BRANCHING PROGRAM UNIFORMIZATION, REWRITING LOWER BOUNDS, AND GEOMETRIC GROUP THEORY

IZAAK MECKLER

Mathematics Department
U.C. Berkeley
Berkeley, CA

Abstract. Geometric group theory is the study of the relationship between the algebraic, geometric, and combinatorial properties of finitely generated groups. Here, we add to the dictionary of correspondences between geometric group theory and computational complexity. We then use these correspondences to establish limitations on certain models of computation.

In particular, we establish a connection between read-once oblivious branching programs and growth of groups. We then use Gromov's theorem on groups of polynomial growth to give a simple argument that if the word problem of a group G is computed by a non-uniform family of read-once, oblivious, polynomial-width branching programs, then it is computed by an $O(n)$-time uniform algorithm. That is, efficient non-uniform read-once, oblivious branching programs confer essentially no advantage over uniform algorithms for word problems of groups.

We also construct a group EffCirc which faithfully encodes reversible circuits and note the correspondence between certain proof systems for proving equations of circuits and presentations of groups containing EffCirc. We use this correspondence to establish a quadratic lower bound on the proof complexity of such systems, using geometric techniques which to our knowledge are new to complexity theory. The technical heart of this argument is a strengthening of the now classical theorem of geometric group theory that groups with linear Dehn function are hyperbolic. The proof also illuminates a relationship between the notion of quasi-isometry and models of computation that efficiently simulate each other.

1. Introduction and our results

A group G equipped with a finite generating set A can be thought of as a metric space: the distance between $x, y \in G$ is the length of the shortest string w over the alphabet A with $x = wy$. Such a distance also defines a norm on G, given by the distance to the identity element e. Up to a constant factor, this metric does not depend on the choice of the generating set[24]. The study of groups from this metric point of view is called geometric group theory. It involves studying the relationship between a group's algebraic properties, geometric properties, and combinatorial properties which can be gleaned from a generating set.

There is a fairly well-known, though scattered, dictionary connecting concepts in geometric group theory with those in the theory of computing. A key entry in this dictionary which we make use of is the correspondence between NP-type certificates and disks embedded in groups' Cayley complexes. This correspondence is made precise by Birget, Ol'shanskii, Rips, and Sapir's[5] result that a finitely generated group G has word problem in NP iff G is a subgroup of a finitely presented group with polynomial isoperimetric function. Other entries include the theory of automatic groups[12] – that

E-mail address: izaak@berkeley.edu.

is, the relationship between a group's geometry and its word problem being encoded by a finite-state automaton, and Dehn's algorithm for the word problem in hyperbolic groups[14].

In this paper, we add two entries to the dictionary.

1.1. **Growth and branching programs.** Branching programs are a well-studied non-uniform model of space-bounded computation. Barrington's theorem[4] tells us that polynomial-length branching programs are exactly as powerful as NC^1, so there has been much study of weaker subclasses. Read-once oblivious branching programs (roOBPs) are one such subclass. An roOBP is essentially a non-uniform finite state machine, which is allowed to read each input symbol once in some fixed order (which may depend on the input length). Such a program is also sometimes called an ordered binary decision diagram, or OBDD[21]. Exponential lower bounds on natural functions have been known for this subclass for some time (see e.g., [3]). However, to our knowledge, there is not a general understanding of the capabilities of the class, and the relationship between non-uniform roOBPs and general uniform computation is not well-understood. Here, we improve understanding of these aspects of roOBPs.

A subset A of a group G is a *generating set* if every element of G can be written as a product of elements of A. G is *finitely generated* if it has a finite generating set. The word problem for (G, A) is the following decision problem: given a word w over the alphabet A, is w equal to the identity element of G? Our first result is a "uniformization" theorem, which states that for word problems, non-uniform roOBPs confer essentially no advantage over uniform algorithms.

Theorem 1. *Let (G, A) be a finitely generated group such that the word problem for (G, A) is solved by a polynomial width family of roOBPs. Then the word problem for (G, A) is solved by an $O(n)$-time multi-tape Turing machine.*

We will now give the main ideas of the proof. If (G, A) is a finitely generated group, we define $\rho_{G,A}(n)$ to be the number of elements of G which can be expressed as a word of length at most n over the alphabet A. The function $\rho_{G,A}(n)$ is called the *growth of G with respect to A*. The above theorem follows as a corollary of our result that the growth of (G, A) and the optimal size of an roOBP computing the word problem for (G, A) are commensurate:

Theorem 2. *If (G, A) is a finitely generated group and $W(n)$ is the minimum width of a family of roOBPs computing the word problem for (G, A), then*

$$\rho_{G,A}(n/4) \leq W(n) \leq \rho_{G,A}(n).$$

The upper bound is easy to obtain and the corresponding lower bound relies on the simple observation that any roOBP must have enough states to distinguish distinct elements in G because of the presence of inverses.

This bound then allows us to obtain a good deal of information about G. If $W(n)$ is bounded by a polynomial, then so too is $\rho_{G,A}(n)$. Gromov's theorem – a true gem of geometric group theory – then tells us that G is virtually nilpotent. Algorithmically, this is a heavy constraint: while the word problem for general finitely generated groups is undecidable, Building on work of Goodman and Shapiro[14], Holt and Rees[16] show that the word problem for a finitely generated, virtually nilpotent group can be solved in linear time on a multi-tape Turing machine, and the theorem follows.

Using a refinement of Gromov's theorem due to Shalom and Tao[27], we obtain a sharper uniformization result which allows us to assume that the word problem for (G, A) merely has a polynomial size roOBP for one sufficiently large input size.

1.2. **Rewriting systems on reversible circuits.** The second entry that we add to the dictionary connecting complexity with geometric group theory is a construction of a finitely generated group EffCirc whose elements encode boolean functions in a precise sense. A similar construction was given by Birget[7] as an example of a group with coNP-complete word problem. A distinguishing property of our construction is that the word norm approximately coincides with circuit size. We then use this group to prove a quadratic lower bound on the proof complexity of certain classes of proof systems for proving equations of circuits.

The easiest way of encoding boolean circuits using groups is to consider reversible circuits – that is, circuits in which all gates compute bijections. For a reversible circuit C on n bits, we will construct an element $\widehat{C} \in \mathsf{EffCirc}$ which encodes C. We construct $\widehat{C}$ as a word of length $O(|C| \log^2 n)$ in the generators of EffCirc, where $|C|$ is the number of gates in C.

We then use this group to prove a quadratic lower bound on a class of rewriting systems for proving equalities of reversible circuits. We call these systems generalized local transformation systems, augmenting the terminology of Iwama and Yamashita[17]. Such a lower bound is of interest as these rewriting systems are essentially universal proof systems: a super-polynomial lower bound would imply $\mathsf{NP} \neq \mathsf{coNP}$.

Making explicit the definition implicitly used in [17], a local transformation system is a collection of rewrite rules $C_1 \to C_2$, with the C_i two reversible circuits computing the same function. A generalized local transformation system (GLTS) allows the rewrite rules to replace portions of the circuits with arbitrary strings over an extended alphabet A, along with certain axioms ensuring soundness of the rewrite rules. A GLTS proves that a circuit computes the identity function if there is a sequence of rewrites taking that circuit to the empty string.

We prove the following lower bound on the proof complexity of such systems, stated informally here.

Theorem 3 (Informal)**.** *Let S be a GLTS over the alphabet A. Suppose that functions cannot be represented much more efficiently over the alphabet A than as circuits.*

Then there is some constant K and infinitely many circuits C computing the identity function – encoded as words w in EffCirc *– such that any sequence of rewrites proving w computes the identity function has length at least $K\,|w|^2$.*

The formal statement corresponding to this is Theorem 24. The proof has several key ingredients. The first is essentially the well-known fact that certain rewriting systems can be thought of as group presentations. In our situation, there is a simple correspondence associating to a GLTS S over an alphabet A a finitely presented group G_S with generating set A and which contains EffCirc. The next is the formalization of the hypothesis that circuits cannot be represented much more efficiently over A than as circuits as the requirement that the inclusion of EffCirc into G_S is a quasi-isometric embedding. It proceeds by using the connection between proof complexity and isoperimetric functions, as well as a novel refinement (Lemma 26) of the standard fact that any group with $o(n^2)$ isoperimetric function is hyperbolic[2].

We stress that the theorem states that there is *some* encoding $w \in \mathsf{EffCircGens}^*$ of C which requires on the order of $|w|^2$ rewrites; it does not guarantee that w is in the form of our chosen encoding $\widehat{C}$. However, since our encoding is close to optimal, we believe it may be possible to improve Lemma 26 to prove that there exist circuits C size s over n bits for which $\widehat{C}$ requires on the order s^2 many rewrites to rewrite to the empty string.

The hypothesis that the inclusion of EffCirc into G_S is a quasi-isometric embedding is surprisingly natural from a complexity-theoretic viewpoint. It states that a function cannot be represented more

succinctly as a word in A than as a reversible circuit (except up to constant factor). There is also a relationship between this hypothesis and the intermediate words over the alphabet A being – as circuits are – easy to compute with descriptions of functions. For example, if words over A represent circuits over a different basis gate set than the one used in the construction of $\mathsf{EffCirc}$, then φ is a quasi-isometric embedding.

More generally, if the elements of G_S admit an efficient notion of "evaluation", then the inclusion is close to being a quasi-isometric embedding. More precisely, let us define a family of *evaluation circuits* for G_S to be a family of circuits $E_{s,n}$ such that for any w of length s over the alphabet A, with $w =_{G_S} \varphi(\widehat{C})$ for C a circuit on n bits, we have $C(x) = E_{s,n}(w, x)$. By analogy with the fact that circuits themselves have evaluation circuits of size $\widetilde{O}(s+n)$[11], we can then show

Proposition 4. *If G_S admits a family of evaluation circuits $E_{s,n}$ of size $\widetilde{O}(s+n)$, then there is a constant C_0 so that for any $g \in \mathsf{EffCirc}$ encoding an n-bit circuit, $|g|_{\mathsf{EffCirc}}$ is $\widetilde{O}(|g|_{G_S} + n)$.*

This is close to φ being a quasi-isometric embedding, as that is equivalent to saying that $|g|_{\mathsf{EffCirc}}$ is $O(|g|_{G_S})$.

2. Preliminaries

We now give the necessary background and establish some notation.

2.1. **Branching programs.** Branching programs are a non-uniform model of space-bounded computation. A read-once, oblivious branching program (roOBP) B of width W, over an alphabet Σ, is a layered directed graph with $n+1$ layers of vertices along with a permutation $\pi : [n] \to [n]$. Each layer has at most w vertices. The vertices will also be referred to as *states*. For each vertex v in layer i and each $\sigma \in \Sigma$, there is an edge from v into layer $i+1$ labeled by σ. The first layer has one vertex, the "start" vertex. The vertices of the last layer are partitioned into two sets: "accepting vertices" and "rejecting" vertices.

An roOBP B accepts a word $w = w_1 \cdots w_n \in \Sigma^n$ if starting from the start vertex, the path labeled by $w_{\pi(1)}, \ldots, w_{\pi(n)}$ ends in an accepting vertex. Otherwise, B rejects w. We say L is computed by a family of width $W = W(n)$ read-once branching programs if there is a family of branching programs $B_0, B_1, \ldots$ such that B_n has width $W(n)$ and $w \in L$ iff $B_{|w|}$ accepts w.

As a corollary of Theorem 2, we also see that the word problem of any finitely generated group with exponential growth will have exponential size branching programs. This class of groups includes free groups, hyperbolic groups[30], the group $\mathsf{EffCirc}$ which we define, and many others.

2.2. **Group theoretic preliminaries.** 2 will denote the set $\{0, 1\}$. The identity element of a group will always be denoted by e. We will always assume that generating sets are symmetrized. That is, if $a \in A$, then $a^{-1} \in A$ as well. Moreover, we will assume that generating sets contain the identity element e. We will often call a pair (G, A) – with A a finite generating set of G – a finitely generated group.

Let (G, A) be a finitely generated group. Let A^* denote the strings over the alphabet A, A^n denote the strings of length exactly n, and $A^{\leq n}$ the strings of length at most n. For $w \in A^*$ let $|w|$ represent the length of w. An element of A^* will be called a *word*. By analogy to the identity element, the empty word will be denoted e as well.

There is a natural map $\mathsf{eval}_{A,G} \colon A^* \to G$ given by $\mathsf{eval}_{A,G}(a_1 \ldots a_k) = a_1 \cdot \cdots \cdot a_k$ where $\cdot$ denotes multiplication in G. When there is no confusion, we will write eval, omitting A and G. Also, when it is clear from context, we will omit explicit reference to eval, and think of a word $w \in A^*$ as being the element $\mathsf{eval}_{A,G}(w)$ of G. For two words w_1, w_2, we will use the notation

$w_1 =_G w_2$ when $\mathsf{eval}_{A,G}(w_1) = \mathsf{eval}_{A,G}(w_2)$. There is also a "formal inversion" on words given by $(a_1 \dots a_k)^{-1} = a_k^{-1} \cdots a_1^{-1}$.

Finally, we will denote by $\mathsf{Loops}_{G,A}(n)$ (or sometimes just $\mathsf{Loops}_G(n)$) the words $w \in A^*$ of length at most n with $w =_G e$, and by $\mathsf{Loops}_{G,A}$ (resp., Loops_G) the union $\bigcup_{n\in\mathbb{N}} \mathsf{Loops}_{G,A}(n)$.

For each element $g \in G$, we define the length of g with respect to A to be

$$|g|_A = \min \{ |w| : w \in A^*, \mathsf{eval}\,(w) = g \}$$

For two generating sets A, A', the functions $|-|_A$ and $|-|_{A'}$ differ by at most a constant factor, and thus we will sometimes write $|-|_G$, keeping the generating set implicit. When the group G is clear from context, we will simply write $|g|$. We also have a metric on G, called *the word metric*, defined by $d_G(g,h) = |hg^{-1}|_G$.

The Cayley graph of G with respect to A, denoted $\mathsf{Cay}(G,A)$, is the graph whose vertices are the elements of G and where there is an edge (g, ag) for each $g \in G$ and $a \in A$. The word metric defined above is also the "shortest path" metric for the Cayley graph of G. See [24] for more details.

Definition 5. A *presentation* of (G,A) is a set $R \subseteq A^*$ – called the set of relations – such that for every $w \in A^*$ with $\mathsf{eval}\,(w) = e$ we have for some $u_i \in A^*$ and $r_i \in R$

$$w = \prod_{i=1}^{N} u_i r_i u_i^{-1}$$

as elements of the free group on the letters A. A presentation $R = \{ r_1, \dots, r_k \}$ of (G,A) (where $A = \{ a_1, \dots, a_m \}$) is typically notated as $\langle a_1, \cdots a_m \mid r_1, \dots, r_k \rangle$. One may think of the presence of each r_i as encoding the equation $r_i = e$ in G.

Definition 6. For each $w \in \mathsf{Loops}_{G,A}$, define $\mathsf{area}_{G,A,R}\,(w)$ to be the minimal N such that w may be expressed as a product of N terms of the above form. It is called area because such an expression can be thought of as an embedding of a disk of simplicial area N, called a van Kampen diagram, into the Cayley complex of G. Van Kampen diagrams are also sometimes called fillings. See [1] for details.

Define the *Dehn function* of (G,A,R) by

$$\mathsf{dehn}_{(G,A,R)}(n) = \max_{w \in \mathsf{Loops}_{G,A}(n)} \mathsf{area}_{(G,A,R)}\,(w)$$

The Dehn function may be thought of as the maximal proof length required to prove any equation $w =_G e$ of length n. A function which upper bounds the Dehn function is called an isoperimetric function.

Definition 7. We define $\rho_{G,A}$, the *growth function of G with respect to A*, as $\rho_{G,A}(n) = |\mathsf{eval}\,(A^{\leq n})|$. That is, $\rho_{G,A}(n)$ is the number of elements in G which may be expressed as a word of length at most n. If A, A' are two finite generating sets of G, then $\rho_{G,A'} = \Theta(\rho_{G,A})$[24]. Thus, we will often keep the generating set implicit and write ρ_G.

Example 8 (Growth)**.** Here are several examples of groups with various group types.

(1) $\mathbb{Z}^k$ has growth function $\Theta(n^k)$.[24]
(2) The group $U_\ell(\mathbb{Z})$ of upper triangular matrices ℓ by ℓ with integer entries and 1s on the diagonal has growth function $\Theta(n^k)$ for some k[1]
(3) The free group on k generators has growth function $\Theta((2k-1)^n)$.[24]

[1]This is because it is nilpotent. The statement then follows from Gromov's theorem.

Definition 9. An inclusion $\varphi : (H, B) \to (G, A)$ of one finitely generated group into another is called a *quasi-isometric embedding* if there exists a constant C_0 for which $|g|_{G,A} \geq C_0 |g|_{H,B}$. That is, the most efficient encoding of a group element in the alphabet A is not more than a constant factor better than its encoding in B.

2.3. **The word problem.** The word problem for a finitely generated group (G, A) is the problem of deciding when a given word represents the identity element of G. That is, it is the decision problem associated to the set

$$\mathsf{WP}_{G,A} = \{ w \in A^* : \mathsf{eval}(w) = e \}.$$

The word problem for groups is a classical algorithmic problem introduced by Dehn[10]. Word problems serve as a fairly general model of computational phenomena: there are groups whose word problems are undecidable[25], NP-complete[6], coNP-complete[7], and complete for logspace-uniform NC^1 under logspace-uniform AC^0 reductions[26]. In fact, Birget showed[6] that every decision problem can be reduced by a one-to-one linear time reduction to the word problem of a group which has the same time complexity up to a linear factor. "Word problems" – broadly interpreted as deciding semantic equivalence of syntactic representations of algebraic objects – are ubiquitous in theoretical computer science. The problem UNSAT of deciding whether a boolean formula is unsatisfiable may be thought of as the word problem for the lattice of boolean functions, polynomial identity testing over a field $\mathbb{F}$ is the word problem in the ring $\mathbb{F}[x_1, x_2, \dots]$, and non-commutative polynomial identity testing is essentially the word problem in the free-skew ring over many variables[13].

3. Groups with word problem solved by efficient roOBPs

We will now proceed with the characterization of groups whose word problems are solved by polynomial-width families of roOBPs. It is straightforward to see that the following holds:

Proposition 10. *Let (G, A) be a finitely generated group with growth function $\rho_{G,A}$. Then $\mathsf{WP}_{G,A}$ is computed by a width $\rho_{G,A}$ roOBP.*

For input length n, one simply takes as states the $\rho_{G,A}(n)$-many elements of the group which can be written as a word of length at most n. A full proof appears in the appendix.

We now give the first technical contribution of this paper, which is an approximate converse to the above claim:

Theorem 11. *Let (G, A) be a finitely generated group with growth function $\rho_{G,A}$. If the word problem for G is solved by a width $W(4n)$ roOBP B for inputs of length $4n$, then $W(4n) \geq \rho_{G,A}(n)$.*

The proof of Theorem 11 is relatively straightforward, but due to space constraints we give the proof in the special case that the input symbols are read in order, which gives the general flavor. The idea of the proof is simply the following: after reading a word w, the branching program could read the word w^{-1}, which would cause it to accept. If $w \neq w'$ in G, then the branching program would not accept after reading w' and then w^{-1}. Thus, if $w \neq w'$ in a group G, then the states of the branching program after reading w and w' must be distinct. The general argument is reminiscent of, though distinct to, the argument that m-mixed boolean functions do not have read-once branching programs of width less than $2^m - 1$ (see [19]).

We now give this argument more precisely.

Claim 12. *Let (G, A) be a finitely generated group with growth function $\rho_{G,A}$. If the word problem for G is computed by a width W family of in-order, read-once branching programs B, then $\rho_{G,A}(n) \leq W(2n)$.*

Proof. Let $L_0, \dots L_{2n}$ be the layers of B_{2n}. Since $\rho_{G,A}(n) = \left|\mathsf{eval}\left(A^{\leq n}\right)\right|$ and $|L_n| \leq w(2n)$, to prove the claim, it suffices to exhibit an injection $q : \mathsf{eval}\left(A^{\leq n}\right) \to L_n$.

Let $x \in \mathsf{eval}\left(A^{\leq n}\right)$ and take $w \in A^{\leq n}$ such that $x = \mathsf{eval}\left(w\right)$. Since we assume $e \in A$, w may be padded to a word w_x of length exactly n with $\mathsf{eval}\left(w_x\right) = x$. Define $q(x)$ to be the state in L_n which B_{2n} enters upon reading w_x.

Let us show that f is injective. Suppose $q(x) = q(y)$. That is, the state $q(x) \in L_n$ corresponding to w_x equals the state $q(y)$ corresponding to w_y. Let $w = {w_x}^{-1}$ so that $|w| = n$ and $\mathsf{eval}\left(w\right) = x^{-1}$. Then $\mathsf{eval}\left(w_x w\right) = e$, so the the path beginning at $q(x)$ labelled by w terminates at an accepting vertex. Thus, since $q(x) = q(y)$, the path corresponding to $w_y w$ terminates at the same accepting vertex. This implies that $e = \mathsf{eval}\left(w_y\right)\mathsf{eval}\left(w\right) = yx^{-1}$. Multiplying both sides by x^{-1} yields $x = y$. Thus f is injective. □

Now, if $\mathsf{WP}_{G,A}$ is computed by a width $W(n)$ family of roOBPs with w a polynomial, then $\rho_G(n)$ is bounded by the polynomial $W(4n)$. A celebrated theorem of Gromov allows us to transform this coarse information into a precise algebraic characterization of G:

Theorem 13 (Gromov)**.** *If ρ_G is bounded by a polynomial, then G is virtually nilpotent*[15].[2]

As mentioned, the work of Goodman and Shapiro[14] and Holt and Rees[16] shows that virtually nilpotent groups have efficient uniform algorithms. Precisely,

Theorem 14 (Holt and Rees)**.** *Let (G, A) be a finitely generated, virtually nilpotent group. Then $\mathsf{WP}_{G,A}$ is solved by an $O(n)$-time multi-tape Turing machine.*

Taken together, these two theorems immediately imply the following corollary to Theorem 11:

Corollary 15. *Let (G, A) be a finitely generated group such that the word problem for (G, A) is solved by a polynomial width family of roOBPs. Then the word problem for (G, A) is solved by an $O(n)$-time multi-tape Turing machine.*

There is a wonderful refinement of Gromov's theorem due to Shalom and Tao[27] which states that in fact, polynomial growth at a single scale is sufficient to guarantee virtual nilpotence. In our situation, we thus obtain a corresponding refinement of Corollary 15:

Corollary 16. *Let $\{ B_n \}_{n\in\mathbb{N}}$ be a family of branching programs computing $\mathsf{WP}_{G,A}$.*

There is a constant C such that for any $k > 0$, if B_n has width at most n^k for a single $n \geq \exp\left(\exp\left(Ck^C\right)\right)$, then $\mathsf{WP}_{G,A}$ is computed by an $O(n)$-time multi-tape Turing machine.

4. Reversible circuits

Reversible circuits are a model for computing bijections $2^k \to 2^k$[28]. A gate computing a function $2^k \to 2^k$ is *reversible* if the function it computes is a bijection. A *reversible circuit* over a basis set U of reversible gates is simply a boolean circuit over U. We assume that a reversible circuit comes equipped with a topological ordering of its gates. We note that reversible circuits are also sometimes referred to as quantum boolean circuits[17].

[2]We omit the definition of virtual nilpotence, as we will use the concept purely as a black box. We direct interested readers to [23].

Note that since any reversible gate necessarily has the same number of inputs as outputs, a reversible circuit does as well. Moreover, since a reversible circuit arises as a composition of reversible gates, it will also compute a bijection. The *size* of a reversible circuit is number of gates it contains. Reversible circuits are a general model of computation and for any function $f\colon \mathbb{2}^n \to \mathbb{2}^m$ with a circuit of size s, there is a reversible circuit of size $O(s+m)$ computing the function $(x,z) \mapsto (x, f(x) \oplus z)$[29].

4.1. **Groups of circuits.** In order to properly define the rewriting systems of interest, we construct two finitely generated groups (Circ, CircGens) and (EffCirc, EffCircGens) in which words encode reversible circuits in a straightforward way.

Let $\mathbb{2}^{\mathbb{Z}}$ denote the set of maps from the integers $\mathbb{Z}$ to $\mathbb{2}$. Such a map may be thought of as a bi-infinite bitstring. Let $S(\mathbb{2}^{\mathbb{Z}})$ be the group of bijections $\mathbb{2}^{\mathbb{Z}} \to \mathbb{2}^{\mathbb{Z}}$, and let $(i;j)$ denote the element of $S(\mathbb{2}^{\mathbb{Z}})$ which swaps bit i with bit j. Finally, let U be the set of all $2^3!$ three bit reversible gates.

Definition 17. Let CircGens be the subset of $S(\mathbb{2}^{\mathbb{Z}})$ containing the followings maps:

(1) The right-shift t given by $t(b) = i \mapsto b(i-1)$.
(2) The map $(0;1)$ which swaps bits 0 and 1.
(3) For each gate $g \in U$, the map $\widehat{g}$ given by $\widehat{g}(\dots b_{-1}b_0b_1b_2b_3\dots) = \dots b_{-1}g(b_0b_1b_2)b_3\dots$.

and let Circ be the group generated by CircGens. It is straigthforward to show that any reversible circuit C may be represented by an element by an element of Circ.

Claim 18. *Let C be an n-bit reversible circuit of size s. Then there is a word $\widehat{C} \in \mathsf{CircGens}^*$ of length $O(sn)$ such that*

$$\widehat{C}(\dots b_{-1}b_0\dots b_{n-1}b_n\dots) = \dots b_{-1}C(b_0\dots b_{n-1})b_n\dots$$

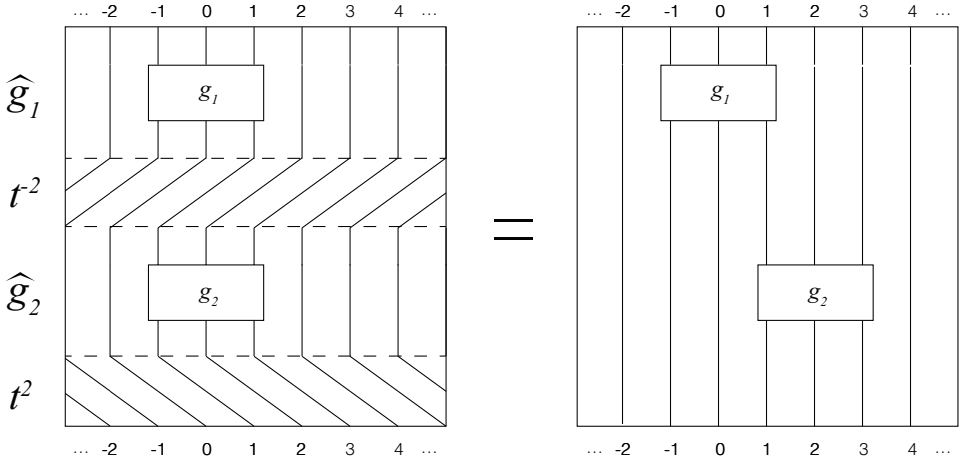

FIGURE 1. The encoding of a ciruit as a word in $\mathsf{CircGens}^*$.

The idea of the construction is to use the shift (and transpositions) to bring the desired wires down to the first three, where they can then be acted upon by a gate. The proof of this claim appears in the appendix.

The inefficiency of the above construction comes from the fact that shifting bits down into the first three bits requires quadratically long words. By using a simple trick of combinatorial group theory called an HNN extension[22], we can obtain the following more economical encoding:

Theorem 19. Circ *includes into a finitely generated group* (EffCirc, EffCircGens) *such that for any reversible circuit C of size s acting on n bits, the element $\widehat{C}$ described in Claim 18 can be written as a word of length $O(s \log^2 n)$.*

The construction of this group is in the appendix. We will treat it as a black box for the rest of the paper.

4.2. **Local transformation systems.** Circuit-UNSAT – the problem of deciding whether a circuit computes an identically zero function – is the prototypical coNP-complete problem. The corresponding problem for reversible circuits is the Circuit-Identity problem: given a reversible circuit, does it compute the identity function? Jordan showed that this problem is coNP-complete[18]. Thus, from the point of view of proof complexity, it is natural to investigate proof systems that prove that circuits compute the identity function.

Following Iwama and Yamashita[17], we will describe a class of proof systems termed *local transformation systems*. Iwama and Yamashita do not give a formal definition of local transformation system, but we give one now which generalizes their use of the term.

Definition 20. A *local transformation* is a rule $x \to y$, with x and y words in the generators of EffCirc. If x occurs as a subword of a word w, one calls the replacement that subword with y a *rewrite*.

A *local transformation system* (LTS) is a collection of local transformations S such that

(1) If $x \to y$ is in S, then so is $y \to x$.
(2) For every $a \in$ EffCircGens, S contains the rule $aa^{-1} \to e$.
(3) (Soundness and completeness) A circuit C computes the identity function iff there exists a sequence of rewrites taking $\widehat{C}$ to the empty string.

In group theoretic terms, an alternative, simpler definition is that an LTS is a presentation of EffCirc. We give the above definition to emphasize the way in which presentations of EffCirc may be thought of as rewriting systems for reversible circuits.

Iwama and Yamashita[17] constructed an explicit local transformation system. Although their list of rules is quite easy to describe, it is infinite, which makes it unsuited for the group theoretic tools we will use. To that end, we define the closely related notion of a *generalized local transformation system*.

Definition 21. Fix an alphabet A containing the generating set of EffCirc.

A *generalized local transformation* is a rule $x \to y$, with $x, y \in A^*$.

A *generalized local transformation system* (GLTS) is a finite collection of generalized local transformations S, such that

(1) If $x \to y$ is in S, then so is $y \to x$.
(2) For every $a \in A$, there is some $a^{-1} \in A$ such that S contains the rule $aa^{-1} \to e$.
(3) (Soundness and completeness) A circuit C computes the identity function iff there exists a sequence of rewrites taking $\widehat{C}$ to the empty string.

If there is a sequence of rewrites taking w to w', we write $w =_S w'$. In addition, for $w \in A^*$ with $w =_S e$, define the *proof complexity of* w, denoted $\mathsf{pc}_S(w)$, to be the the least number of rewrites required to take w to the empty string, not counting rewrites of the form $aa^{-1} \to e$ or $e \to aa^{-1}$. Define the *proof complexity of* S to be

$$\mathsf{PC}_S(n) = \max_{w \in \mathsf{EffCircGens}^{\leq n}, w =_S e} \mathsf{pc}_S(w).$$

Note that the maximum is only over words in the alphabet $\mathsf{EffCircGens}$ and not the larger alphabet A.

As with LTSs, there is a group theoretic interpretation of GLTSs. Let S be a GLTS over the alphabet A. We associate to S the finitely presented group $G_S = \langle A \mid xy^{-1}, (x \to y) \in S \rangle$. Once comfortable with the definitions, it is straightforward to prove the following proposition:

Proposition 22. *Given a GLTS S over an alphabet A, let $G_S = \langle A \mid xy^{-1}, (x \to y) \in S \rangle$. Then* $\mathsf{EffCirc}$ *embeds into G_S.*

Moreover, for $w \in \mathsf{EffCircGens}^*$, $\mathsf{area}_{G_S}(w) = \mathsf{pc}_S(w)$.

So, GLTSs are essentially finitely presented groups containing $\mathsf{EffCirc}$, since any such group also gives rise to a GLTS (Proposition 32).

We will be interested in the case where $\mathsf{EffCirc}$ is a quasi-isometrically embedded subgroup of G_S. We first verify that such GLTSs exist.

Claim 23. *There exists a GLTS S such that* $\mathsf{EffCirc}$ *is quasi-isometrically embedded in G_S and $\mathsf{PC}_S(n) = 2^{O(n^2)}$.*

As indicated above, G_S is closely tied to S complexity-wise as $\mathsf{area}_{G_S}(w) = \mathsf{pc}_S(w)$. However, it is important to note that in general $\mathsf{dehn}_{G_S}(n) \neq \mathsf{PC}_S(n)$. The former takes the maximum over all words in $A^{\leq n}$, while the latter only takes a maximum over the subset $\mathsf{EffCircGens}^*$. This presents a difficulty in using the theorem that groups with subquadratic Dehn function are hyperbolic directly, which we remedy by proving Lemma 26.

We exploit the relationship between area and proof complexity to obtain the following quadratic lower bound.

Theorem 24. *Let S be a GLTSs and suppose the inclusion $\varphi\colon \mathsf{EffCirc} \to G_S$ is a quasi-isometric embedding. Then $\mathsf{PC}_S(n)$ is not $o(n^2)$.*

The proof of this theorem is one of the primary technical contributions of this paper, as it uses mathematical tools which to our knowledge have yet to be applied to complexity theory. Let us motivate the requirement that the inclusion $\mathsf{EffCirc} \to G_S$ is a quasi-isometric embedding from a complexity theoretic point of view. First, we note that non-deterministic algorithms give rise to such rewriting systems.

Proposition 25. *If there is an $\mathsf{NTIME}(T(n))$ algorithm for solving the word problem of* $\mathsf{EffCirc}$*, then there is a GLTS S such that the inclusion $\mathsf{EffCirc} \to G_S$ is a quasi-isometric embedding and such that $\mathsf{PC}_S(n)$ is bounded by $O(n^2 T(n^2)^4)$.*

Proof. This follows immediately from the embedding theorem of [5], which states that given such an algorithm we obtain a group G in which $\mathsf{EffCirc}$ is quasi-isometrically embedded and whose Dehn function is $O(n^2 T(n^2)^4)$. Then, taking S to be the rewriting system associated to G (as in Proposition 32), we have $G_S = G$ and by Proposition 22, the proof complexity of S is bounded by the Dehn function of G. □

Second, we relate the notion of quasi-isometry to the condition of words in G_S being representations of functions which are as easy to compute with, as circuits are. Recall that we define a family of evaluation circuits for G_S to be a family of circuits $E_{s,n}$ such that if $w \in A^*$ is equal in G_S to $\widehat{C}$ for a circuit C of size s on n bits, then $E_{s,n}(w, x) = C(x)$ for every n-bit string x.

Proposition 4. *If G_S admits a family of evaluation circuits $E_{s,n}$ of size $\widetilde{O}(s+n)$, then there is a constant C_0 so that for any $g \in \mathsf{EffCirc}$ encoding an n-bit circuit, $|g|_{\mathsf{EffCirc}}$ is $\widetilde{O}(|g|_{G_S} + n)$.*

Proof. Let $\widehat{C} \in \mathsf{EffCirc}$ be the encoding of an n-bit circuit C. Let w be the string of length $s := |g|_{G_S}$ with $w =_{G_S} g$.

Let C_0 be the not necessarily reversible circuit $C_0(x) = E_{s,n}(w, x)$, with w hard-coded. C_0 has size $\widetilde{O}(s+n) + s = \widetilde{O}(s+n)$. By [29], we can turn C into a reversible circuit C_1 of size $O(\widetilde{O}(s+n)+n) = \widetilde{O}(s+n)$. Now, by Theorem 19, we have a word $\widehat{C_1}$ over $\mathsf{EffCircGens}$ of size $\widetilde{O}(s+n)\log^2 n = \widetilde{O}(s+n)$ representing C_1. Since C_1 computes the same function as C, and C computes g, this implies that $|g|_{\mathsf{EffCirc}}$ is $\widetilde{O}(s+n)$. □

This is substantially weaker than $\mathsf{EffCirc}$ being a quasi-isometry, which is equivalent to saying $|g|_{\mathsf{EffCirc}}$ is $O(|g|_{G_S})$ for all $g \in \mathsf{EffCirc}$ (whereas this proposition only proves the weaker upper bound for g which encode circuits). However, it does have a similar flavor and we believe illustrates the interplay between $\mathsf{EffCirc}$ as a computational object and as a geometric space.

To prove Theorem 24, we need several geometric lemmas involving the standard notion of *hyperbolicity*. In what follows, we will treat this notion as a black-box, so we defer its definition to the appendix, where we will need it.

Lemma 26. *Let $G = \langle A \mid R\rangle$ be a finitely presented group and H a quasi-isometrically embedded subgroup generated by $B \subseteq A$. Suppose there is some constant K such that*

$$\max_{w \in \mathsf{Loops}_H(n)} \mathsf{area}_G(w) \leq Kn.$$

Then H is hyperbolic.

The proof of this lemma (which appears in the appendix) is a refinement of a proof appearing in [2] of the fact that groups which themselves have linear Dehn function are hyperbolic.

Lemma 27. *Let $G = \langle A \mid R\rangle$ be a finitely presented group and H a quasi-isometrically embedded subgroup generated by $B \subseteq A$. Suppose that $\max_{w \in \mathsf{Loops}_H(n)} \mathsf{area}_G(w) = o(n^2)$. Then H is hyperbolic.*

Proof. Let $\mathsf{Cay}(H, B)$ be the Cayley graph of H with respect to B. and let Ω_H denote the set of simple loops (not necessarily based at e) in $\mathsf{Cay}(H, B)$. Let $\mathsf{area}_H : \Omega_H \to \mathbb{N}$ denote the restriction of area_G to $\mathsf{Cay}(H, B)$.

We would like to apply the main theorem of [8], which states that $\mathsf{area}_H(n) = O(n)$ if area_H satisfies the following two properties.

1 Take any $x, y \in H$ and let $\alpha_0, \alpha_1, \alpha_2$ be three paths in $\mathsf{Cay}(H, B)$ from x to y. Let γ_i be the path obtained by concatenating α_i with the reverse of α_{i+1} (taking subscripts mod 3). Then $\mathsf{area}_H(\gamma_2) \leq \mathsf{area}_H(\gamma_0) + \mathsf{area}_H(\gamma_1)$.

2 Suppose $\gamma \in \Omega_H$ is the concatenation of the four paths $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. Then there is some constant K_0 such that $\mathsf{area}_H(\gamma) \geq K_0 d_H(\alpha_1, \alpha_3) \cdot d_H(\alpha_2, \alpha_4)$, where $d_H(\alpha_i, \alpha_j)$ is the Hausdorff distance

$$d_H(\alpha_i, \alpha_j) = \max\left\{ \sup_{x_i \in \alpha_i} \inf_{x_j \in \alpha_j} d_H(x_i, x_j), \sup_{x_j \in \alpha_j} \inf_{x_i \in \alpha_i} d_H(x_i, x_j) \right\}.$$

Property (1) clearly holds since it holds more generally already for paths in the Cayley graph of G, as stated in [8]. Property (2) holds since as stated in [8], the analogous statement holds for d_G instead of d_H, and d_H is equivalent to d_G up to constant multiplicative error.

Thus, there is some K such that $\mathsf{area}_H(w) \leq K|w|$ for every loop w in $\mathsf{Cay}(H, B)$. By Lemma 26, we conclude that H is hyperbolic. □

We can now prove Theorem 24 without much difficulty.

Proof. Let S be a GLTSs and suppose and suppose that the inclusion $\varphi\colon \mathsf{EffCirc} \to G_S$ is a quasi-isometric embedding. Assume for a contradiction that $\mathsf{PC}_S(n) = o(n^2)$. Untangling the definition of PC_S, this says precisely that $\max_{w \in \mathsf{Loops}_{\mathsf{EffCirc}}(n)} \mathsf{area}_{G_S}(w) = o(n^2)$. Thus, by Lemma 27, $\mathsf{EffCirc}$ is hyperbolic. The finite subgroups of a hyperbolic group have bounded order[20]. $\mathsf{EffCirc}$, on the other hand, has arbitrarily large finite subgroups: for every n, it contains the group $A(2^n)$ of all even permutations on n-bits[9], which has order $\frac{(2^n)!}{2}$. Thus we have a contradiction, so PC_S is not $o(n^2)$. □

5. Discussion

It seems likely that the conclusion of Theorem 11 can be strengthened to hold for a read-once branching program which is not fixed order. That is, a branching program such that each variable is read at most once along any computation path. For read-many branching programs, no directly analogous statement can be proved: By Barrington's theorem, read-many branching programs can compute all of NC^1. NC^1contains the word problem of the free group on two generators, which has exponential growth. However, there is no immediate reason to rule out an analogous statement for read-twice or even read-k times branching programs.

Another interesting direction for future research is the fact that although proving that $\mathsf{EffCirc}$ does not quasi-isometrically embed into a group with polynomial Dehn function is equivalent to proving $\mathsf{NP} \neq \mathsf{coNP}$, there are, to our knowledge no immediate complexity theoretic implications of proving exponential lower bounds on the Dehn function of $\mathsf{EffCirc}$ itself. Put another way, it may be possible to show that any efficient rewriting system for circuits would have to use strings which are not circuits as intermediate stages in a rewrite sequence.

Finally, it seems very likely that the method of Lemma 26 can be strengthened to prove that for any GLTS S, there exist circuits C of size s such that the exact encoding $\widehat{C}$ (rather than just some encoding as we have shown) requires on the order of s^2 rewrites. It also seems likely that the condition that the inclusion $\mathsf{EffCirc} \to G_S$ is a quasi-isometric embedding can be weakened to the condition that the length of a word in $\mathsf{EffCirc}$ is only a polylogarithmic factor longer than in G_S. Taken together, these two facts along with Proposition 4 would then imply that for any S admitting a a family of evaluation circuits, there are circuits C of size s for which $\widehat{C}$ requires on the order of s^2 many rewrites to reduce to e.

Appendix A. Branching programs for word problems

First, let us show prove an upper bound on the width of a minimal branching program for computing the word problem in terms of the growth.

Proposition 10. *Let (G, A) be a finitely generated group with growth function $\rho_{G,A}$. Then $\mathsf{WP}_{G,A}$ is computed by a width $\rho_{G,A}$ roOBP.*

Proof. We describe the layers $L_0, \dots, L_n$ of an in-order branching program B computing $\mathsf{WP}_{G,A}$ over the alphabet A. Let $L_i = \mathsf{eval}\left(A^{\leq i}\right)$. L_0 consists of only the identity element of G, which will be the start vertex. The identity element of G is the sole accepting vertex in L_n and all other vertices are rejecting.

For each $g \in L_i$ and $a \in A$, it is clear that $ga \in L_{i+1}$: if $g = \mathsf{eval}(w)$ with $|w| = i$, then $ga = \mathsf{eval}(wa)$ and $|wa| = i + 1$. Thus, for each $g \in L_i$ and $a \in A$ our branching program will have the edge (g, ga) labelled by a.

Let us verify that B computes $\mathsf{WP}_{G,A}$. Let $w = a_1 \cdots a_n$ be a word of length n. We claim that for each $i \leq n$, the target of the path induced by $a_1 \cdots a_i$ is $\mathsf{eval}(a_1 \cdots a_i)$. For $i = 0$ this is clear as $a_1 \cdots a_0$ is the empty word, which evaluates to $e \in G$, the target of the empty path. Now suppose it holds for $i < n$. The path induced by $a_1 \cdots a_{i+1}$ is the path induced by $a_1 \cdots a_i$ followed by the edge labelled a_{i+1}. So, if g is the target of the path induced by $a_1 \cdots a_i$, the target of the path induced by $a_1 \cdots a_{i+1}$ is

$$\begin{aligned} g a_{i+1} &= \mathsf{eval}(a_1 \cdots a_i)\, a_{i+1} \\ &= \mathsf{eval}(a_1 \cdots a_i a_{i+1}) \end{aligned}$$

as desired.

Thus, the vertex in L_n induced by w is $\mathsf{eval}(w)$. If $\mathsf{eval}(w) = e$, then this is an accepting vertex, and it is rejecting otherwise. So B does compute $\mathsf{WP}_{G,A}$. Now, for each L_i, we have $|L_i| \leq |L_n| = \rho_G(n)$, so B has the claimed width. □

Now we will prove the matching lower bound by a similar argument to Claim 12.

Theorem 11. *Let (G, A) be a finitely generated group with growth function $\rho_{G,A}$. If the word problem for G is solved by a width $W(4n)$ roOBP B for inputs of length $4n$, then $W(4n) \geq \rho_{G,A}(n)$.*

Proof. Let $\pi : [4n] \to [4n]$ be the permutation specifying the order in which B reads the input symbols. Let $I = \{\, \pi(1), \dots, \pi(2n) \,\}$ and let $J = [4n] \setminus I = \{\, \pi(2n+1), \dots, \pi(4n) \,\}$.

Let k be the nth smallest element of I, and let

$$\begin{aligned} X_0 &= \{\, 1, \dots, k \,\} \\ X_1 &= \{\, k+1, \dots, 4n \,\} \end{aligned}$$

Since $J = (X_0 \cap J) \sqcup (X_1 \cap J)$, we may pick b so that $|X_b \cap J| \geq |J|/2 = n$. Note that we also have $|X_0 \cap I| = |X_1 \cap I| = n$.

Let L_{2n} be the $2n$th layer in the branching program. Similarly to the proof of Claim 12, we will exhibit an injection $q \colon \mathsf{eval}\left(A^{\leq n}\right) \to L_{2n}$. For each $x \in \mathsf{eval}\left(A^{\leq n}\right)$, let w_x be a word of length n with $\mathsf{eval}(w_x) = x$. If we provide an assignment of symbols to the positions in I, then we may run the machine B up to layer L_{2n}. Let $X_{1-b} \cap I = i_1 < \cdots < i_n$. Let u_x be the partial assignment $i_\ell \mapsto (w_x)_\ell$ and $i \mapsto e$ for $i \in I \setminus X_{1-b}$. In other words, we write w_x in order on $X_{1-b} \cap I$ and put identity symbols e everywhere else. Now let $q(x) \in L_{2n}$ be the state B enters after reading the partial assignment u_x.

Let us now show that q is an injection. Suppose $q(x) = q(y)$. We will extend the partial assignments u_x, u_y to total assignments u'_x, u'_y. To do so, we must provide values for the indices in J. We have $|X_b \cap J| \geq n$, so let $j_1, \ldots, j_n$ be the first n elements of $X_b \cap J$. Extend u_x to u'_x and u_y to u'_y by taking $j_\ell \mapsto ({w_x}^{-1})_\ell$ and $j \mapsto e$ for every other $j \in J$. That is, we write ${w_x}^{-1}$ in order on $X_b \cap J$ and put identity symbols e everywhere else.

Now u'_x has w_x written on X_{1-b} and ${w_x}^{-1}$ written on X_b, while u'_y has w_y written on X_{1-b} and ${w_x}^{-1}$ written on X_b. Whether $b = 0$ or $b = 1$, $\mathsf{eval}\,(u'_x) = e$. This is because $\mathsf{eval}\,(u_x) = \mathsf{eval}\left({w_x}^{-1}\right)\mathsf{eval}\,(w_x) = e$ if $b = 0$ and $\mathsf{eval}\,(u_x) = \mathsf{eval}\,(w_x)\,\mathsf{eval}\left({w_x}^{-1}\right) = e$ if $b = 1$.

Since u'_x agrees with u'_y on the final $2n$ symbols to be read and $q(x) = q(y)$, $\mathsf{eval}\left(u'_y\right) = \mathsf{eval}\,(u'_x) = e$. Thus, $yx^{-1} = e$, so $x = y$. This proves that the map q is indeed an injection, and the desired inequality follows. □

The theorem of Shalom and Tao[27] states

Theorem 28. *Let $B_0, B_1, \ldots$ be a family of branching programs computing $\mathsf{WP}_{G,A}$.*

There is an absolute constant C (not depending on (G, A)) such that for every finitely generated group (G, A) and $k > 0$, if there exists a single $n \geq \exp\left(\exp\left(Ck^C\right)\right)$ for which $\rho_{G,A}(n) \leq n^k$, then G is virtually nilpotent.

Thus, the weakened hypotheses give us the following corollary.

Corollary 16. *Let $\{ B_n \}_{n\in\mathbb{N}}$ be a family of branching programs computing $\mathsf{WP}_{G,A}$.*

There is a constant C such that for any $k > 0$, if B_n has width at most n^k for a single $n \geq \exp\left(\exp\left(Ck^C\right)\right)$, then $\mathsf{WP}_{G,A}$ is computed by an $O(n)$-time multi-tape Turing machine.

Appendix B. Reversible circuits

Here we provide the construction of Circ and EffCirc, the encodings of circuits as words in these groups, prove that the word problem for EffCirc is coNP-complete, and prove that GLTSs correspond to finite presentations of groups containing EffCirc.

Claim 18. *Let C be an n-bit reversible circuit of size s. Then there is a word $\widehat{C} \in \mathsf{CircGens}^*$ of length $O(sn)$ such that*

$$\widehat{C}(\ldots b_{-1}b_0 \ldots b_{n-1}b_n \ldots) = \ldots b_{-1}C(b_0 \ldots b_{n-1})b_n \ldots$$

Proof. Our strategy is as follows. First, we show that we can swap the ith bit with any of the first three bits with a word of length $O(i)$. Then, we will assemble the circuit gate by gate, swapping the correct bits into the first three, acting by an element of U, and then returning them.

Consider the map $\sigma = (0; 1)\,t$ illustrated in Figure 2. One can easily verify that $w_0(i) = \sigma^i\,(0; 1)\,\sigma^{-i}$ is equal to the map $(0; i + 1)$. The length of this word is clearly $O(i)$, and thus $O(n)$. By analogous constructions, we may construct a word $w_1(i)$ of length $O(n)$ representing $(1; i)$ and also a word $w_2(i)$ representing $(2; i)$.

Now let $g_1, \ldots, g_s$ be a topological ordering of the gates in C so that g_s is the final gate. For each g_i, let $x_{0,i}, x_{1,i}, x_{2,i}$ be the three bits that it acts on. Let $u_i = w_0(x_{0,i})w_1(x_{1,i})w_2(x_{2,i})$ and consider the map represented by $u_i\widehat{g_i}u_i$. This element of $S(2^{\mathbb{Z}})$ acts by moving the three input bits of g_i to the first three bits, acting by g_i, and then moving the three bits back. Thus, by taking the product

$$(u_s\widehat{g_s}u_s)(u_{s-1}\widehat{g_{s-1}}u_{s-1})\ldots(u_1\widehat{g_1}u_1)$$

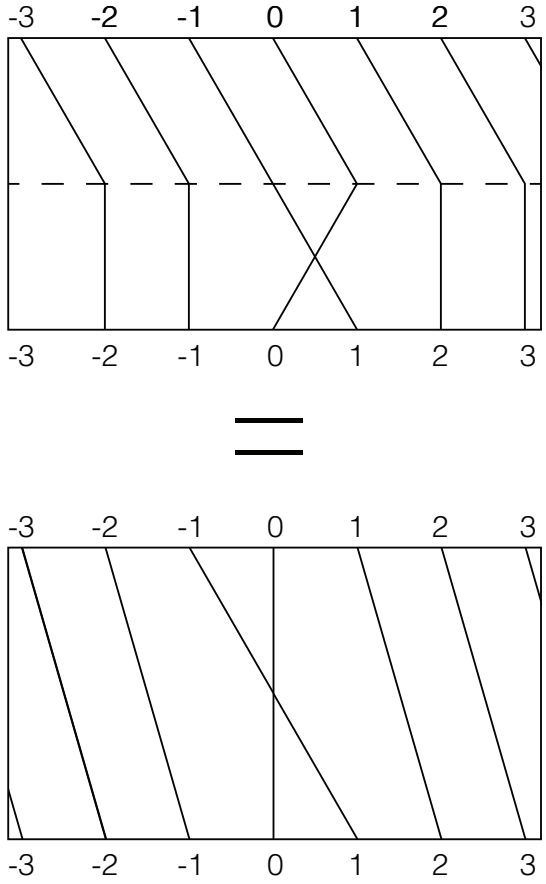

FIGURE 2. The map σ, which just skips over zero.

we obtain a word which represents the map

$$\widehat{C}(\dots b_{-1}b_0\dots b_{n-1}b_n\dots) = \dots b_{-1}C(b_0\dots b_{n-1})b_n\dots$$

and which has length $O(sn)$ since there are s terms, each of length $O(n)$. □

The inefficiency of the encoding in Claim 18 stems from the fact that we must use a word of length $O(n)$ to swap the bits each gate acts on into the first three bits. Let us show that it is possible to modify the encoding using a simple trick called an HNN extension[22] to reduce the length of the word representing that swap to $O(\log^2 n)$.

Proposition 29. Circ *embeds into a finitely presented group* $(\mathsf{EffCirc}, \mathsf{EffCircGens})$ *in which the map* $(0; n+1)$ *is represented by a word of length* $O(\log^2 n)$ *for every* $n \geq 0$.

Proof. Let $\sigma = (0;1)\,t$ as above. We have $(0; n+1) = \sigma^n\,(0;1)\,\sigma^{-n}$. Thus, to construct a more efficient encoding of $(0; n+1)$, we can construct a more efficient encoding of σ^n.

To this end, let $\mathsf{EffCirc}$ be the group obtained by adjoining an element q to Circ and quotienting out by the relation $q\sigma q^{-1} = \sigma^2$. In other words, we take

$$\mathsf{EffCircGens} = \{ \sigma \} \cup \{ \widehat{g} : g \text{ a three bit reversible gate} \}$$
$$\mathsf{EffCirc} = \langle A, q \mid R, q\sigma q^{-1}\sigma^{-2} \rangle$$

where R is the set of all words in A^* evaluating to e in Circ.

Now, we can use q to efficiently express σ^{2^k} for any k.

Claim 30. *For any $k \geq 0$, $q^k \sigma q^{-k} = \sigma^{2^k}$.*

Proof. The claim is clearly true for $k = 0$, supposing it holds for k, we have

$$\begin{aligned} q^{k+1}\sigma q^{-(k+1)} &= q^k q\sigma q^{-1} q^{-k} \\ &= q^k \sigma^2 q^{-k} \\ &= q^k \sigma q^{-k} q^k \sigma q^{-k} \\ &= \sigma^{2^k}\sigma^{2^k} \\ &= \sigma^{2^{k+1}} \end{aligned}$$

as desired. □

For an arbitrary power σ^n, we can use the above claim to express it as a word of length of length $O(\log^2 n)$ as follows: write $n = \sum_{i=0}^{\lfloor \log n \rfloor} a_i 2^i$ with $a_i \in \{ 0, 1 \}$. Now, we have

$$\begin{aligned} \prod_{i=0}^{\lfloor \log n \rfloor} \left(q^i \sigma q^{-i} \right)^{a_i} &= \prod_{i=0}^{\lfloor \log n \rfloor} \sigma^{a_i 2^i} \\ &= \sigma^n \end{aligned}$$

This is a product of at most $\log n$ terms, each of length at most $2 \log n + 2$, so we can express σ^n as a word of length $O(\log^2 n)$, and thus we can express $(0; n+1)$ as a word of length $O(\log^2 n)$. □

Theorem 19. Circ *includes into a finitely generated group* $(\mathsf{EffCirc}, \mathsf{EffCircGens})$ *such that for any reversible circuit C of size s acting on n bits, the element $\widehat{C}$ described in Claim 18 can be written as a word of length $O(s \log^2 n)$.*

Proof. This follows immediately from the construction of Claim 18 and the improved efficiency in Proposition 29. □

Claim 31. *The word problem for* $\mathsf{EffCirc}$ *is* coNP*-complete.*

Proof. By Theorem 19, we have a reduction from the $\mathsf{Circuit\text{-}Identity}$ problem to $\mathsf{WP}_{\mathsf{EffCirc}}$: for a given circuit C, the word $\widehat{C}$ obtained in the construction of Theorem 19 is the identity in $\mathsf{EffCirc}$ iff C computes the identity function. Thus $\mathsf{WP}_{\mathsf{EffCirc}}$ is coNP-hard.

It remains to show that $\mathsf{WP}_{\mathsf{EffCirc}} \in \mathsf{coNP}$. Here we will give an algorithm. Our goal will be to remove all occurrences of q from an input word w. If this is possible, then we can intepret w as an element of $S(2^{\mathbb{Z}})$ and universally check that w is the identity on a suitable set of bitstrings. If this is not possible, then we can show that $w \neq e$.

Let $w \in \mathsf{EffCircGens}^*$ and let $n = |w|$. First, assume w is freely reduced (this can clearly be done in polynomial time). As we manipulate w, we will represent it by compressing repeated occurences

of characters. E.g., the string $g_0qqqg_1g_1$ will be represented as $(g_0, 1)(q, 3)(g_1, 2)$, where integers are represented in binary.

Call a word in $\mathsf{EffCircGens}^*$ *reduced* if it does not contain a subsequence of the form $q\sigma^k q^{-1}$ or $q^{-1}\sigma^{2k}q$ for $k \in \mathbb{Z}$.

We can turn w into a reduced word by iteratively checking for and eliminating occurences of subwords of the above form. On one iteration, we do the following: Check if w contains a subword of the form $q\sigma^k q^{-1}$. If it does, replace it with σ^{2k}. Then check if w contains a subword of the form $q^{-1}\sigma^{2k}q$. If it does, replace it with σ^k. Neither of these replacements changes the group element represented by w. Moreover, each of these replacements decreases the number of occurences of q in w by 2, and so w will be reduced after at most $n/2$ iterations. Finally, each iteration increases the size of the representation of w by at most 1, so this can all be done in $\mathsf{poly}\,(n)$ time.

Britton's Lemma[22] says that if w is reduced and contains q, then $w \neq e$. So, now that w is reduced, if it contains q, then it is not equal to e and we reject. If it does not, then it is a word in the alphabet $\{\,\sigma\,\} \cup \{\,\widehat{g} : g \text{ a three bit reversible gate}\,\}$. It is clear that such a w can be interpreted as a map $S(2^{\mathbb{Z}}) \to S(2^{\mathbb{Z}})$. Let N be the highest power of σ now occurring in w. We have $N \leq 2^n$. Since w has at most n gates, it only inspects at most n bits. The indices of those bits necessarily lie in $[-N, N]$. and we can determine these indices by inspection of w. Now, universally choose an assignment to these bits, evaluate the map corresponding to w on these bits, and accept iff the output computed is the identity. □

Let us now demonstrate the correspondence between GLTSs and group presentations. First, we show that a group presentation can be thought of as a GLTS.

Proposition 32. *If $G = \langle A \mid R\rangle$ is a finitely presented group containing* $\mathsf{EffCirc}$ *(and* $\mathsf{EffCircGens} \subseteq A$*), then there is a GLTS S_G with proof complexity bounded by* dehn_G*.*

Proof. Let R be a finite set of relations for G. Now take S_G to consist of the rewrite rules

- $aa^{-1} \to e$ and $e \to aa^{-1}$ for $a \in A$.
- $r \to e$ for $r \in R$.

Let us show that S has the desired proof complexity. Take $w \in \mathsf{EffCircGens}^*$ with $w = e$. Now, there exist $r_1, \ldots, r_N \in R$ and $u_1, \ldots, u_N \in A_S^*$ with $N \leq \mathsf{dehn}_G(|w|)$ such that

$$w = \prod_{i=1}^{N} u_i r_i u_i^{-1}$$

as elements of the free group on the letters A.

Thus, we can perform a series of rewrites on w, introducing or removing subwords of the form uu^{-1}, so that $w = \prod_{i=1}^{N} u_i r_i u_i^{-1}$ on the nose as a string. Then, we simply apply the rewrites $r_1 \to e, \ldots, r_N \to e$ so that w becomes the string $\prod_{i=1}^{N} u_i u_i^{-1}$. Finally, we simply apply the rewrites $u_1 {u_1}^{-1} \to e, \ldots, u_N {u_N}^{-1} \to e$ to reduce w to the empty string. Not counting reductions of the form $uu^{-1} \to e$ or $e \to uu^{-1}$, we used $N \leq \mathsf{dehn}_G(|w|)$ rewrites, as desired. □

Now, we show that a GLTS can be thought of as a group presentation.

Proposition 22. *Given a GLTS S over an alphabet A, let $G_S = \langle A \mid xy^{-1}, (x \to y) \in S\rangle$. Then* $\mathsf{EffCirc}$ *embeds into G_S.*

Moreover, for $w \in \mathsf{EffCircGens}^*$, $\mathsf{area}_{G_S}(w) = \mathsf{pc}_S(w)$.

Proof. Let G_S be as stated. Let us show that two words $w_1, w_2 \in A^*$ are equal in G_S iff w_1 can be rewritten to w_2 in S. First, suppose w_1 equals w_2 in G_S. So, $w_1^{-1} w_2 =_{G_S} e$. This means that there

exist words $u_1, \ldots, u_N \in A^*$ and $r_1, \ldots r_N \in R$ such that

$$w_1^{-1} w_2 = \prod_{i=1}^{N} u_i r_i u_i^{-1}$$

or, multiplying by w_1,

$$w_2 = w_1 \prod_{i=1}^{N} u_i r_i u_i^{-1}.$$

Thus, to rewrite w_1 to w_2, it suffices to show that the empty string can be rewritten to $\prod_{i=1}^{N} u_i r_i u_i^{-1}$. Say $r_i = x_i y_i^{-1}$. Since S contains the rules $e \to aa^{-1}$ for each generator, we can rewrite $e \to u_i y_i y_i^{-1} u_i^{-1}$ by repeated application of these rules. Thus, e can be rewritten to $\prod_{i=1}^{N} u_i y_i y_i^{-1} u_i^{-1}$. Now, since $x_i \to y_i$ is in S for each i, we can rewrite this to

$$\prod_{i=1}^{N} u_i x_i y_i^{-1} u_i^{-1} = \prod_{i=1}^{N} u_i r_i u_i^{-1}.$$

as desired.

Conversely suppose that w_1 can be rewritten to w_2 in S. Say the sequence of intermediate words in the rewriting process are $w_1 = u_1, u_2, \ldots, u_N = w_2$, where each u_i differs from u_{i-1} from an application of a rewrite rule $x \to y$ in S. We have that $xy^{-1} =_{G_S} e$. Thus $x =_{G_S} y$, which means that $u_i =_{G_S} u_{i+1}$. Thus by induction, $w_1 =_{G_S} w_2$.

Now let us show that $\mathsf{EffCirc}$ embeds into G_S. Since $\mathsf{EffCircGens} \subseteq A$, we have a map $\Phi : \mathsf{EffCircGens}^* \to G_S$. In order for Φ to descend to a group homomorphism $\varphi : \mathsf{EffCirc} \to G_S$, it suffices to check that every $w \in \mathsf{EffCircGens}^*$ with w equal to e in $\mathsf{EffCirc}$ has $\Phi(w) = e$. Suppose we have such a w. Then since GLTSs are complete, w can be rewritten to the empty string e by S. Thus, by the above, $w =_G e$ as desired.

Now let's show that φ is an injection. This amounts to showing that the only element mapping to e is e. Suppose $\varphi(w) =_G e$. Then by the above, there is a sequence of rewrites in S taking w to e. Soundness of S then states that w is the identity element of $\mathsf{EffCirc}$. Thus $\mathsf{EffCirc}$ is a subgroup of G_S as desired.

The claim about proof complexity follows from the above argument relating sequences of rewrites with expression as a product of relations. □

Claim 23. *There exists a GLTS S such that* $\mathsf{EffCirc}$ *is quasi-isometrically embedded in G_S and* $\mathsf{PC}_S(n) = 2^{O(n^2)}$.

Proof. The coNP algorithm in Claim 31 for $\mathsf{WP}_{\mathsf{EffCirc}}$ can be turned into a $\mathsf{DTIME}\left(2^{O(n)}\right)$ algorithm, and hence trivially an $\mathsf{NTIME}\left(2^{O(n)}\right)$ algorithm. Proposition 25 then gives the desired GLTS. □

We now recall some standard definitions in geometric group theory. (See [24] for more details.) Let X be a geodesic metric space.

Definition 33. For points $x, y \in X$, $[xy]$ denotes a geodesic from x to y.

Definition 34. A triangle xyz in X is defined to be the union of geodesics $[xy], [yz]$, and $[zx]$.

Definition 35. A triangle xyz is r-thick if there is a point on the side $[xy]$ such that the minimal distance to a point on $[yz] \cup [zx]$ is at least $2r$.

Definition 36. X is hyperbolic if there is some r for which no triangle is r-thick.

Definition 37. Let $G = \langle A \mid R \rangle$ be a finitely presented group. Let R_* be the set of all cyclic permutations of elements of R and their inverses.

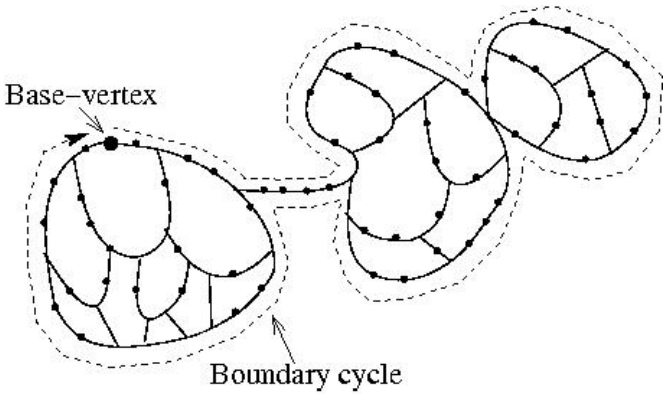

FIGURE 3. A van Kampen diagram.

A van Kampen diagram for G is a finite, planar cell complex D equipped with a base point p such that

(1) D is simply connected.
(2) Every 1-cell of D is oriented and labelled by some $a \in A$.
(3) For every 2-cell c of D and every vertex v on c, the word obtained by starting at v and reading the labels of the boundary of c in either of the two orders (and reading a as a^{-1} if an edge is negatively oriented) is a word in R_*.

For any van Kampen diagram D, we obtain a word ∂D (the boundary of D) by reading the boundary edges clockwise starting at p and, as above, reading reading a as a^{-1} if an edge is negatively oriented.

The diagram D is a combinatorial certificate of the fact that $\partial D =_G e$. A van Kampen diagram is illustrated in Figure 3. The key fact we will use about van Kampen diagrams is that for any word $w \in A^*$ with $w =_G e$, there is a van Kampen diagram D containing $\mathsf{area}_G(w)$ 2-cells and such that $\partial D = w$. See [22] for more details.

We now prove the key lemma that we needed to establish Theorem 24.

Lemma 26. *Let $G = \langle A \mid R \rangle$ be a finitely presented group and H a quasi-isometrically embedded subgroup generated by $B \subseteq A$. Suppose there is some constant K such that*

$$\max_{w \in \mathsf{Loops}_H(n)} \mathsf{area}_G(w) \leq Kn.$$

Then H is hyperbolic.

Proof. We essentially follow the proof of [2], departing from it to get around the fact that we have made weaker assumptions. In what follows, we abuse notation and refer to the Cayley graph of G or of H with respect to the given generating sets simply as G or as H.

H is quasi-isometrically embedded in G, so take $\delta > 0$ for which $|g|_G \geq \delta |g|_H$ for all $g \in H$. Also let ρ be the longest length of a relator in R. Suppose for a contradiction that H is not hyperbolic. Then for every r, H contains a triangle which is r-thick in H. In G, the triangle will be at least δr-thick, but since we can take r arbitrarily large, we may assume that we have a triangle in H which is r-thick in G.

Suppose $\epsilon > \rho$ and take $r > 6\epsilon$. Let xyz be a triangle in H which is r-thick, with $w \in [xy]$ a point which is distance at least $2r$ from the other two sides. As in [2], by pinching off degeneracies, we may also assume that xyz is non-degenerate (i.e., the sides intersect only at the vertices and no two vertices lie on the same geodesic).

Cutting off the corners of xyz such that the truncated sides are all distance at least 4ϵ from each other and the cut-off segments (which we take to be geodesics in G) are of length exactly ϵ.

The resulting figure F falls into one of three cases:

(1) A non-degenerate hexagon with three sides of length 4ϵ
(2) A non-degenerate quadrilateral with two sides of length 4ϵ.
(3) A degenerate hexagon.

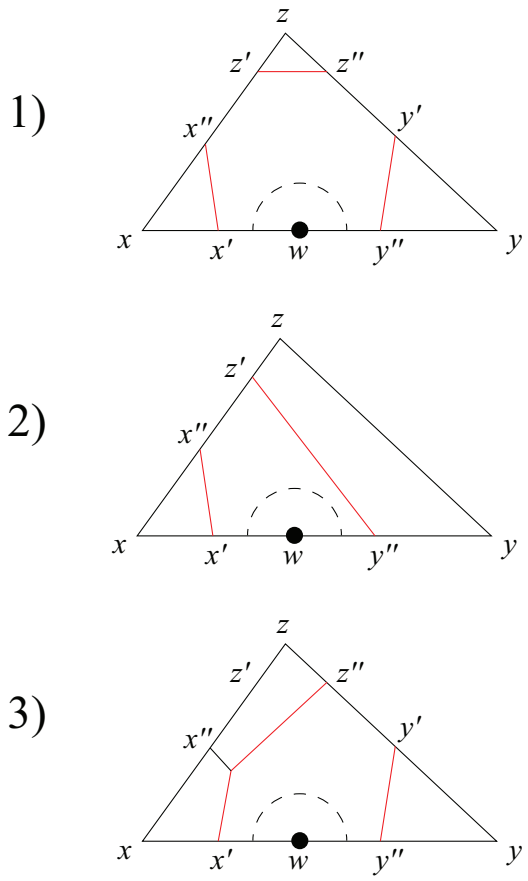

FIGURE 4. The figure F

The three cases are illustrated in Figure 4. Let α, β, γ be the lengths of the three truncated sides of xyz (the black sides in Figure 4).

We need a lemma, whose proof we defer.

Lemma 38. *There is some constant C depending only on ϵ such that*

$$\mathsf{area}_G(F) > \delta(\alpha + \beta + \gamma)\epsilon/\rho^2 - C + r/\rho.$$

We would like to contradict this lower bound by placing a smaller upper bound on $\mathsf{area}_G(F)$. To do so, we need to make use of our isoperimetric inequality for curves in H. The trouble is that the red sides of F in Figure 4 may be not be curves in H, but only in G. So, we will compare the area of

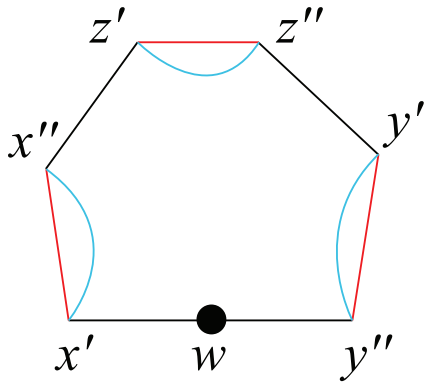

FIGURE 5. Distances in H may be larger than in G.

F to a figure F' whose boundary is a path in H. The red arcs as indicated in Figure 4 are geodesics in G. Let F' be the figure obtained by replacing these arcs with the corresponding geodesics in H, which may in general be longer by a factor of at most $1/\delta$. The situation is illustrated for case (1) in Figure 5, with the geodesics in H appearing in blue and those in G appearing in red.

Adding up all the side lengths and accounting for the at most $1/\delta$ lengthening when changing a geodesic in G to a geodesic in H, the perimeter of F' is at most

$$(\alpha + \beta + \gamma) + 12\epsilon/\delta.$$

Thus, since the perimeter of F' is now a loop in H rather than G, we can use our isoperimetric inequality to conclude that

$$\mathsf{area}_G(F') \leq (\alpha + \beta + \gamma)K + 12K\epsilon/\delta$$

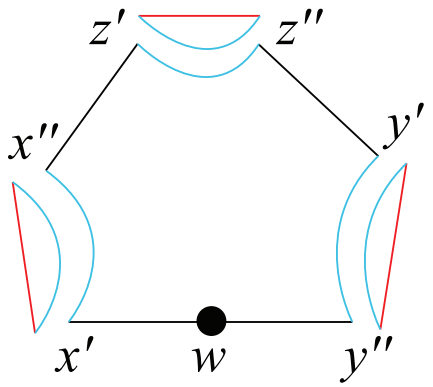

FIGURE 6. A decomposition of F into 4 loops.

Now we must relate $\mathsf{area}_G(F')$ to $\mathsf{area}_G(F)$. Consider the decomposition of F illustrated in Figure 6 into F' and three loops each of length at most $2\epsilon/\delta$. The area of each loop is thus bounded

by $\mathsf{dehn}_G(2\epsilon/\delta)$. Then since we can compose together fillings for each of these loops to get a filling for F, we have

$$\begin{aligned}\mathsf{area}_G(F) &\leq \mathsf{area}_G(F') + 3\mathsf{dehn}_G(2\epsilon/\delta) \\ &\leq (\alpha+\beta+\gamma)K + 12K\epsilon/\delta + 3\mathsf{dehn}_G(2\epsilon/\delta).\end{aligned}$$

Combining this with our lower bound yields

$$\delta(\alpha+\beta+\gamma)\epsilon/\rho^2 - C + r/\rho < (\alpha+\beta+\gamma)K + 12K\epsilon/\delta + 3\mathsf{dehn}_G(2\epsilon/\delta).$$

Now, taking $\epsilon = K\rho^2/\delta$, the $\alpha+\beta+\gamma$ terms cancel, leaving us with

$$r/\rho < 12K\epsilon/\delta + 3\mathsf{dehn}_G(2\epsilon/\delta) + C.$$

Since ϵ is fixed, the right hand side is constant, so taking r large enough yields a contradiction. □

It remains to prove Lemma 38, which we do now.

Lemma 38. *There is some constant C depending only on ϵ such that*

$$\mathsf{area}_G(F) > \delta(\alpha+\beta+\gamma)\epsilon/\rho^2 - C + r/\rho.$$

Proof. We prove the lemma in case (1). The proof in the other cases is essentially the same. See Lemma 2.7 of [2] for more details. Label the points on F as in Figure 4. Let D be a van Kampen diagram with boundary ∂F. For θ a path in G, let $\ell(\theta)$ denote the number of edges in θ.

If T is a subcomplex of D, define $\mathsf{star}(T)$ to be the set of all cells in D which intersect T. If θ is one of the black curves in F ($[x''z']$, $[z''y']$, or $[x'y'']$), let $\mathsf{star}^k(\theta)$ be the subcomplex of D obtained by iterating star k times times starting with θ and let $N(\theta) = \mathsf{star}^{\lfloor\epsilon/\rho\rfloor+1}$.

Claim 39. *There is some C_1 depending only on ϵ such that the number of 2-cells in $N(\theta)$ is at least $\delta\ell(\theta)\epsilon/\rho^2 - C_1$.*

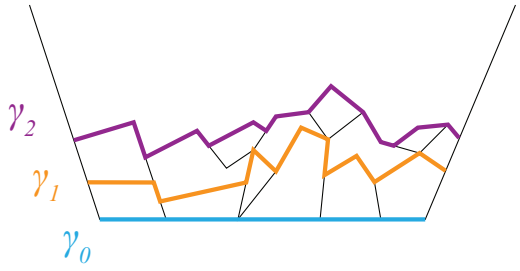

FIGURE 7. The curves γ_k.

Proof. Let γ_k denote the curve running along the bottom of $D \setminus \mathsf{star}^k(\theta)$, as illustrated in Figure 7. Formally, letting D^1 denote the 1-skeleton of D, γ_k is $\mathsf{star}^k(\theta) \cap D^1$ intersected with the closure of $\mathrm{int}(D) \setminus \mathsf{star}^k(\theta)$, where $\mathrm{int}(D)$ is the interior of D.

For each k, there is a path α_k from p to q which travels a path L_k up the left side of F, traverses γ_k, then a path R_k down the right side of F. Since the paths L_k and R_k touch at most k 2-cells, they each have length at most $k\rho$.

The distance from p to q is at least $\delta\ell(\theta)$ (as θ is a geodesic in H). Thus, since α_k is a path from p to q, we have

$$\begin{aligned}\delta\ell(\theta) &\leq \ell(\alpha_k)\\ &= \ell(\gamma_k) + \ell(L_k) + \ell(R_k)\\ &\leq \ell(\gamma_k) + 2k\rho.\end{aligned}$$

And so, $\ell(\gamma_k) \geq \delta\ell(\theta) - 2k\rho$. Thus, since each edge of γ_k lies on a 2-cell in $\mathsf{star}^k(\theta) \setminus \mathsf{star}^{k-1}(\theta)$ and each such 2-cell has at most ρ edges, the total number of 2-cells in $\mathsf{star}^{k+1}(\theta) \setminus \mathsf{star}^k(\theta)$ is at least

$$(\delta\ell(\theta) - 2k\rho)/\rho$$

Thus, the number of two cells in $N(\theta)$ is at least

$$\frac{1}{\rho}\sum_{k=0}^{\lfloor\epsilon/\rho\rfloor} \delta\ell(\theta) - 2k\rho \geq \delta\ell(\theta)\epsilon/\rho^2 - 2\epsilon^2/\rho^2$$

and the claim follows with $C_1 = 2\epsilon^2/\rho^2$. □

Let us now show that $N(\theta_1)$ and $N(\theta_2)$ are disjoint if $\theta_1 \neq \theta_2$, where $\theta \in \{\, [x'y''], [x''z'], [z''y'] \,\}$. Every point in $N(\theta_i)$ is distance at most $(\lfloor\epsilon/\rho\rfloor + 1)\rho < 2\epsilon$ from θ_i. Thus, if there were a point common to θ_1 and θ_2, the distance between θ_1 and θ_2 would be less than 4ϵ. However, F was constructed specifically so that θ_1 was distance at least 4ϵ from θ_2. Thus the two sides are disjoint, so their contributions to the area of F are distinct.

Let p be a point on the boundary of $N([x'y''])$ distance at most ϵ from w and q a point on the boundary of $N([x''z'])$. Let γ_1 be a geodesic in D from w to p, γ_2 a path in D from p to q not entering $N(\theta)$ for any θ, and γ_3 a path of length at most ϵ from q to $[x''z']$. Let γ be the concatenation of these three paths. Since w is distance at least r from $[x''z']$, we have

$$\begin{aligned}r &\leq \ell(\gamma)\\ &= \ell(\gamma_1) + \ell(\gamma_2) + \ell(\gamma_3)\\ &\leq \ell(\gamma_2) + 2\epsilon.\end{aligned}$$

Thus, $\ell(\gamma_2) \geq r - 2\epsilon$, which means there at least $(r - 2\epsilon - 2)/\rho$ 2-cells in D outside of any $N(\theta)$.

Adding up these 2-cells with those contributed by the $N(\theta)$ yields at least

$$\delta(\alpha + \beta + \gamma) + r/\rho - 3C_1 - (2\epsilon + 2)/\rho$$

2-cells in D. This concludes the proof, taking $C = 3C_1 + (2\epsilon + 2)/\rho$. □

References

[1] *Van Kampen Diagrams and Pictures*, pages 163–177. Birkhäuser Basel, Basel, 2007.

[2] J.M. Alonso, T. Brady, D. Cooper., V. Ferlini, M. Lustig, M. Mihalik, M. Shapiro, and H. Short. Notes on word hyperbolic groups. `http://math.hunter.cuny.edu/olgak/hyperbolic%20groups/MSRInotes2004.pdf`, 1990.

[3] László Babai, Péter Hainal, Endre Szemerédi, and Gy˙A lower bound for read-once-only branching programs. *Journal of Computer and System Sciences*, 35(2):153 – 162, 1987.

[4] D A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pages 1–5, New York, NY, USA, 1986. ACM.

[5] J.-C. Birget, A. Yu Ol'shanskii, E. Rips, and M. V. Sapir. Isoperimetric functions of groups and computational complexity of the word problem. *Annals of Mathematics*, 156(2):467–518, 2002.

[6] Jean-Camille Birget. Reductions and functors from problems to word problems. *Theoretical Computer Science*, 237(1):81 – 104, 2000.

[7] Jean-Camille Birget. Circuits, the groups of richard thompson, and conp-completeness. *International Journal of Algebra and Computation*, 16(01):35–90, 2006.

[8] B. H. Bowditch. A short proof that a subquadratic isoperimetric inequality implies a linear one. *Michigan Math. J.*, 42(1):103–107, 1995.

[9] Tim Boykett, Jarkko Kari, and Ville Salo. *Strongly Universal Reversible Gate Sets*, pages 239–254. Springer International Publishing, Cham, 2016.

[10] M. Dehn. Über unendliche diskontinuierliche gruppen. (mit 5 figuren im text). *Mathematische Annalen*, 71:116–144, 1912.

[11] Ryan Williams Dylan McKay. Small circuits for circuit evaluation problem. Theoretical Computer Science Stack Exchange. URL:http://cstheory.stackexchange.com/q/37898 (version: 2017-04-04).

[12] Benson Farb. Automatic groups: A guided tour. *L'Enseignement Math.*, 38:291–313, 1992.

[13] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 109–117, Oct 2016.

[14] Oliver Goodman and Michael Shapiro. On a generalization of dehn's algorithm. *International Journal of Algebra and Computation*, 18(07):1137–1177, 2008.

[15] Michael Gromov. Groups of polynomial growth and expanding maps (with an appendix by jacques tits). *Publications MathÃľmatiques de l'IHÃLS*, 53:53–78, 1981.

[16] Derek F. Holt and Sarah Rees. Solving the word problem in real time. *J. London Math. Soc*, 2:623–639, 1999.

[17] Kazuo Iwama and Shigeru Yamashita. Transformation rules for cnot-based quantum circuits and their applications. *New Generation Computing*, 21(4):297–317, 2003.

[18] Stephen P. Jordan. Strong equivalence of reversible circuits is conp-complete. *Quantum Info. Comput.*, 14(15-16):1302–1307, November 2014.

[19] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer Publishing Company, Incorporated, 2012.

[20] S. O. Juriaans, I. B. S. Passi, and Dipendra Prasad. Hyperbolic unit groups. *Proceedings of the American Mathematical Society*, 133(2):415–423, 2005.

[21] Heh-Tyan Liaw and Chen-Shang Lin. On the obdd-representation of general boolean functions. *IEEE Transactions on Computers*, 41(6):661–664, Jun 1992.

[22] Roger C. Lyndon and Paul E. Schupp. *Combinatorial Group Theory*. Springer, 2001.

[23] Vipul Naik. Virtually nilpotent group - groupprops. `https://groupprops.subwiki.org/wiki/Virtually_nilpotent_group`. Accessed: 2017-04-01.

[24] Piotr W. Nowak and Guoliang Yu. *Large Scale Geometry*. European Mathematical Society, 2012.

[25] A.Yu. Ol'shanskii and Mark V. Sapir. Groups with undecidable word problem and almost quadratic dehn function. *Journal of Topology*, 5(4):785–886, 2012.

[26] David Hill Robinson. *Parallel Algorithms for Group Word Problems*. PhD thesis, La Jolla, CA, USA, 1993. UMI Order No. GAX93-17153.

[27] Yehuda Shalom and Terence Tao. A finitary version of gromov's polynomial growth theorem. *Geometric and Functional Analysis*, 20(6):1502–1547, 2010.

[28] Tommaso Toffoli. Reversible computing. In *Proceedings of the 7th Colloquium on Automata, Languages and Programming*, pages 632–644, London, UK, UK, 1980. Springer-Verlag.

[29] Umesh Vazirani. Cs 191 lecture notes: Reversible computation. `http://www-inst.eecs.berkeley.edu/~cs191/sp12/notes/reversible.pdf`, 2012.

[30] Xiangdong Xie. Growth of relatively hyperbolic groups. *Proceedings of the American Mathematical Society*, 135(3):695–704, 2007.