1007/978-3-540-30550-7_26.

Markus Lohrey (2014). *The Compressed Word Problem for Groups*. Springer Briefs in Mathematics. Springer. URL https://doi.org/10.1007/978-1-4939-0748-9.

Roger Lyndon & Paul Schupp (2001). *Combinatorial Group Theory*. Classics in Mathematics. Springer. ISBN 978-3-540-41158-1. First edition 1977.

Wilhelm Magnus (1932). Das Identitätsproblem für Gruppen mit einer definierenden Relation. *Mathematische Annalen* **106**, 295–307. URL https://doi.org/10.1007/BF01455888.

Wilhelm Magnus, Abraham Karrass & Donald Solitar (2004). *Combinatorial Group Theory*. Dover. ISBN 0486438309.

Caroline Mattes & Armin Weiss (2021). Parallel Algorithms for Power Circuits and the Word Problem of the Baumslag Group. In *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021*, volume 202 of *LIPIcs*, 74:1–74:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL https://doi.org/10.4230/LIPIcs.MFCS.2021.74.

Caroline Mattes & Armin Weiss (2022). Improved Parallel Algorithms for Generalized Baumslag Groups. In *LATIN 2022: Theoretical Informatics*, 658–675. Springer International. URL https://doi.org/10.1007/978-3-031-20624-5_40.

Alexei Miasnikov & Andrey Nikolaev (2020). On parameterized complexity of the word search problem in the Baumslag-Gersten group. In *ISSAC '20: International Symposium on Symbolic and Algebraic Computation, 2020*, 360–363. URL https://doi.org/10.1145/3373207.3404042.

Alexei G. Myasnikov, Alexander Ushakov & Dong-Wook Won (2011). The Word Problem in the Baumslag group with a non-elementary Dehn function is polynomial time decidable. *Journal of Algebra* **345**, 324–342. URL https://doi.org/10.1016/j.jalgebra.2011.07.024.

Alexei G. Myasnikov, Alexander Ushakov & Dong-Wook Won (2012). Power Circuits, exponential Algebra, and Time Complexity. *International Journal of Algebra and Computation* **22**(6), 3–53. URL https://doi.org/10.1142/S0218196712500476.

Alexei G. Myasnikov & Sasha Ushakov (2004–2013). Cryptography And Groups (CRAG). Software Library. URL http://www.stevens.edu/algebraic/downloads.php.

P. S. Novikov (1955). On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov* 1–143. URL https://doi.org/10.2307/2964487. In Russian.

A. N. Platonov (2004). Isoparametric function of the Baumslag-Gersten group. *Vestnik Moskov. Univ. Ser. I Mat. Mekh.* **3**, 12–17. URL https://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=vmumm&paperid=1241&option_lang=eng. Russian. Engl. transl. Moscow Univ. Math. Bull. 59 (3) (2004), 12–17.

George W. Reitwiesner (1960). Binary Arithmetic. volume 1 of *Advances in Computers*, 231–308. Elsevier. URL https://doi.org/10.1016/S0065-2458(08)60610-5.

David Robinson (1993). *Parallel Algorithms for Group Word Problems*. Ph.D. thesis, University of California, San Diego. URL https://dl.acm.org/doi/10.5555/165413.

Mark V. Sapir, Jean-Camille Birget & Eliyahu Rips (2002). Isoperimetric and Isodiametric Functions of Groups. *Ann. Math.* **156**(2), 345–466. URL https://doi.org/10.2307/3597195.

A. L. Semenov (1983). Logical theories of one-place functions on the natural number series. *Izv. Akad. Nauk SSSR Ser. Mat.* **47**(3), 623–658. URL https://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=im&paperid=1415&option_lang=eng.

J. O. Shallit (1993). A Primer on Balanced Binary Representations. Technical report. URL https://cs.uwaterloo.ca/~shallit/Papers/bbr.pdf.

Hans-Ulrich Simon (1979). Word problems for groups and contextfree recognition. In *Proceedings of Fundamentals of Computation Theory (FCT'79), Berlin/Wendisch-Rietz (GDR)*, 417–422. Akademie-Verlag.

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

Heribert Vollmer (1999). *Introduction to Circuit Complexity*. Springer, Berlin. ISBN 3540643109.

Armin Weiss (2015). *On the Complexity of Conjugacy in Amalgamated Products and HNN Extensions*. Dissertation, Institut für Formale Methoden der Informatik, Universität Stuttgart. URL http://dx.doi.org/10.18419/opus-3538.

Armin Weiss (2016). A Logspace Solution to the Word and Conjugacy problem of Generalized Baumslag-Solitar Groups. In *Algebra and Computer Science*, volume 677 of *Contemporary Mathematics*, 185–212. American Mathematical Society. URL https://doi.org/10.1090/conm/677/13628.

Manuscript received 6 December 2021

Caroline Mattes
Universität Stuttgart
Institut für Formale Methoden der Informatik (FMI)
70569 Stuttgart, Germany
caroline.mattes@fmi.uni-stuttgart.de

Armin Weiß
Universität Stuttgart
Institut für Formale Methoden der Informatik (FMI)
70569 Stuttgart, Germany
armin.weiss@fmi.uni-stuttgart.de
https://orcid.org/0000-0002-7645-5867