Parallel Algorithms for Power Circuits and the Word Problem of the Baumslag Group

Caroline Mattes
Institut für Formale Methoden der Informatik (FMI), University of Stuttgart, Germany

Armin Weiß
Institut für Formale Methoden der Informatik (FMI), University of Stuttgart, Germany

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.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Circuit complexity

Keywords and phrases Word problem, Baumslag group, power circuit, parallel complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.74

Related Version *Full Version*: `https://arxiv.org/abs/2102.09921` [28]

Funding *Armin Weiß*: Funded by DFG project DI 435/7-1.

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 as one of the basic algorithmic problems in group theory [8]. Already in the 1950s, Novikov and Boone succeeded to construct finitely presented groups with an undecidable word problem [5, 33]. 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 [22, 38] – hence, in particular, in NC, i.e., decidable by Boolean circuits of polynomial size and polylogarithmic depth.

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 TC^0 (constant depth with threshold gates) [19] and the word problems of Baumslag-Solitar groups and of right-angled Artin groups are AC^0-Turing-reducible to the word problem of a non-abelian free group [42, 18]. Moreover, Thompson's groups are co-context-free [21] and hyperbolic groups have word problem in LOGCFL [23]. All these 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 [36].

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 [26] showed that one-relator groups have a decidable word problem; his algorithm is called the Magnus breakdown procedure (see also [25, 27]). 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, in [3] the question is raised whether they are actually decidable in polynomial time.

In 1969 Gilbert Baumslag defined the group $\mathbf{G}_{1,2} = \langle a, b \mid bab^{-1}a = a^2bab^{-1} \rangle$ as an example of a one-relator group which enjoys certain remarkable properties. It is infinite and non-abelian, but all its finite quotients are cyclic and, thus, it is not residually finite [4]. Moreover, Gersten showed that the Dehn function of $\mathbf{G}_{1,2}$ is non-elementary [15] and Platonov [34] 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 group 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 and Won succeeded to show that the word problem of $\mathbf{G}_{1,2}$ is, indeed, decidable in polynomial time [30]. Their crucial contribution was to introduce so-called *power circuits* in [31] for compressing the x_i in the description above.

Roughly speaking, a *power circuit* is a directed acyclic graph (a dag) where the edges are labelled by ± 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-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 and Ushakov [10, 9] 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 [16]. In [32] these algorithms have been implemented in C++. Subsequently, more applications of power circuits to these groups emerged: in [20] a polynomial time solution to the word problem in generalized Baumslag and Higman groups is given, in [12, 11] the conjugacy problem of the Baumslag group is shown to be strongly generically in P and in [2] the same is done 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 [17]).

Other examples where compression techniques lead to efficient algorithms in group theory can be found e.g. in [13, 14] or [24, Theorems 4.6, 4.8 and 4.9]. Finally, notice that in [29] the word search problem for the Baumslag group has been examined 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 and show that ordinary binary representations can be converted into compact representations by constant depth circuits (i.e., in 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 [31, 10] 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* 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 TC^1 algorithm for computing the Britton reduction of uv if 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* 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* 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 the corresponding logic over natural numbers with addition and 2^x has been shown to be decidable by Semënov [37]. In the full version [28] 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.

Due to space constraints we present only short outlines of the proofs for our main theorems; the full proofs as well as further details can be found in the full version on arXiv [28]. Details of the reduction process also can be found in the appendix.

2 Notation and preliminaries

General notions. We use standard $\mathcal{O}$-notation for functions from $\mathbb{N}$ to non-negative reals $\mathbb{R}^{\geq 0}$, see e.g. [7]. 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. We denote the support of a function $f\colon X \to \mathbb{R}$ by $\sigma(f) = \{x \in X \mid f(x) \neq 0\}$. Furthermore, the interval of integers $\{i, \dots, j\} \subseteq \mathbb{Z}$ is denoted 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 Σ be a set. The set of all words over Σ 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 Γ we write $\mathrm{depth}(\Gamma)$ for its depth, which is the length (number of edges) of a longest path in Γ.

Complexity. We assume the reader to be familiar with the complexity classes $\mathsf{LOGSPACE}$ and P (polynomial time); see e.g. [1] for details. Most of the time, however, we use circuit complexity within NC.

Throughout, we assume that languages L (resp. inputs to functions f) are encoded over the binary alphabet $\{0, 1\}$. A Boolean circuit is a dag where the vertices are either input gates $x_1, \dots, x_n$, or Not, And, or Or gates. There are one or more designated output gates

and there is an order given on the output gates. All gates may have unbounded fan-in (i.e., there is no bound on the number of incoming wires). Let $k \in \mathbb{N}$. A language $L \subseteq \{0,1\}^*$ belongs to AC^k if there exists a family $(C_n)_{n\in\mathbb{N}}$ of Boolean circuits such that $x \in L \cap \{0,1\}^n$ if and only if the (unique) output gate of C_n evaluates to 1 when assigning $x = x_1 \cdots x_n$ to the input gates. Moreover, C_n may contain at most $n^{\mathcal{O}(1)}$ gates and have depth $\mathcal{O}(\log^k n)$. Likewise AC^k-computable functions are defined.

The class 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 [40]. Our algorithms (or circuits) rely on two basic building blocks which can be done in TC^0:

▶ **Example 1.** Iterated addition is the following problem: on input of n binary numbers $A_1, \ldots, A_n$ each having n bits, compute $\sum_{i=1}^n A_i$. This is well-known to be in TC^0 – see e.g. [40, Theorem 1.37] for a proof.

▶ **Example 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 TC^0 whether $k_i < k_j$. Then the problem of sorting the list according to the keys is in TC^0: the desired output is a list $(k_{\pi(1)}, v_{\pi(1)}), \ldots, (k_{\pi(n)}, v_{\pi(n)})$ for some permutation π 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 3. The class NC is contained in P if we consider uniform circuits. Roughly speaking, a circuit family is called *uniform* if the n-input circuit can be computed efficiently from the string 1^n. 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 TC^k, which we call *depth-parametrized* 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)[1] and has depth $\mathcal{O}(D \cdot \log^k n)$. A language L is said to be accepted by this circuit family if for all n and D and all $x \in \{0,1\}^n$ with $\mathrm{par}(x) \leq D$ we have $x \in L$ if and only if $C_{n,D}$ evaluates to 1 on input x. Similarly, $f\colon \{0,1\}^* \to \{0,1\}^*$ 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. We define $\mathsf{DepParaTC}^k$ as the class of languages (resp. functions) for which there are such parametrizations $\mathrm{par}\colon \{0,1\}^* \to \mathbb{N}$ and families of circuits $(C_{n,D})_{n,D\geq 0}$. Note that this is not a standard definition – but it perfectly fits our purposes.

▶ **Lemma 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 $L \in \mathsf{DepParaTC}^k$ (parametrized by par). Then $\widetilde{L} = \left\{w \in L \;\middle|\; \mathrm{par}(w) \leq C \cdot \lfloor \log |w| \rfloor^\ell\right\}$ is in $\mathsf{TC}^{k+\ell}$.*

Power circuits. Consider a pair (Γ, δ) where Γ is a set of n vertices and δ is a mapping $\delta\colon \Gamma \times \Gamma \to \{-1, 0, +1\}$. Notice that $(\Gamma, \sigma(\delta))$ is a directed graph. 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

[1] Here and in most other natural applications the parameter D is bounded by the input size n. In this case, we could let the size of $C_{n,D}$ be a polynomial in both n and D – without changing the actual class.

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. We define the *evaluation* $\varepsilon(P) \in \mathbb{R}_{>0}$ of a node ($\varepsilon(M) \in \mathbb{R}$ of a marking resp.) bottom-up in the dag by induction: leaves (nodes of out-degree 0) evaluate to 1 and, in general,

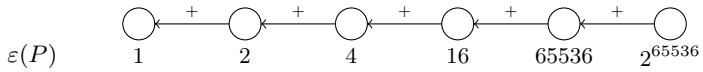
$$\varepsilon(P) = 2^{\varepsilon(\Lambda_P)} \quad \text{for a node } P, \qquad \varepsilon(M) = \sum_P M(P)\varepsilon(P) \quad \text{for a marking } M.$$

▶ **Definition 5.** *A* power circuit *is a pair* (Γ, δ) *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 Γ a power circuit and suppress δ whenever it is clear. If M is a marking on Γ and $S \subseteq \Gamma$, we write $M|_S$ for the restriction of M to S. Let (Γ', δ') be a power circuit, $\Gamma \subseteq \Gamma'$, $\delta = \delta'|_{\Gamma\times\Gamma}$, and $\delta'|_{\Gamma\times(\Gamma'\setminus\Gamma)} = 0$. Then (Γ, δ) itself is a power circuit. We call it a *sub-power circuit* and denote this by $(\Gamma, \delta) \leq (\Gamma', \delta')$ or, if δ is clear, by $\Gamma \leq \Gamma'$.

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

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



▶ **Example 7.** 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 and $\mathcal{O}(n \log n)$ edges (each successor marking requires at most $\lfloor \log n \rfloor + 1$ edges). For an example of a marking representing the integer 23, see Figure 1.

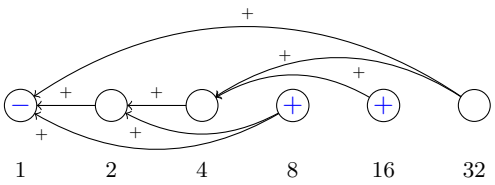


Figure 1 Each integer $z \in [-63\,..\,63]$ can be represented by a marking in the following power circuit. The marking given in blue is representing the number 23.

▶ **Definition 8.** *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* (Γ, δ) *with* Γ *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 Γ are sorted by their values. Still, sometimes it is convenient to treat Γ as a set – we write $P \in \Gamma$ or $S \subseteq \Gamma$ with the obvious meaning. For more details on power circuits see [10, 31].

▶ Remark 9. If (Γ, δ) is a reduced power circuit with $\Gamma = (P_0, \ldots, P_{n-1})$, we have $\delta(P_i, P_j) = 0$ for $j \geq i$. Thus, the order on Γ by evaluations is also a topological order on the dag $(\Gamma, \sigma(\delta))$.

3 Compact signed-digit representations

▶ **Definition 10.**

(i) *A sequence* $B = (b_0, \dots, b_{m-1})$ *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* $B = (b_0, \dots, b_{m-1})$ *is the maximal* i *with* $b_{i-1} \neq 0$.

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

Henceforth, we abbreviate "compact signed-digit representation" with csdr. A non-negative 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). In particular, every integer k can be represented as a signed-digit representation. However, in general, a signed-digit representation for an integer k is not unique. In [31, Section 2.1] a linear-time algorithm for calculating csdrs has been given; here we aim for optimizing the parallel complexity.

▶ **Theorem 11.** *The following is in* AC^0*:*

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

Proof sketch. Computation of the csdr is in the spirit of a carry-lookahead adder: On input of the binary number $A = (a_0, \dots, a_{m-1})$ we define

$$c_i = \bigvee_{1 \le j \le i} \Big(a_j \wedge a_{j-1} \wedge \bigwedge_{j < k \le i} (a_k \vee a_{k-1}) \Big), \quad \text{and} \qquad b_i = (a_i \oplus c_i) \cdot (-1)^{a_{i+1}}.$$

Here $\oplus$ denotes the *exclusive or* and we treat the Boolean values $0, 1$ as a subset of the integers. Then $B = (b_0, \dots, b_{m-1}, b_m)$ can be calculated in AC^0 using the above formulas. The main part of the proof consists in showing that B, indeed, is compact and that $\mathrm{val}(B) = \mathrm{val}(A)$. This is done by induction using the recurrence $c_0 = 0$ and $c_i = (a_i \wedge a_{i-1}) \vee \big(c_{i-1} \wedge (a_i \vee a_{i-1})\big)$ for $i \ge 1$. ◀

▶ **Lemma 12** ([31, Lemma 4])**.** *Let* $A = (a_0, \dots, a_{m-1})$, $B = (b_0, \dots, b_{m-1})$ *be csdrs. Then:*

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

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

From this lemma together with Theorem 11 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.

If A and B are signed digit representations, it follows from Theorem 11 and Lemma 12 that we can calculate $\mathrm{CR}(A)$ and $\mathrm{CR}(A + B)$ and decide whether $\mathrm{val}(A) < \mathrm{val}(B)$ in AC^0.

4 Operations on power circuits

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 from [31, 10] under circuit complexity aspects.

▶ **Definition 13.** *Let* (Γ, δ) *be a reduced power circuit with* $\Gamma = (P_0, \dots, P_{n-1})$.

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

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

(iii) *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 Γ and denote it by $C_0 = C_0(\Gamma)$.*

For an example of a power circuit with three maximal chains, see Figure 2.

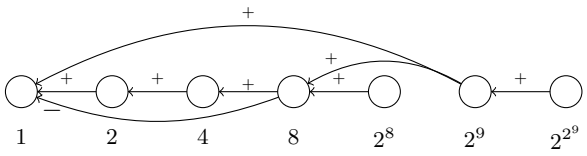


Figure 2 This power circuit is an example for a reduced power circuit with three maximal chains: The first one consists of the nodes of values $1, 2, 4, 8$, the next one is formed by the nodes of values 2^8 and 2^9 and the node of value 2^{2^9} is a maximal chain of length 1.

▶ **Proposition 14.** *Let $\triangle \in \{=, \neq, <, \leq, >, \geq\}$. The following problem is in* AC^0*:*

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

▶ **Lemma 15.** *The following problems are all in* TC^0*:*

(a) Input: *A power circuit (Π, δ_Π) 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 $\operatorname{depth}(\Pi') = \operatorname{depth}(\Pi)$.*

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

(c) Input: *A power circuit (Π, δ_Π) 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 $\operatorname{depth}(\Pi') \leq \operatorname{depth}(\Pi) + 1$.*

Lemma 15 applies the constructions from [31, Section 7] and [10, Section 2]. For (c) it can be summarized as follows: Add L to the successor marking of every node in $\sigma(K)$. To prevent other nodes from changing their value, first create disjoint copies of $\sigma(K)$ and $\sigma(L)$.

▶ Remark 16. Since membership in AC^0 often highly depends on the encoding of the input, we assume that power circuits are given in a suitable way, e.g. as an $n \times n$ matrix representing δ where each entry from $\{0, \pm 1\}$ is encoded using two bits, similarly for markings. If the power circuit is reduced, the nodes appear sorted according to their values.

We need these assumptions for proving the AC^0-bound in Proposition 14. However, in the following, we do not consider these encoding issues because, as soon as we are dealing with TC^0 circuits, there is a lot of freedom how to encode inputs. Also note that in Lemma 15 we only state membership in TC^0, although, with some proper work (and suitable encodings), one could also show AC^0.

Power circuit reduction

While compact markings on a reduced power circuit yield unique representations of integers, in an arbitrary power circuit (Π, δ_Π) 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 17.** *The following is in* $\mathsf{DepParaTC}^0$ *parametrized by* $\mathrm{depth}(\Pi)$*:*

Input: *A power circuit* (Π, δ_Π) *together with a marking* M *on* Π.
Output: *A reduced power circuit* (Γ, δ) *together with a compact marking* $\widetilde{M}$ *on* Γ *such that* $\varepsilon(\widetilde{M}) = \varepsilon(M)$.

For a power circuit (Π, δ_Π) with a marking M we call the power circuit (Γ, δ) together with the marking $\widetilde{M}$ obtained by Theorem 17 the *reduced form* of Π.

The proof of Theorem 17 consists of several steps, which we introduce on the next pages. The high-level idea is as follows: Like in [31, 10], we keep the invariant that there is an already reduced part and a non-reduced part (initially the non-reduced part is Π). 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 TC^0; after $\mathrm{depth}(\Pi) + 1$ iterations we obtain a reduced power circuit.

Insertion of new nodes. The following procedure is a basic tool for the reduction process. Let (Γ, δ) 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 $\Lambda_P \colon \Gamma \to \{-1, 0, 1\}$ such that Λ_P is compact and $\varepsilon(\Lambda_P) \geq 0$ for all $P \in I$, and $\varepsilon(\Lambda_P) \neq \varepsilon(\Lambda_Q)$ for all $P, Q \in I \cup \Gamma$ with $P \neq Q$.

We wish to add I to the reduced power circuit (Γ, δ). 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, (Γ', δ') is a power circuit with $(\Gamma, \delta) \leq (\Gamma', \delta')$ and for every $P \in I$ the map Λ_P is the successor marking of P. Moreover, each node of Γ' has a unique value. Since for every node $P \in \Gamma'$ the marking Λ_P is a compact marking on the reduced power circuit Γ, by Proposition 14, for $P, Q \in \Gamma'$ we are able to decide in AC^0 whether $\varepsilon(\Lambda_Q) \leq \varepsilon(\Lambda_P)$. Therefore, by Example 2 we can sort Γ' according to the values of the nodes in TC^0 and, hence, assume that $\Gamma' = (P_0, \ldots, P_{|\Gamma'|-1})$ is in increasing order. This yields the following:

▶ **Lemma 18** (InsertNodes)**.** *The following problem is in* TC^0*:*

Input: *A power circuit* (Γ, δ) *and a set* I *with the properties described above.*
Output: *A reduced power circuit* (Γ', δ') *such that* $(\Gamma, \delta) \leq (\Gamma', \delta')$ *and such that for every* $P \in I$ *there is a node* Q *in* Γ' *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 (Π, δ_Π) with a marking M consists of several iterations. Each iteration starts with a power circuit $(\Gamma_i \cup \Xi_i, \delta_i)$ such that Γ_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 Γ_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 UpdateNodes, ExtendChains, and UpdateMarkings, which can be done in TC^0. We have $\Xi_{i+1} = \Xi_i \setminus \mathrm{Min}(\Xi_i)$. Thus, the full reduction process consists of $\mathrm{depth}(\Pi) + 1$ many TC^0 computations. Let us now describe these three steps. The proofs of these Lemmas can be found in the appendix.

We write $(\Gamma \cup \Xi, \delta) = (\Gamma_i \cup \Xi_i, \delta_i)$ for the power circuit at the start of one iteration. 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 19** (UpdateNodes). *The following problem is in* TC^0*:*

Input: *A power circuit* $(\Gamma \cup \Xi, \delta)$ *as above.*

Output: *A reduced power circuit* (Γ', δ') *such that* $(\Gamma, \delta|_{\Gamma\times\Gamma}) \leq (\Gamma', \delta')$ *and such that for every node* $Q \in \mathrm{Min}(\Xi)$ *there exists a node* $P \in \Gamma'$ *with* $\varepsilon(P) = \varepsilon(Q)$*. In addition,* $|\Gamma'| \leq |\Gamma| + |\mathrm{Min}(\Xi)|$*, and* $|\mathcal{C}_{\Gamma'}| \leq |\mathcal{C}_\Gamma| + |\mathrm{Min}(\Xi)|$*.*

▶ **Lemma 20** (ExtendChains). *The following problem is in* TC^0*:*

Input: *A reduced power circuit* (Γ', δ') *and* $\mu \in \mathbb{N}$ *such that* $\mu \leq \left\lfloor \frac{2^{|C_0(\Gamma')|+1}}{3} \right\rfloor$.

Output: *A reduced power circuit* (Γ'', δ'') *such that* $(\Gamma', \delta') \leq (\Gamma'', \delta'')$ *and such that for each* $P \in \Gamma'$ *and each* $i \in [0\,..\,\mu]$ *there is a node* $Q \in \Gamma''$ *with* $\varepsilon(\Lambda_Q) = \varepsilon(\Lambda_P) + i$*. In addition,* $|\Gamma''| \leq |\Gamma'| + |\mathcal{C}_{\Gamma'}| \cdot \mu$*, and* $|\mathcal{C}_{\Gamma''}| \leq |\mathcal{C}_{\Gamma'}|$*.*

In the following, (Γ', δ') denotes the power circuit obtained by UpdateNodes when starting with $(\Gamma \cup \Xi, \delta)$, and (Γ'', δ'') denotes the power circuit obtained by ExtendChains with $\mu = \lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 1$ on input of the power circuit (Γ', δ') (observe that, by the assumption $|C_0(\Gamma)| \geq \lceil \log(|\Xi|) \rceil + 1$, the condition on μ in Lemma 20 is satisfied). The value of μ 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 21** (UpdateMarkings). *The following problem is in* TC^0*:*

Input: *The power circuit* (Γ'', δ'') *as a result of* ExtendChains *with* $\mu = \lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 1$ *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* $\widetilde{M}|_{\Gamma''}$ *is compact.*

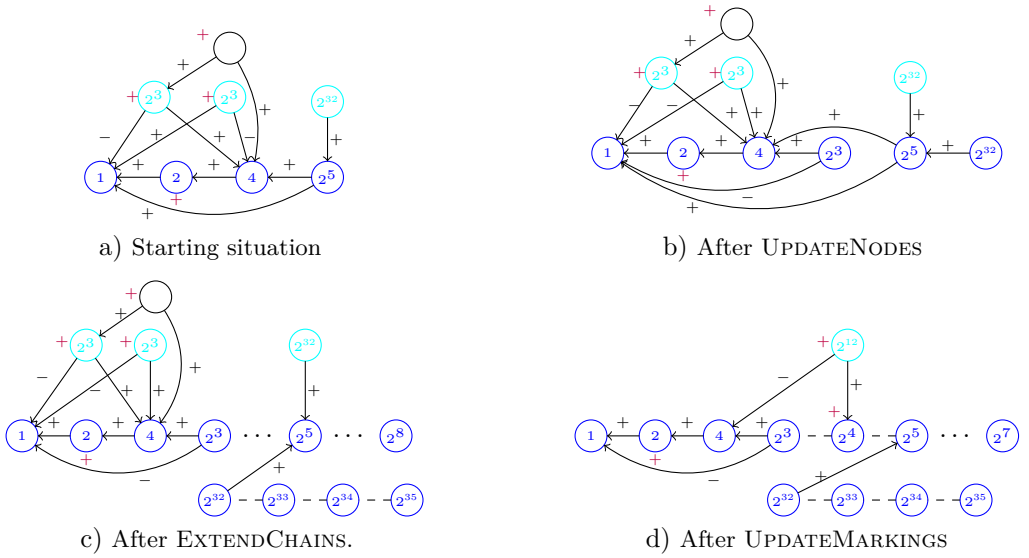
Proof sketch of Theorem 17. We start with an initial reduced power circuit (Γ_0, δ_0) (a chain of length $\lceil \log(|\Pi|) \rceil + 1$) and a non-reduced part $\Xi_0 = \Pi$ and successively apply the three steps (Lemma 19, 20, and 21) to obtain a sequence of power circuits $(\Gamma_i \cup \Xi_i, \delta_i)$ and markings M_i for $i = 0, 1 \ldots$ with $\Xi_{i+1} = \Xi_i \setminus \mathrm{Min}(\Xi_i)$ while keeping the invariants $(\Gamma_i, \delta_i|_{\Gamma_i\times\Gamma_i}) \leq (\Gamma_i \cup \Xi_i, \delta_i)$, Γ_i is reduced, $\Gamma_{i-1} \leq \Gamma_i$, $\Xi_i \subseteq \Xi_{i-1}$, and $\varepsilon(M_i) = \varepsilon(M)$. After $\mathrm{depth}(\Pi) + 1$ iterations we reach $\Xi_{d+1} = \Xi_d \setminus \mathrm{Min}(\Xi_d) = \emptyset$ where $d = \mathrm{depth}(\Pi)$. Thus, $(\Gamma, \delta) = (\Gamma_{d+1}, \delta_{d+1})$ is a reduced power circuit and M_{d+1} is a compact marking on Γ_{d+1} with $\varepsilon(M_{d+1}) = \varepsilon(M)$.

▷ Claim 22. Let $d = \mathrm{depth}(\Pi)$ and $\Gamma_0, \ldots, \Gamma_{d+1}$ be as constructed above. Then for all i we have $|\mathcal{C}_{\Gamma_i}| \leq |\Pi| + 1$ and $|\Gamma_i| \leq (|\Pi| + 1)^2 \cdot (\log(|\Pi|) + 2)$.

Let $D \in \mathbb{N}$ and assume that $\mathrm{depth}(\Pi) \leq D$. By Lemma 19, 20, and 21, each iteration can be done in TC^0. The construction of the markings $\widetilde{M}_i$ and $\tilde{\Lambda}_P$ during UpdateMarkings can be done in parallel – so it is in TC^0, although Lemma 21 is stated only for a single marking. Now, the crucial observation is that, due to Claim 22, the input size for each iteration is polynomial in the original input size of (Π, δ_Π). Therefore, we can compose the individual iterations and obtain a circuit of polynomial size and depth bounded by $\mathcal{O}(D)$. ◀

▶ Remark 23.

(1) While Theorem 17 is only stated for one input marking, the construction works within the same complexity bounds for any number of markings on (Π, δ_Π) since during UpdateMarkings these all can be updated in parallel.



a) Starting situation

b) After UpdateNodes

c) After ExtendChains.

d) After UpdateMarkings

Figure 3 The three steps of power circuit reduction. The already reduced part consist of blue nodes and $\mathrm{Min}(\Xi_i)$ 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.

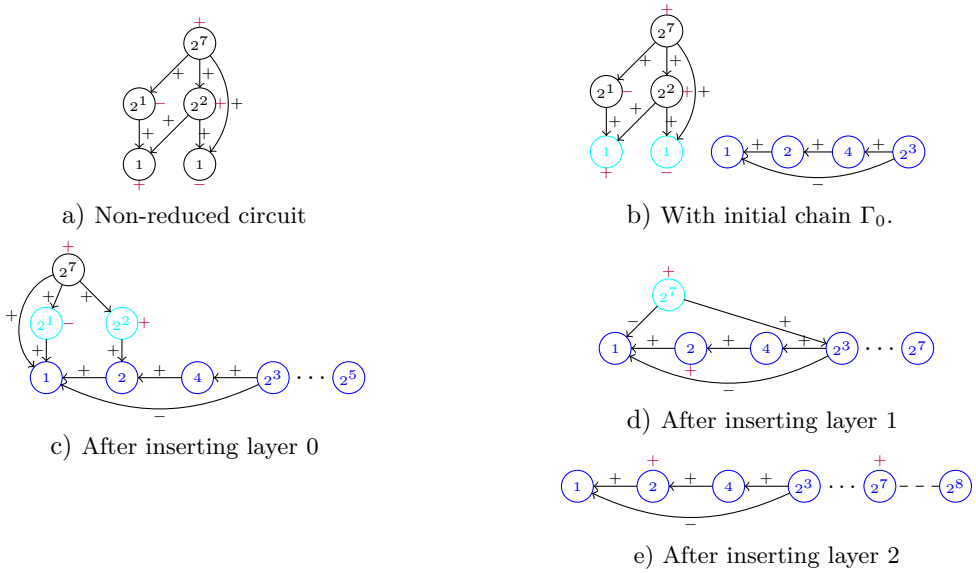
(2) Moreover, note that for every maximal chain $C \in \mathcal{C}_\Gamma$ there exists a node $Q \in \Pi$ (i.e., in the original power circuit) such that $\varepsilon(Q) = \varepsilon(\mathrm{start}(C))$. This is because new chains are only created during UpdateNodes, the other steps only extend already 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.

▶ **Example 24.** In Figure 3 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 deleted the nodes from $\mathrm{Min}(\Xi)$.

▶ **Example 25.** In Section 4 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 15), then reduce the power circuit (Theorem 17) and, finally, compare the resulting compact marking with zero (Proposition 14). This shows the next corollary and, together with Lemma 4, also proves Proposition A.



a) Non-reduced circuit

b) With initial chain Γ_0.

c) After inserting layer 0

d) After inserting layer 1

e) After inserting layer 2

Figure 4 The complete process of power circuit reduction – inserting layer after layer. For the meaning of the colors, see Figure 3.

▶ **Corollary 26.** *The following is in* DepParaTC^0 *parametrized by* depth(Π)*:*

Input: *A power circuit* (Π, δ_Π) *together with markings* L, M *on* Π.
Question: *Is* $\varepsilon(L) \leq \varepsilon(M)$*?*

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 $r = u \cdot 2^e$.

▶ **Definition 27.** *A power circuit representation of* $r \in \mathbb{Z}[1/2]$ *consists of a power circuit* (Π, δ_Π) *together with a pair of markings* (U, E) *on* Π *such that* $\varepsilon(U)$ *is either zero or odd and* $r = \varepsilon(U) \cdot 2^{\varepsilon(E)}$.

▶ **Lemma 28.** *The following problems are in* DepParaTC^0 *parametrized by* depth(Π)*:*

Input: *A power circuit representation for* $r, s \in \mathbb{Z}[1/2]$ *over a power circuit* (Π, δ_Π) *and a marking* M *on* Π.
Output A: *A power circuit representation of* $\varepsilon(M) \in \mathbb{Z}[1/2]$ *over a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$.
Output B: *A power circuit representation of* $r \cdot 2^{\varepsilon(M)}$ *over a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$.
Output C: *A power circuit representation of* $-r$ *over* (Π, δ_Π).
Output D: *If* $\frac{r}{s}$ *is a power of two, a marking* L *in a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ *such that* $\varepsilon(L) = \log(\frac{r}{s})$ *(otherwise the output is undefined).*
Output E: *A power circuit representation of* $r + s$ *over a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$.
Output F: *Is* $r \in \mathbb{Z}$*? If yes, a marking* L *in a power circuit* $(\tilde{\Pi}, \delta_{\tilde{\Pi}})$ *such that* $\varepsilon(L) = r$.
Question G: *Is* $r \triangle 0$ *for* $\triangle \in \{=, \neq, <, \leq, >, \geq\}$*?*

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

Proof sketch. We only outline the proof for the first point, which is the most difficult one. The other points follow rather easily using Corollary 26 and Lemma 15. Given a marking M, we wish to compute markings U, E such that $\varepsilon(M) = \varepsilon(U) \cdot 2^{\varepsilon(E)}$ and $\varepsilon(U)$ is zero or odd.

First, we construct the reduced form (Γ, δ) of Π to obtain a compact marking $\widetilde{M}$ on Γ such that $\varepsilon(M) = \varepsilon(\widetilde{M}) = \sum_{i=1}^{k} \widetilde{M}(Q_i) \cdot 2^{\varepsilon(\Lambda_{Q_i})}$ where $\sigma(\widetilde{M}) = \{Q_1, \ldots, Q_k\} \subseteq \Gamma$ and the Q_i are ordered according to their value. This is possible in $\mathsf{DepParaTC}^0$ according to Theorem 17. It is easy to see that $|\sigma(\widetilde{M})| \leq |\sigma(M)|$.

Our aim is $\varepsilon(E) = \varepsilon(\Lambda_{Q_1})$ and $\varepsilon(U) = \sum_{i=1}^{k} \widetilde{M}(Q_i) \cdot 2^{\varepsilon(\Lambda_{Q_i}) - \varepsilon(E)}$. For this, we add nodes S_i to Π with $\varepsilon(\Lambda_{S_i}) = \varepsilon(\Lambda_{Q_i}) - \varepsilon(E)$ for $i \in [1 \,..\, k]$ as follows: Looking closely at the reduction process, we can find nodes $R_i \in \Pi$ and integers $m_i \in [0 \,..\, |\Gamma|]$ such that $\varepsilon(\Lambda_{Q_i}) = \varepsilon(\Lambda_{R_i}) + m_i$. To define markings M_i that evaluate to m_i, we construct nodes of depth 1 and values 2^j for $j \in [0 \,..\, \lfloor \log(|\Gamma|) \rfloor]$ in Π. Then $\Lambda_{S_i} = \Lambda_{R_i} + M_i - E$. So $U(S_i) = \widetilde{M}(Q_i)$ for $i \in [1 \,..\, k]$ and $E = \Lambda_{R_1} + M_1$ satisfies $\varepsilon(M) = \varepsilon(U) \cdot 2^{\varepsilon(E)}$. ◀

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. Let G be a group and $\eta\colon \Sigma^* \to G$ a surjective monoid homomorphism. We treat words over Σ both as words and as their images under η. We write $v =_G w$ with the meaning that $\eta(v) = \eta(w)$. The word problem of G is as follows: given a word $w \in \Sigma^*$, is $w =_G 1$? For further background on group theory, we refer to [25].

The Baumslag-Solitar group and the Baumslag group. 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 $t \mapsto (0, 1)$. The multiplication in $\mathbb{Z}[1/2] \rtimes \mathbb{Z}$ isdefined by $(r, m) \cdot (s, n) = (r + 2^m s, m + n)$. In the following we use $\mathbf{BS}_{1,2}$ and $\mathbb{Z}[1/2] \rtimes \mathbb{Z}$ as synonyms.

A convenient way to understand the Baumslag group $\mathbf{G}_{1,2}$ is as an HNN extension[2] of the Baumslag-Solitar group:

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

Note that the letter t can be seen as an abbreviation for bab^{-1}; by removing it, 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. We have $b(q, 0)b^{-1} = (0, q)$, 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).

Britton reductions. Britton reductions are a standard way to solve the word problem in HNN extensions. 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 $(r, m)(s, n) \to (r + 2^m s, m + n)$, $b(q, 0)b^{-1} \to (0, q)$, or $b^{-1}(0, k)b \to (k, 0)$ in order to obtain a shorter word representing the same group element. The following lemma is well-known (see also [25, Section IV.2]).

▶ **Lemma 29** (Britton's Lemma for $\mathbf{G}_{1,2}$ [6]). *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.*

[2] Named after Graham Higman, Bernhard H. Neumann and Hanna Neumann. For a precise definition, we refer to [25]. This is also the way how the Magnus breakdown procedure works.

▶ **Example 30.** 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 $\tau(n)$.

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.

▶ **Lemma 31.** *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.

β_1	β_2	*Condition*	$\hat{w}$
b	b	$r + 2^{m+k}s \in \mathbb{Z}, \quad m+n+k = 0$	$\big(0,\ r + 2^{-n}s\big)$
b	b^{-1}	$r + 2^{m}(q+s) \in \mathbb{Z}, \quad m+n = 0$	$(0,\ r + 2^{m}(q+s))$
b^{-1}	b	$r + 2^{m+k}s = 0$	$\big(n + \log(\frac{-r}{s}),\ 0\big)$
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 32.** Let us illustrate how to read Lemma 31 by giving an example. 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 31, in particular, $\beta_2\, x\, \beta_2^{-1} =_{\mathbf{G}_{1,2}} (q,k) \in \mathbf{BS}_{1,2}$. Further assume that $\beta_1 = \beta_2 = b$. Then, according to Lemma 31, $w \in \mathbf{BS}_{1,2}$ if and only if $m_1 + n_1 = -k$ and $r_1 + 2^{m_1+k} \cdot s_1 \in \mathbb{Z}$. So we need to compute k.

Assume that $(q,k) =_{\mathbf{G}_{1,2}} \beta_2 x \beta_2^{-1} = \beta_2(r_2,m_2)\beta_3\, x'\, \beta_3^{-1}(s_2,n_2)\beta_2^{-1}$ for some r_2, m_2, s_2, n_2. Moreover, consider the case that $\beta_3 = b$. By applying Lemma 31 again we obtain that $(q,k) = (0, r_2 + 2^{-n_2} \cdot s_2)$. 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, then $w =_{\mathbf{G}_{1,2}} (0,\ r_1 + 2^{-n_1}s_1)$.

Proof sketch of Lemma 31. Consider the case that $\beta_1 = b$ and $\beta_2 = b$: 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

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

Thus, since $\beta_1 = b$, 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 $w =_{\mathbf{G}_{1,2}} b(r + 2^{m+k}s,\ 0)b^{-1} = b(r + 2^{-n}s,\ 0)b^{-1} =_{\mathbf{G}_{1,2}} (0,\ r + 2^{-n}s)$. This shows the first row of the table in Lemma 31. The other rows follow with a similar calculation. ◀

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

$$u = (r_h,m_h)\beta_h \cdots (r_1,m_1)\beta_1(r_0,m_0), \qquad v = (s_0,n_0)\tilde{\beta}_1(s_1,n_1)\cdots\tilde{\beta}_\ell(s_\ell,n_\ell) \tag{1}$$

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 (1) 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 $j \leq i$. Moreover, uv is Britton-reduced if and only if $\beta_1(r_0, m_0)(s_0, n_0)\widetilde{\beta}_1 \notin \mathbf{BS}_{1,2}$.

For $\ell \in \mathbb{N}$ let $\mathcal{X}_\ell$ denote some set of ℓ 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 33.** *For every $\vec{\beta} \in \{b, b^{-1}, \bot\}^4$ 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 $u, v \in \mathbf{G}_{1,2}$ as in (1) be Britton-reduced and assume that $uv[i-1, i-1] \in \mathbf{BS}_{1,2}$ and $\beta_i = \widetilde{\beta}_i^{-1}$ and let $V_i = \{r_j, s_j, m_j, n_j \mid j \in \{i-1, i-2, i-3\}\}$. If $\vec{\beta} = (\beta_i, \beta_{i-1}, \beta_{i-2}, \beta_{i-3})$ (where $\beta_j = \bot$ for $j \leq 0$), then*

1. *$uv[i,i] \in \mathbf{BS}_{1,2}$ if and only if $\theta_{\vec{\beta}}(V_i) \in \mathbb{Z}$ and $\xi_{\vec{\beta}}(V_i) = 0$,*
2. *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 (at most) 12 as 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 34 below).

Proof. W. l. o. g. $i \geq 4$. We follow the approach of Example 32. 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 31, to show Lemma 33 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 31. 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 31 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. ◀

The algorithm. A power circuit representation of $u \in \mathbf{G}_{1,2}$ written as in (1) consists of the sequence $(\beta_h, \ldots, \beta_1)$ and a power circuit (Π, δ_Π) 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 27) and $m_i = \varepsilon(M_i)$.

▶ **Lemma 34.** *The following problem is in $\mathsf{DepParaTC}^0$ parametrized by $\max_i \mathrm{depth}(\Pi_i)$:*

Input:	*Britton-reduced power circuit representations of $u, v \in \mathbf{G}_{1,2}$ over power circuits Π_1, Π_2.*
Output:	*A Britton-reduced power circuit representation of $w \in \mathbf{G}_{1,2}$ over a power circuit Π' such that $w =_{\mathbf{G}_{1,2}} uv$ and $\mathrm{depth}(\Pi') = \max_i \mathrm{depth}(\Pi_i) + \mathcal{O}(1)$ and $\lvert\Pi'\rvert \in \mathcal{O}(\lvert\Pi_1\rvert + \lvert\Pi_2\rvert)$.*

Proof. Let Π be the disjoint union of Π_1 and Π_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 28 one can evaluate the expressions $\theta_{\vec{\beta}}(V_i)$ and $\xi_{\vec{\beta}}(V_i)$ of Lemma 33 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 33 one can compute again using Lemma 28 $(q,k) = (\varphi_{\vec{\beta}}(V_{i_0}), \psi_{\vec{\beta}}(V_{i_0})) =_{\mathbf{G}_{1,2}} uv[i_0, i_0]$ in $\mathsf{DepParaTC}^0$. Again using Lemma 28, 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 $\mathrm{depth}(\Pi') \in \mathrm{depth}(\Pi) + \mathcal{O}(1)$. Now, the output is

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

Instead of Theorem B, we prove the following slightly more general result. Theorem B then easily follows by Britton's Lemma. Recall that $\Sigma = \{1, a, a^{-1}, t, t^{-1}, b, b^{-1}\}$.

▶ **Theorem 35.** *The following problem is in* 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 sketch. Let $w = w_1 \cdots w_n$ with $w_j \in \Sigma$ be the input. First, we transform each letter w_j into a power circuit representation. Then, the first layer computes the Britton reduction of two-letter words using Lemma 34, the next layer takes always two of these Britton-reduced words and joins them to a new Britton-reduced word and so on. After $\log n$ layers a single Britton-reduced word remains. The crucial observation is that, due to Lemma 34, the size of the power circuits stays polynomial in n and their depth in $\mathcal{O}(\log n)$. Thus, by Lemma 4 each application of Lemma 34 is in TC^1 and the whole computation in TC^2. ◀

6 P-hardness of power circuit comparison

Finally, we prove some hardness results on comparison in power circuits. In particular, they imply that Theorem 17 is optimal in a certain sense. Here, we use LOGSPACE-reductions.

▶ **Proposition 36.** *The following problem is* NL*-hard:*

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

The proof of Proposition 36 is a straightforward reduction from s-t-connectivity. For comparison with $\leq$, we obtain a more interesting hardness result:

▶ **Theorem 37.** *The following problem is* P*-complete:*

Input: *A power circuit* (Π, δ_Π) *and nodes* $R, S \in \Pi$ *such that for all* $P \in \Pi$ *the marking* Λ_P *is compact and for all* $P \neq Q$, $\varepsilon(P) \neq \varepsilon(Q)$.
Question: *Is* $\varepsilon(R) \leq \varepsilon(S)$*?*

A weaker form of this result already has been stated in the second author's dissertation [41], but it never appeared in a refereed journal or conference proceedings. Notice that the only feature the power circuit in Theorem 37 lacks for being reduced is the sorting of the nodes. In particular, under the assumption $\mathsf{NC} \neq \mathsf{P}$, it is not possible to sort the nodes of a given power circuit in NC.

▶ Remark 38.

(a) It is an immediate consequence of Theorem 37 that the comparison problem of two markings in a power circuit is P-complete. This is because for two nodes R and S in a power circuit (Π, δ_Π) we have $\varepsilon(R) \leq \varepsilon(S)$ if and only if $\varepsilon(\Lambda_R) \leq \varepsilon(\Lambda_S)$.

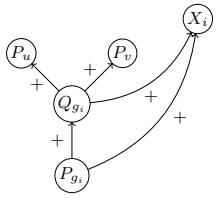


Figure 5 Power circuit for an OR gate g_i.

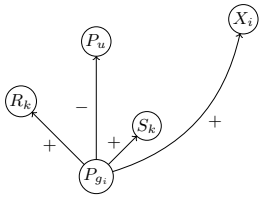


Figure 6 Power circuit for a NOT gate g_i on level k.

(b) If the input is given as in Theorem 37, we can check in AC^0 whether $\varepsilon(R) = \varepsilon(S)$ because this is the case if and only if $\Lambda_R(P) = \Lambda_S(P)$ for all $P \in \Gamma$ (see Lemma 12). 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 39.** *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 sketch of Theorem 37. By [31, Proposition 49], we only need to show the hardness part. We give a reduction from the CIRCUITVALUEPROBLEM which is P-complete (see [39, Thm. 10.44]). We start with a circuit $\mathcal{C}$ of size L and depth D consisting of *input gates*, NOT gates, OR gates (of fan-in two) and one *output gate*. W. l. o. g. the circuit is layered: input gates are on level 0, and gates on level k only receive inputs from level $k-1$. After fixing an evaluation $\mathrm{eval}(x) \in \{0,1\}$ for all input gates x, each gate g evaluates to a truth value $\mathrm{eval}(g) \in \{0,1\}$ in a natural way. The task is to compute $\mathrm{eval}(\mathit{output})$. We construct a power circuit (Γ, δ) such that for every gate g on level k in $\mathcal{C}$ there exists a node P_g in Γ satisfying

$$\begin{aligned} \tau(L-1) < \ \varepsilon(\Lambda_{P_g}) \ &\leq \tau(2k+L)-2 && \text{if } \mathrm{eval}(g)=0, \\ \tau(2k+L) \leq \ \varepsilon(\Lambda_{P_g}) \ &\leq \tau(2k+L+1)-2 && \text{if } \mathrm{eval}(g)=1. \end{aligned} \tag{2}$$

For this, we first create nodes X_k, R_k and S_k such that $\varepsilon(X_k) = 2^k$, $\varepsilon(R_k) = \tau(2k+L)$, $\varepsilon(S_k) = \tau(2k+L-1)/2$. For an input gate g_i we set $\varepsilon(\Lambda_{P_{g_i}}) = \tau(L-1)+i$ if $\mathrm{eval}(g_i) = 0$ and $\varepsilon(\Lambda_{P_{g_i}}) = \tau(L)+i$ otherwise. For the *output* gate with incoming edge from gate u, we define $\varepsilon(\Lambda_{P_{\mathrm{output}}}) = \varepsilon(P_u)$. Figure 5 and 6 illustrate the construction for OR and NOT gates. Now all nodes of Γ have pairwise different evaluations in $2^{\mathbb{N}}$ (this is essentially because we always add i to the successor marking) and compact successor markings. A rather tedious but straightforward induction shows Equation (2). Let us consider an OR gate as in Figure 5 as example: if both $\varepsilon(\Lambda_{P_u}), \varepsilon(\Lambda_{P_v}) \leq \tau(2(k-1)+L)-2$, then $\varepsilon(\Lambda_{P_g}) \leq 2^{2^{\varepsilon(\Lambda_{P_u})}+2^{\varepsilon(\Lambda_{P_v})}} \leq 2^{2\cdot 2^{\tau(2(k-1)+L)-2}} \leq \tau(2k+L)-2$. On the other hand, if $\varepsilon(\Lambda_{P_u}) \geq \tau(2(k-1)+L)$, then also $\varepsilon(\Lambda_{P_g}) \geq 2^{2^{\varepsilon(\Lambda_{P_u})}} \geq \tau(2k+L)$. The other cases of the induction follow similarly.

Thus, we have that $\varepsilon(P_{\mathrm{output}}) \geq \varepsilon(R_D)$ if and only if $\mathrm{eval}(\mathit{output}) = 1$. ◀

Conclusion. We showed that the word problem of the Baumslag group can be solved in TC^2. The proof relies on the fact that all power circuits used during the execution of the algorithm have logarithmic depth. The-23 comparison problem for such power circuits is in 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? Can we prove some non-trivial lower bounds (the word problem is NC^1-hard as $\mathbf{G}_{1,2}$ contains a non-abelian free group [35])?
- 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 39 this holds for the subgroup membership problem for $\mathbf{BS}_{1,2}$ in $\mathbf{G}_{1,2}$. Moreover, as a consequence of Proposition 36, the word problem is NL-hard.)
- By Corollary 26 for every k the comparison problem for power circuits of depth $\log^k n$ is in TC^k. Moreover, the proof of Theorem 37 can be modified to show that the same problem is hard for AC^k under AC^0-Turing-reductions. Thus, the question remains whether, indeed, this problem is complete for TC^k under AC^0-Turing-reductions.

References

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

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

3 G. Baumslag, A. G. Myasnikov, and V. Shpilrain. Open problems in combinatorial group theory. Second Edition. In *Combinatorial and geometric group theory*, volume 296 of *Contemporary Mathematics*, pages 1–38. American Mathematical Society, 2002.

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

5 W. W. Boone. The Word Problem. *Ann. of Math.*, 70(2):207–265, 1959.

6 John L. Britton. The word problem. *Ann. of Math.*, 77:16–32, 1963.

7 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3 edition, 2009.

8 Max Dehn. Ueber unendliche diskontinuierliche Gruppen. *Math. Ann.*, 71:116–144, 1911.

9 Volker Diekert, Jürn Laun, and Alexander Ushakov. Efficient algorithms for highly compressed data: The word problem in Higman's group is in P. In *Proc. 29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, Paris, France*, volume 14 of *LIPIcs*, pages 218–229. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.STACS.2012.218.

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

11 Volker Diekert, Alexei G. Myasnikov, and Armin Weiß. Conjugacy in Baumslag's Group, Generic Case Complexity, and Division in Power Circuits. In Alberto Pardo and Alfredo Viola, editors, *Latin American Theoretical Informatics Symposium*, volume 8392 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2014. doi:10.1007/978-3-642-54423-1_1.

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

13 Will Dison, Eduard Einstein, and Timothy R. Riley. Ackermannian integer compression and the word problem for hydra groups. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 – Kraków, Poland*, pages 30:1–30:14, 2016. doi:10.4230/LIPIcs.MFCS.2016.30.

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

15 S. M. Gersten. Isodiametric and isoperimetric inequalities in group extensions. Preprint, 1991.

16 Graham Higman. A finitely generated infinite simple group. *J. London Math. Soc.*, 26:61–64, 1951.

17 I. Kapovich, A. G. Miasnikov, P. Schupp, and V. Shpilrain. Generic-case complexity, decision problems in group theory and random walks. *Journal of Algebra*, 264:665–694, 2003.

18 Jonathan Kausch. *The parallel complexity of certain algorithmic problems in group theory.* Dissertation, Institut für Formale Methoden der Informatik, Universität Stuttgart, 2017.

19 Daniel König and Markus Lohrey. Evaluation of circuits over nilpotent and polycyclic groups. *Algorithmica*, 80(5):1459–1492, 2018. doi:10.1007/s00453-017-0343-z.

20 Jürn Laun. Efficient algorithms for highly compressed data: The word problem in generalized Higman groups is in P. *Theory Comput. Syst.*, 55(4):742–770, 2014. doi:10.1007/s00224-013-9509-5.

21 J. Lehnert and P. Schweitzer. The co-word problem for the Higman-Thompson group is context-free. *Bull. London Math. Soc.*, 39:235–241, 2007. doi:10.1112/blms/bdl043.

22 Richard J. Lipton and Yechezkel Zalcstein. Word problems solvable in logspace. *J. ACM*, 24:522–526, 1977.

23 Markus Lohrey. Decidability and complexity in automatic monoids. *International Journal of Foundations of Computer Science*, 16(4):707–722, 2005.

24 Markus Lohrey. *The Compressed Word Problem for Groups.* Springer Briefs in Mathematics. Springer, 2014. doi:10.1007/978-1-4939-0748-9.

25 Roger Lyndon and Paul Schupp. *Combinatorial Group Theory.* Classics in Mathematics. Springer, 2001. First edition 1977.

26 Wilhelm Magnus. Das Identitätsproblem für Gruppen mit einer definierenden Relation. *Mathematische Annalen*, 106:295–307, 1932.

27 Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial Group Theory.* Dover, 2004.

28 Caroline Mattes and Armin Weiß. Parallel algorithms for power circuits and the word problem of the Baumslag group. *CoRR*, abs/2102.09921, 2021. arXiv:2102.09921.

29 Alexei Miasnikov and Andrey Nikolaev. On parameterized complexity of the word search problem in the Baumslag-Gersten group. In *ISSAC '20: International Symposium on Symbolic and Algebraic Computation, Kalamata, Greece, July 20-23, 2020*, pages 360–363, 2020. doi:10.1145/3373207.3404042.

30 Alexei G. Myasnikov, Alexander Ushakov, and Won Dong-Wook. The Word Problem in the Baumslag group with a non-elementary Dehn function is polynomial time decidable. *Journal of Algebra*, 345:324–342, 2011. URL: http://www.sciencedirect.com/science/article/pii/S0021869311004492.

31 Alexei G. Myasnikov, Alexander Ushakov, and Won Dong-Wook. Power circuits, exponential algebra, and time complexity. *International Journal of Algebra and Computation*, 22(6):3–53, 2012.

32 Alexei G. Myasnikov and Sasha Ushakov. Cryptography and groups (CRAG). Software Library. URL: http://www.stevens.edu/algebraic/downloads.php.

33 P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov*, pages 1–143, 1955. In Russian.

34 A. N. Platonov. Isoparametric function of the Baumslag-Gersten group. *Vestnik Moskov. Univ. Ser. I Mat. Mekh.*, 3:12–17, 2004. Russian. Engl. transl. Moscow Univ. Math. Bull. 59 (3) (2004), 12–17.

35 David Robinson. *Parallel Algorithms for Group Word Problems.* PhD thesis, University of California, San Diego, 1993.

36 Mark V. Sapir, Jean-Camille Birget, and Eliyahu Rips. Isoperimetric and Isodiametric Functions of Groups. *Ann. Math.*, 156(2):345–466, 2002.

37 A. L. Semenov. Logical theories of one-place functions on the natural number series. *Izv. Akad. Nauk SSSR Ser. Mat.*, 47(3):623–658, 1983.

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

39 Michael Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1st edition, 1996.

40 Heribert Vollmer. *Introduction to Circuit Complexity*. Springer, Berlin, 1999.

41 Armin Weiß. *On the Complexity of Conjugacy in Amalgamated Products and HNN Extensions*. Dissertation, Institut für Formale Methoden der Informatik, Universität Stuttgart, 2015.

42 Armin Weiß. A logspace solution to the word and conjugacy problem of generalized Baumslag-Solitar groups. In *Algebra and Computer Science*, volume 677 of *Contemporary Mathematics*, pages 185–212. American Mathematical Society, 2016.

A Details on power circuit reduction

In the following we present more details concerning the reduction process for power circuits. We give the proofs of the three steps UpdateNodes, ExtendChains, UpdateMarkings and of Theorem 17. We need the following definition and lemmas. Their proofs can be found in the full version on arXiv [28].

▶ **Lemma 40.** *Let A be a csdr and let $B = (b_0, \dots, b_{n-1})$ be a csdr 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) $\operatorname{val}(B) = \left\lfloor \frac{2^{n+1}}{3} \right\rfloor$,

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

▶ **Definition 41.** *Let M be a marking in the reduced power circuit (Γ, δ) and let $C = (P_i, \dots, P_{i+\ell-1}) \in \mathcal{C}_\Gamma$ and define $a_j = M(P_{i+j})$ for $i \in [\ell]$. Then we write* $\operatorname{digit}_C(M) = (a_0, \dots, a_{\ell-1})$.

▶ **Lemma 42.** *Let (Γ, δ) be a reduced power circuit. Let L and M be compact markings in Γ such that $\varepsilon(L) > \varepsilon(M)$ and let $0 \le k \le \left\lfloor \frac{2^{|C_0|+1}}{3} \right\rfloor$. Then $\varepsilon(L) \le \varepsilon(M) + k$ if and only if $\varepsilon(M|_{\Gamma \setminus C_0}) = \varepsilon(L|_{\Gamma \setminus C_0})$ and $\varepsilon(L|_{C_0}) \le \varepsilon(M|_{C_0}) + k$.*

For the proof of Lemma 19, we define the following equivalence relation $\sim_\varepsilon$ on $\Gamma \cup \operatorname{Min}(\Xi)$: $P \sim_\varepsilon Q$ if and only if $\varepsilon(P) = \varepsilon(Q)$. For $P \in \Gamma \cup \operatorname{Min}(\Xi)$ we write $[P]_\varepsilon$ for the equivalence class containing P.

Proof of Lemma 19. Define $I \subseteq \operatorname{Min}(\Xi)$ by taking one representative of each $\sim_\varepsilon$-class not containing a node of Γ. Such a set I can be computed in TC^0: Clearly, $\operatorname{Min}(\Xi)$ can be computed in TC^0. The $\sim_\varepsilon$-classes can be computed in AC^0 by Proposition 14. Finally, for defining I one has to pick representatives, which can be done in TC^0. Now, we can apply Lemma 18 to insert I into Γ in TC^0. This yields our power circuit (Γ', δ'). The size bounds follow now immediately from those in Lemma 18. ◀

Proof of Lemma 20. First assume that $|C_0| = 1$. Then $|\Gamma'| = 1$ and $\mu \le 1$. 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 $|C_0| \ge 2$. Now, the proof of Lemma 20 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 afterwards 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 Γ'. The resulting power circuit will be denoted by $\tilde{\Gamma}$. We define

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

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 Γ' 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 18 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 csdrs 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 Γ' 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 Γ' 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 9), 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 TC^0: As $|C_0| \geq 2$, according to Proposition 14, we are able to decide in AC^0 whether the markings Λ_{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 TC^0 via its definition as above. Likewise I and the function λ can be computed in TC^0. Using Theorem 11, $\operatorname{CR}(\varepsilon(\Lambda_{P_i}) + 1)$ for $i \in I$ can be computed in AC^0 (since $|\tilde{C}_0| \leq 2 \cdot |\Gamma'|$) showing that altogether $\tilde{\Gamma}$ can be computed in 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 (Γ'', δ''). We start by defining

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

In order to obtain (Γ'', δ'') 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 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 14, we can check in 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 42 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 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 40, $\varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) + h$ 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 $|\tilde{C}_0| \leq 2\,|\Gamma'|$. With Theorem 11 we are able to calculate $\operatorname{CR}(\varepsilon(\Lambda_{\tilde{P}_i}|_{\tilde{C}_0}) + h)$ in AC^0.

Now we set $I = \left\{R^{(i,h)} \;\middle|\; d_i \neq 0, h \in [1 \,..\, d_i]\right\}$. According to Lemma 18 we are able to construct in TC^0 a reduced power circuit (Γ'', δ'') 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 Γ'', observe that during the whole construction, for every node $P_i \in \Gamma'$ we create at most μ 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 Γ'. Furthermore, notice that the only node of Γ' 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 Γ' we have introduced at most μ 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. ◀

Proof of Lemma 21. Consider again the equivalence relation $\sim_\varepsilon$ as defined above on $\Gamma'' \cup \operatorname{Min}(\Xi)$. For the equivalence class of a node $P \in \Gamma'' \cup \operatorname{Min}(\Xi)$ we write $[P]_\varepsilon$. We will define the marking $\widetilde{M}$ on Γ'' 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

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

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,

$$\begin{aligned} Z_{M,C} &\leq \mathrm{val}(\mathrm{digit}_C(M)) + |\mathrm{Min}(\Xi)| \cdot 2^k \\ &\leq \frac{1}{3} \cdot 2^{k+2} + 2^{k+\log(|\mathrm{Min}(\Xi)|)} && \text{(by Lemma 40)} \\ &\leq \frac{4}{3} \cdot \left(2^k + 2^{k+\log(|\mathrm{Min}(\Xi)|)} \right) \\ &\leq \frac{2}{3} \cdot 2^{k+\lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 2}. \end{aligned}$$

Thus, by Lemma 40, the digit-length of $\mathrm{CR}(Z_{M,C})$ is at most $k + \lceil \log(|\mathrm{Min}(\Xi)|) \rceil + 2 \leq h$. As an easy consequence of Proposition 14, the maximal chains can be determined in TC^0. Now, for every maximal chain C the (binary) number $Z_{M,C}$ can be computed in TC^0 using iterated addition and made be compact in AC^0 using Theorem 11. Thus, the marking $\tilde{M}_C$ can be computed in 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 17. 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 (Π, δ_Π) together with a marking M on Π. We start with some initial reduced power circuit (Γ_0, δ_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 Γ_i to Ξ_i),
- Γ_i is reduced,
- $\Gamma_{i-1} \leq \Gamma_i$ and $\Xi_i \subseteq \Xi_{i-1}$,
- $\varepsilon(M_i) = \varepsilon(M)$.

Moreover, 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 Γ_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 19) we compute a reduced power circuit (Γ_i', δ_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. Using ExtendChains (Lemma 20) with $\mu = \lceil \log(|\mathrm{Min}(\Xi_i)|)\rceil + 1$ we extend each maximal chain in (Γ_i', δ_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 20 is satisfied. The result of this step is denoted by (Γ_i'', δ_i'').
3. We apply UpdateMarkings (Lemma 21) 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 Γ_i'' are compact.
4. Each iteration ends by setting $\Gamma_{i+1} = \Gamma_i''$ and $\Xi_{i+1} = \Xi_i \setminus \mathrm{Min}(\Xi_i)$ and $M_{i+1} = \widetilde{M}_i$. Finally, δ_{i+1} is defined as δ_i'' on Γ_{i+1} and via the successor markings $\tilde{\Lambda}_P$ for $P \in \Xi_{i+1}$.

After exactly $\mathrm{depth}(\Pi) + 1$ 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 19, Lemma 20 and Lemma 21 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 Γ_{d+1} with $\varepsilon(M_{d+1}) = \varepsilon(M)$.

▷ Claim 43 (see Claim 22). Let $d = \mathrm{depth}(\Pi)$ and $\Gamma_0, \ldots, \Gamma_{d+1}$ be as constructed above. Then for all i we have $|\mathcal{C}_{\Gamma_i}| \leq |\Pi| + 1$ and $|\Gamma_i| \leq (|\Pi| + 1)^2 \cdot (\log(|\Pi|) + 2)$.

Proof. According to Lemma 19 and Lemma 20 we have $\left|\mathcal{C}_{\Gamma_{i+1}}\right| \leq |\mathcal{C}_{\Gamma_i}| + |\mathrm{Min}(\Xi_i)|$. Further observe that Π 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

$$\left|\mathcal{C}_{\Gamma_{i+1}}\right| \leq |\mathcal{C}_{\Gamma_i}| + |\mathrm{Min}(\Xi_i)| \leq 1 + \sum_{0\leq j\leq i} |\mathrm{Min}(\Xi_j)| \leq |\Pi| + 1. \tag{3}$$

Again by Lemma 19 and Lemma 20 we have

$$\begin{aligned}
|\Gamma_{i+1}| &\leq |\Gamma_i'| + \left|C_{\Gamma_i'}\right| \cdot (\lceil \log(|\mathrm{Min}(\Xi_i)|)\rceil + 1) && \text{(by Lemma 20)}\\
&\leq |\Gamma_i| + |\mathrm{Min}(\Xi_i)| + (|\mathcal{C}_{\Gamma_i}| + |\mathrm{Min}(\Xi_i)|) \cdot (\lceil \log(|\mathrm{Min}(\Xi_i)|)\rceil + 1) && \text{(by Lemma 19)}\\
&\leq |\Gamma_i| + |\mathrm{Min}(\Xi_i)| + (|\Pi| + 1) \cdot (\lceil \log(|\Pi|)\rceil + 1)\,. && \text{(by (3))}
\end{aligned}$$

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 19, Lemma 20 and Lemma 21 each iteration of the three steps above can be done in 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 TC^0, although Lemma 21 is stated only for a single marking. Now, the crucial observation is that, due to Claim 43, the input size for each iteration is polynomial in the original input size of (Π, δ_Π). Therefore, we can compose the individual iterations and obtain a circuit of polynomial size and depth bounded by $\mathcal{O}(D)$. Thus, we have described a $\mathsf{DepParaTC}^0$ circuit (parametrized by $\mathrm{depth}(\Pi)$) for the problem of computing a reduced form for (Π, δ_Π). This completes the proof of Theorem 17. ◀