5-1-2016

Thompson's group F is 1-counter graph automatic

Murray Elder
The University of Newcastle, Australia

Jennifer Taback
Bowdoin College

Research rticle

Murray Elder and Jennifer Taback*

Thompson’s group is 1-counter graph automatic

DOI: 10.1515/gcc-2016-0001
Received January 18, 2015

bstract: It is not known whether Thompson’s group F is automatic. With the recent extensions of the notion of an automatic group to graph automatic by Kharlampovich, Khoussainov and Miasnikov and then to -graph automatic by the authors, a compelling question is whether F is graph automatic or -graph automatic for an appropriate language class . The extended definitions allow the use of a symbol alphabet for the normal form language, replacing the dependence on generating set. In this paper we construct a 1-counter graph automatic structure for F based on the standard infinite normal form for group elements.

Keywords: Thompson’s group F, automatic group, graph automatic group, -graph automatic group

MSC 2010: 20F65, 68Q45

1 Introduction

The notion of an automatic group was introduced in the 1990s based on ideas of Thurston, Cannon, Gilman, Epstein and Holt in the hopes of categorizing the fundamental groups of compact 3-manifolds. s not all nilpotent groups proved to be automatic, the definition required expansion to make it more robust. simultaneous motivation behind this definition was to use the automatic structure to ease computation within these groups. To that end, it was shown that automatic groups are finitely presented, have word problem solvable in quadratic time and possess at most a quadratic Dehn function [9]. In an automatic group there is a natural set of quasigeodesic normal forms which define the structure, and this normal form representative for any group element can always be found in quadratic time.

It is not known whether Thompson’s group F is automatic. Guba and Sapir present a regular normal form for elements of F and prove that the Dehn function of F is quadratic in [11]. The word problem in F is solvable in $O\ n \log n)$ time (see, for example, [13]) and it is shown in [2] that F is of type FP_∞. However, no one has been able to prove that the group is (or is not) automatic. In [6] it is shown that F cannot have a regular combing by geodesics with respect to the standard finite generating set, while in [5] it is shown that F admits a tame 1-combing by quasigeodesics for this generating set, with a linear tameness function.

In [12] Kharlampovich, Khoussainov and Miasnikov expand the definition of an automatic group to that of a graph automatic group by allowing the normal forms for elements to be expressed in terms of a symbol alphabet, rather than the group generators. The precise definition is given below in Section 2.1. Groups which are not automatic, but are graph automatic with this definition include the Baumslag–Solitar groups and finitely generated groups of nilpotency class 2 (see [1, 12]). Graph automatic groups retain the properties that the word problem is solvable in quadratic time, and the normal form for any group element is computable in quadratic time.

Murray Elder: School of Mathematical and Physical Sciences, The University of Newcastle, Callaghan NSW 2308, ustralia, e-mail: murrayelder@gmail.com
***Corresponding author: Jennifer Taback:** Department of Mathematics, Bowdoin College, Brunswick, ME 04011, US , e-mail: jtaback@bowdoin.edu

The concept of a graph automatic group was extended by the authors in [8] to a -graph automatic group, where all languages comprising the structure are required to be in the language class . This definition is stated precisely in Section 2.1.

The introduction of a symbol alphabet makes F a natural candidate for the class of graph (or -graph) automatic groups, as there are two natural symbol alphabets associated to this group.

(1) The first symbol alphabet is based on the standard infinite normal form for elements of F, and in this paper we prove that there is a language of quasi-geodesic normal forms over this alphabet which forms a 1-counter graph automatic structure for F. Moreover, F is not graph automatic with respect to this language of normal forms.

(2) The second symbol alphabet is based on the combinatorial "caret types" of nodes in the reduced pair of trees representing a given element. In [14], Younes and the second author define a language of quasi-geodesic normal forms for elements of F based on this alphabet which they show to be the basis of a 3-counter graph automatic structure for F, although they note that the number of counters can be reduced to 2. They also show that F is not graph automatic with respect to this language of normal forms.

Neither of these results rules out the possibility that F is graph automatic with respect to a different normal form, or alternate symbol alphabet.

We remark that a third possibility for constructing a -graph automatic structure for F would be to use the regular normal form language of Guba and Sapir [11]. In preparing this article we considered the complexity of the multiplier automata for this structure and found that the multiplier automaton for the generator x_1 would require at least four counters, and hence did not include the details in this article.

This paper is organized as follows. In Section 2 we recall the definition of a -graph automatic group introduced in [8] which generalizes the definition of a graph automatic group introduced by Kharlampovich, Khoussainov and Miasnikov in [12], and present a brief introduction to Thompson's group F. Section 3 is devoted to describing a language of normal forms based on the standard infinite normal form for elements of F, and the resulting 1-counter graph automatic structure.

2 Background

We present some background material on -graph automatic groups as well as a brief introduction to Thompson's group F.

2.1 Generalizations of automaticity

The following definitions are taken from [8] and include the definition of a graph automatic group given in [12]. We begin with the definition of a convolution of strings, following [12].

Let G be a group with symmetric generating set X, and Λ a finite set of symbols. In general we do not assume that X is finite. The number of symbols (letters) in a word $u \in \Lambda^*$ is denoted $|u|_\Lambda$.

Definition 2.1 (Convolution, [12, Definition 2.3]). Let Λ be a finite set of symbols, $\diamond$ a symbol not in Λ, and let $L_1, \dots, L_k$ be a finite set of languages over Λ. Set $\Lambda_\diamond = \Lambda \cup \{\diamond\}$. Define the *convolution of a tuple* $w_1, \dots, w_k) \in L_1 \times \dots \times L_k$ to be the string $\otimes\, w_1, \dots, w_k)$ of length $\max |w_i|_\Lambda$ over the alphabet $\Lambda_\diamond)^k$ as follows. The i-th symbol of the string is

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix},$$

where λ_j is the ith letter of w_j if $i \le |w_j|_\Lambda$ and $\diamond$ otherwise. Then

$$\otimes\, L_1, \dots, L_k) = \{\otimes\, w_1, \dots, w_k) \mid w_i \in L_i\}.$$

We note that the convolution of regular languages is again a regular language.

s an example, if $w_1 = aa$, $w_2 = bbb$ and $w_3 = a$, then

$$\otimes\, w_1, w_2, w_3) = \begin{pmatrix} a \\ b \\ a \end{pmatrix}\begin{pmatrix} a \\ b \\ \diamond \end{pmatrix}\begin{pmatrix} \diamond \\ b \\ \diamond \end{pmatrix}.$$

When $L_i = \Lambda^*$ for all i, the definition in [12] is recovered.

Definition 2.2 (Quasigeodesic normal form). *normal form for* $G, X, \Lambda)$ is a set of words $L \subseteq \Lambda^*$ in bijection with G. normal form L is *quasigeodesic* if there is a constant so that

$$|u|_\Lambda \leq \quad \|u\|_X + 1)$$

for each $u \in L$, where $\|u\|_X$ is the length of a geodesic in X^* for the group element represented by u.

The $\|u\|_X + 1$ in the definition allows for normal forms where the identity of the group is represented by a nonempty string of length at most . We denote the image of $u \in L$ under the bijection with G by $\overline{u}$.

Definition 2.3 (-graph automatic group). Let be a formal language class, $G, X)$ a group and symmetric generating set, and Λ a finite set of symbols. We say that $G, X, \Lambda)$ is *-graph automatic* if there is a normal form $L \subset \Lambda^*$ in the language class such that for each $x \in X$ the language $L_x = \{\otimes\, u, v) \mid u, v \in L,\ \overline{v} =_G \overline{u}x\}$ is in the class .

If we take = {regular languages}, we recover the definition of a graph automatic group given in [12]. If we further require Λ to be the set X of group generators, we obtain the original definition of an automatic group.

In this paper we will consider the case when denotes the set of non-blind deterministic 1-counter languages, which are defined as follows.

Definition 2.4 (1-counter automaton). *non-blind deterministic* 1-*counter automaton* is a deterministic finite state automaton augmented with an integer counter: the counter is initialized to zero, and can be incremented, decremented, compared to zero and set to zero during operation. For each configuration of the machine and subsequent input letter, there is at most one possible move. The automaton accepts a word exactly if upon reading the word it reaches an accepting state with the counter equal to zero.

In drawing a 1-counter automaton, we label transitions by the input letter to be read, with subscripts to denote the possible counter instructions:

- $= 0$ to indicate the edge may only be traversed if the value of the counter is 0.
- > 0 (resp. $<$) to indicate the edge may only be traversed if the value of the counter is greater (resp. less) than 0.
- $+1$ to increment the counter by 1.
- -1 to decrement the counter by 1.

The class $_1$ of 1-counter languages is closed under homomorphism and inverse homomorphism [10], intersection with regular languages and finite union [10], see also [7, Lemma 3], and is strictly contained in the class of context-free languages [7, Lemmas 1–2].

We end this section with a lemma which is used to streamline several proofs in later sections, and is a restatement of [14, Lemma 2.6].

Lemma 2.5 ([14, Lemma 2.6]). *Fix a symbol alphabet Λ and let L_1 and L_2 be regular languages over Λ, and $x \in \Lambda^*$ a fixed word. Then the set*

$$\{\otimes\, zw, zxw) \mid z \in L_1,\ w \in L_2\}$$

is a regular language.

If the language of normal forms for a k-counter-graph automatic group is additionally quasigeodesic, it is proven in [8] that given a string of group generators of length n, the normal form word can be computed in time $O\ n^{2k+2})$. It is proven in [9] that normal forms are quasigeodesic in any automatic group, and in [12] that normal forms are quasigeodesic in graph-automatic groups. In [8] an example is given to show that a -graph automatic structure need not have a quasigeodesic normal form language.

2.2 Thompson's group

Thompson's group F can be equivalently viewed from three perspectives:
(1) s the group defined by the infinite presentation

$$\mathcal{P}_{\text{inf}} = \langle x_0, x_1, x_2 \cdots \mid x_j x_i = x_i x_{j+1} \text{ whenever } i < j\rangle$$

or the finite presentation

$$\mathcal{P}_{\text{fin}} = \langle x_0, x_1 \mid [x_0^{-1}x_1, x_0^{-1}x_1x_0], [x_0^{-1}x_1, x_0^{-2}x_1x_0^2]\rangle.$$

(2) s the set of piecewise linear homeomorphisms of the interval $[0, 1]$ satisfying
(a) each homeomorphism has finitely many linear pieces,
(b) all breakpoints have coordinates which are dyadic rationals,
(c) all slopes are powers of two.
(3) s the set of pairs of reduced finite rooted binary trees with the same number of nodes, or carets.
For the equivalence of these three interpretations of this group, as well as a more robust introduction to the group, we refer the reader to [4].

There is a standard normal form for elements of F with respect to the infinite generating set, as discussed by Brown and Geoghegan [2]. ny element $g \in F$ can be written uniquely, by applying the relations from the presentation $\mathcal{P}_{\text{inf}}$, as

$$x_{i_0}^{e_0} x_{i_1}^{e_1} \cdots x_{i_m}^{e_m} x_{j_n}^{-f_n} \cdots x_{j_1}^{-f_1} x_{j_0}^{-f_0}, \tag{2.1}$$

where
(1) $0 \le i_0 < i_1 < i_2 < \cdots < i_m$ and $0 \le j_0 < j_1 < j_2 < \cdots < j_n$,
(2) $e_i, f_j > 0$ for all i, j,
(3) if x_i and x_i^{-1} are both present in the expression, then so is x_{i+1} or x_{i+1}^{-1}.
We will refer to this as the standard infinite normal form for $g \in F$; an expression in this form is called *reduced*. If an expression of the form in (2.1) contains x_i and x_i^{-1} but not x_{i+1} or x_{i+1}^{-1} for some i, it is called *unreduced*. Relations from $\mathcal{P}_{\text{inf}}$ must be applied to obtain an equivalent expression satisfying the third condition above, and then the resulting expression is reduced.

3 1-counter language of normal forms for elements of

In this section we define a language of normal forms, based on the standard infinite normal form for elements of F, which yields a 1-counter graph automatic structure for F. If $g \in F$, let w denote the infinite normal form for g given in (2.1). From the expression w, define a normal form over the finite alphabet $\{\#, a, b\}$ as follows.
(1) First, we require that every generator between x_0 and x_M appear twice in the expression in (2.1), where $M = \max\{i_m, j_n\}$. To accomplish this, insert x_t^0 for any index $t \le M$ not appearing in (2.1). This yields a word of the form

$$x_0^{r_0} x_1^{r_1} x_2^{r_2} \cdots x_M^{r_M} x_M^{-s_M} \cdots x_2^{-s_2} x_1^{-s_1} x_0^{-s_0}, \tag{3.1}$$

where $r_i, s_i \ge 0$, exactly one of r_M, s_M is nonzero, and $r_i s_i > 0$ implies $r_{i+1} + s_{i+1} > 0$.
(2) Second, rewrite the expression in (3.1) in the form

$$a^{r_0} b^{s_0} \# a^{r_1} b^{s_1} \# \cdots \# a^{r_M} b^{s_M},$$

where $r_i, s_i \ge 0$, exactly one of r_M, s_M is nonzero, and $r_i s_j > 0$ implies $r_{i+1} + s_{i+1} > 0$.

Define L_∞ to be the set of all such strings satisfying these conditions on r_i and s_i, together with the empty string.

s an example, the element with standard infinite normal form

$$x_1^2 x_4^3 x_8^{-1} x_5^{-6} x_4^{-2}$$

has the following representative in the language L_∞:

$$\#aa\#\#\#aaabb\#bbbbbb\#\#\#b.$$

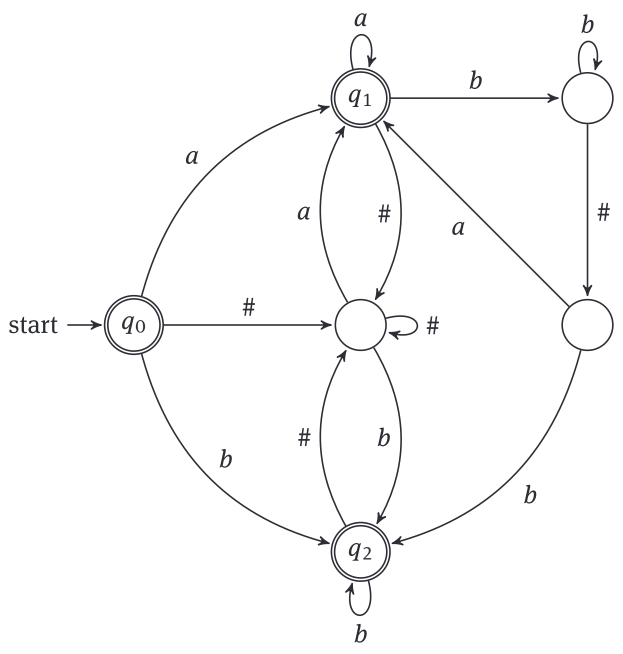

Figure 1. finite state automaton accepting the language L_∞. ccept states are q_0, q_1, q_2.

Note that if the largest index in the standard infinite normal form string w is k, then the normal form string in L_∞ representing w contains exactly k # symbols.

Lemma 3.1. *The language L_∞ is a normal form for F, and is a regular language.*

Proof. The procedure described above clearly gives a bijection between the set of standard infinite normal forms for elements of F and the language L_∞. To see that this language is regular, we give a finite state machine which accepts it in Figure 1. The automaton only accepts strings of the form ε, a^i, b^i, $w\#a^i$, $w\#b^i$ with $i > 0$ and w a word in $\{a, b, \#\}^*$ with no subword ba, and if w has a subword ab, then any maximal substring of the form

$$a^{k_1}b^{k_2}\#a^{k_2}b^{l_2}\#\cdots\#a^{k_z}b^{l_z}\#,$$

where all $k_i, l_i > 0$ is immediately followed by $a^j\#$ or $b^j\#$. □

To prove that this language of normal forms is quasigeodesic we quote a proposition proven by Burillo in [3].

Proposition 3.2 ([3, Proposition 2]). *Let $g \in F$ have standard infinite normal form*

$$x_{i_1}^{e_1}x_{i_2}^{e_2}\cdots x_{i_m}^{e_m}x_{j_n}^{-f_n}\cdots x_{j_2}^{-f_2}x_{j_1}^{-f_1}$$

and define

$$= \sum_{l=1}^{m} e_l + \sum_{l=1}^{n} f_l + i_m + j_n.$$

Then $\frac{}{6} - 2 \le l\ g) \le 3$ *, where l g) denotes the word length of g with respect to the finite generating set* $\{x_0, x_1\}$.

Using Proposition 3.2 we prove the following proposition.

Proposition 3.3. *The language L_∞ is quasigeodesic.*

Proof. Let g have standard infinite normal form $x_{i_1}^{e_1}x_{i_2}^{e_2}\cdots x_{i_m}^{e_m}x_{j_n}^{-f_n}\cdots x_{j_2}^{-f_2}x_{j_1}^{-f_1}$ which we alter as above to the expression in (3.1):

$$x_0^{r_0}x_1^{r_1}\cdots x_M^{r_M}x_M^{-s_M}\cdots x_1^{-s_1}x_0^{-s_0}$$

in which each index 0 through $M = \max\{i_m, j_n\}$ appears twice, although possibly with zero exponent. If $r_k = 0$ (resp. $s_k = 0$), then $r_k = e_i$ (resp. $s_k = f_i$), for some $i \le M$. The length of the resulting normal form for g in the language L_∞ is then

$$\sum_{l=0}^{M} r_l + s_l) + M = \sum_{l=1}^{m} e_l + \sum_{l=1}^{n} f_l + M = \quad '.$$

It is clear that $\quad \le \quad ' \le 2 \quad$, as $M = \max\{i_m, j_n\}$. Combining this with the inequality in Proposition 3.2 we obtain

$$\frac{\ '}{12} - 2 \le \frac{\ }{6} - 2 \le l\ g) \le 3 \quad \le 3 \quad '$$

and hence L_∞ is a language of quasigeodesic normal forms for elements of F. □

Next we consider the multiplier languages $L_{x_0^{-1}}$ and $L_{x_1^{-1}}$. It is proven in [8] that for $\mathscr{R}$ = {regular languages} or $\quad_1$ = {1-counter languages} we have $L_x \in \mathscr{R}$ (resp. $\quad_1$) if and only if $L_{x^{-1}} \in \mathscr{R}$ (resp. $\quad_1$). Hence it suffices to consider only the multiplier languages $L_{x_0^{-1}}$ and $L_{x_1^{-1}}$.

Proposition 3.4. *The language $L_{x_0^{-1}} = \{\otimes\ u, v) \mid u, v \in L_\infty,\ \overline{u}x_0^{-1} =_F \overline{v}\}$ is regular.*

Proof. Let $g \in F$ with L_∞ normal form

$$u = a^{r_0}b^{s_0}\#a^{r_1}b^{s_1}\#\cdots\#a^{r_M}b^{s_M}$$

and infinite normal form w. We consider two cases, depending on whether wx_0^{-1} is already in infinite normal form, or must be simplified in order to obtain the infinite normal form.

First suppose that wx_0^{-1} is in infinite normal form. This occurs if and only if any one of the following conditions holds:

(1) The expression w contains no x_0 terms to a positive exponent, that is, $r_0 = 0$.
(2) The expression w contains x_1 to a nonzero power, that is, $r_1 = 0$ or $s_1 = 0$.
(3) The expression w contains x_0 to a negative power, that is, $s_0 = 0$.

Write $u = a^{r_0}b^{s_0}y \in L_\infty$, where y is empty or has initial letter #. Then $v = a^{r_0}b^{s_0+1}y$ is the word in L_∞ corresponding to ux_0^{-1}.

Next suppose that wx_0^{-1} is not in infinite normal form, so that none of the above conditions hold. That is, $u = a^{r_0}\#\#a^{r_2}b^{s_2}y$, where $r_0 > 0$ and y is either empty (and at least one of r_2, s_2 is zero) or begins with #. If $r_2 = s_2 = 0$ and y is empty, then $u = a^{r_0}$ and $v = a^{r_0-1}$ if $r_0 > 1$, and $u = a$ and $v = \epsilon$ if $r_0 = 1$. Otherwise wx_0^{-1} is an unreduced expression and has the form $x_0^{e_0}x_i^{e_i}\cdots x_j^{-f_j}x_0^{-1}$ for some $i, j > 1$. Rewrite this expression as $wx_0^{-1} = x_0^{e_0}\beta x_0^{-1}$, where $\beta = x_i^{e_i}\cdots x_j^{-f_j}$.

From the infinite presentation for F we have the relations $x_0x_{j+1}x_0^{-1} = x_j$ for $j \ge 1$. We apply this repeatedly to the word $x_0^{e_0}\beta x_0^{-1}$ to obtain $x_0^{e_0-1}\beta'$, where x_i in β is replaced by x_{i-1} to obtain β'. s none of the exponents are altered in this process, as words in L_∞ we have

- $u = a^{r_0}\#\#a^{r_2}b^{s_2}y$,
- $v = a^{r_0}\#a^{r_2}b^{s_2}y$

where v denotes the string in L_∞ corresponding to wx_0^{-1}.

Under either assumption, let K be the set of convolutions $\otimes\ u, v)$ where u and v have any of the above forms„ altered so that any string y is allowed to lie in the set $\{a, b, \#\}^*$. It follows from Lemma 2.5 that K is a regular language. Then $L_{x_0^{-1}} = K \cap \otimes\ L_\infty, L_\infty)$ is a regular language and accepts exactly those convolutions $\otimes\ u, v)$, where $u, v \in L_\infty$ and $\overline{u}x_0^{-1} =_F \overline{v}$. □

We now show that the multiplier language $L_{x_1^{-1}}$ is a deterministic non-blind 1-counter language. In Lemma 3.7 we prove this language cannot be regular.

Proposition 3.5. *The language $L_{x_1^{-1}} = \{\otimes\ u, v) \mid u, v \in L_\infty,\ \overline{u}x_1^{-1} =_F \overline{v}\}$ is a deterministic non-blind 1-counter language.*

Proof. Let $g \in F$ have L_∞ normal form

$$u = a^{r_0}b^{s_0}\#a^{r_1}b^{s_1}\#\cdots\#a^{r_M}b^{s_M}$$

and infinite normal form w. We will always use v to denote the representative in L_∞ of gx_1^{-1}. We consider several cases depending on the value of certain exponents.

Case 1. Suppose first that $s_0 = 0$, that is, the infinite normal form w for g does not contain x_0 to a negative exponent. We consider three subcases.

Case 1.1. If $u = a^{r_0}$ for $r_0 \geq 0$, then $v = a^{r_0}\#b$.

Case 1.2. If u contains at least one # symbol, then wx_1^{-1} is the infinite normal form for gx_1^{-1} if at least one of the following conditions holds:
(1) The expression w contains no x_1 terms to a positive exponent, that is, $r_1 = 0$.
(2) The expression w contains an x_1 to a negative power, that is, $s_1 = 0$.
(3) The expression w contains x_2 to a nonzero power, that is, $r_2 = 0$ or $s_2 = 0$.
If we write $u = a^{r_0}\#a^{r_1}b^{s_1}\gamma \in L_\infty$, where γ is empty or begins with #, then $v = a^{r_0}\#a^{r_1}b^{s_1+1}\gamma$.

Case 1.3. If none of the previous conditions hold, we must have $u = a^{r_0}\#a^{r_1}\gamma \in L_\infty$ with $r_1 > 0$ and γ either empty or $\gamma = \#\#\gamma'$. The corresponding infinite normal form is then $w = x_0^{r_0}x_1^{r_1}\eta$, where η is either empty or $\eta = x_i^{r_i}\cdots x_j^{-s_j}$ for some $i, j > 2$ and $r_k, s_l > 0$. Then $wx_1^{-1} = x_0^{r_0}x_1^{r_1}\eta x_1^{-1}$. s in Proposition 3.4, this simplifies to $wx_1^{-1} = x_0^{r_0}x_1^{r_1-1}\eta'$, where $x_i^{\pm 1}$ in η is replaced by $x_{i-1}^{\pm 1}$ to obtain η'. Since η and η' have the same sequence of exponents, we have, for $r_1 > 0$,
- $u = a^{r_0}\#a^{r_1}\gamma$,
- (1) $v = a^{r_0}\#a^{r_1-1}$ if $r_1 > 1$ and γ is empty,
  (2) $v = a^{r_0}$ if $r_1 = 1$ and γ is empty,
  (3) $v = a^{r_0}\#a^{r_1-1}\#\gamma'$ if $r_1 > 1$ and $\gamma = \#\#\gamma'$.

Let K denote the regular language of convolutions $\otimes\ u, v)$ arising from Case 1, with the string γ replaced by any string in $\{a, b, \#\}^*$. Then K is the union of four languages, splitting Case 1.3 according to whether γ is empty or not. The languages of pairs $\otimes\ u, v)$ arising from Cases 1.1 or 1.3 with γ empty are clearly regular, and it follows from Lemma 2.5 that the languages arising from Cases 1.2 and 1.3, with γ nonempty in the latter, are also regular. Thus K is a regular language, and $L_{s_0=0} = K \cap \otimes\ L_\infty, L_\infty)$ is a regular language, and contains exactly those convolutions of strings covered by Case 1.

Case 2. Next suppose that $s_0 = 0$, so w ends in x_0^{-1}, and hence wx_1^{-1} is not the infinite normal form for gx_1^{-1}. We will describe $L_{s_0=0}$, which is the set of all strings $\otimes\ u, v) \in L_{x_1^{-1}}$ in which u satisfies $s_0 = 0$.

We will apply the relation $x_i^{-1}x_j^{-1} = x_{j+1}^{-1}x_i^{-1}$ for $i < j$ repeatedly, to "push" the final x_1^{-1} to the left in this expression, at the "cost" of increasing its index.

Let
$$w = x_{i_0}^{e_0}x_{i_1}^{e_1}\cdots x_{i_m}^{e_m}x_{j_n}^{-f_n}\cdots x_{j_1}^{-f_1}x_{j_0}^{-f_0} \tag{3.2}$$
be the infinite normal form for g, where $f_0 = s_0 = 0$. pplying the relation above f_0 times to the expression wx_1^{-1} yields
$$x_{i_0}^{e_0}x_{i_1}^{e_1}\cdots x_{i_m}^{e_m}x_{j_n}^{-f_n}\cdots x_{j_2}^{-f_2}x_{j_1}^{-f_1}\left(x_{1+f_0}\right)^{-1}x_0^{-f_0}.$$
If $1 + f_0 \leq j_1$, then this process is completed, and we must determine if the resulting word is reduced, or can be simplified further. If $1 + f_0 > j_1$, then we can apply the relation again f_1 times to obtain
$$x_{i_0}^{e_0}x_{i_1}^{e_1}\cdots x_{i_m}^{e_m}x_{j_n}^{-f_n}\cdots x_{j_2}^{-f_2}\ x_{1+f_0+f_1})^{-1}x_{j_1}^{-f_1}x_0^{-f_0}.$$
We continue applying this relation until the first time we obtain x_R^{-1}, where either
- $R = 1 + f_0 + f_1 + \cdots + f_n > j_n$, or
- $R = 1 + f_0 + f_1 + \cdots + f_t \leq j_{t+1}$ for some $1 \leq t \leq n - 1$.

Case 2.1. Suppose that $R > j_n$. We must consider the relative values of R and $M = \max\{i_m, j_m\}$.

First assume that $R > M$. Then the infinite normal form expression for gx_1^{-1} is
$$x_{i_0}^{e_0}x_{i_1}^{e_1}\cdots x_{i_m}^{e_m}x_R^{-1}x_{j_n}^{-f_n}\cdots x_{j_2}^{-f_2}x_{j_1}^{-f_1}x_0^{-f_0}$$
with $f_0 = 0$. It follows that the representatives of the elements g and gx_1^{-1} in L_∞ are:
- $u = a^{r_0}b^{s_0}\gamma$, where γ is either empty or starts with #, ends with a or b and contains exactly M # symbols.
- $v = a^{r_0}b^{s_0}\gamma\#^{R-M}b$. Note that the number of # symbols in this word is R, as required.

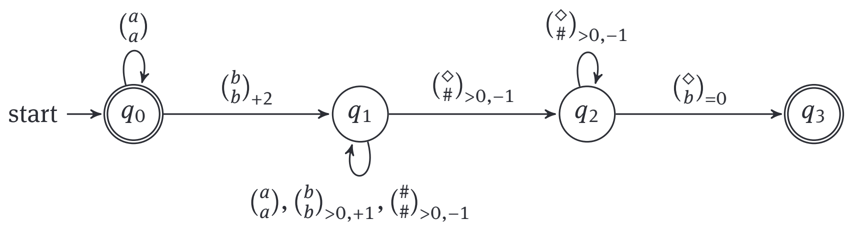
start
q_0
q_1
q_2
q_3
$\binom{a}{a}$
$\binom{b}{b}_{+2}$
$\binom{a}{a}, \binom{b}{b}_{>0,+1}, \binom{\#}{\#}_{>0,-1}$
$\binom{\diamond}{\#}_{>0,-1}$
$\binom{\diamond}{\#}_{>0,-1}$
$\binom{\diamond}{b}_{=0}$

Figure 2. deterministic non-blind 1-counter automaton used in Case 2.1 of the proof of Proposition 3.5 when $R > M$. The start state is q_0 and the accept state is q_3.

The deterministic non-blind 1-counter automaton given in Figure 2 accepts exactly those strings of the form $\otimes\ a^{r_0}b^{s_0}\gamma, a^{r_0}b^{s_0}\gamma\#^{R-M}b)$ in which the suffix of the convolution has the correct number of $\binom{\#}{\#}$ symbols and $\gamma \in \{a, b, \#\}^*$. We denote this language by L'. The automaton operates as follows.

- fter reading $\binom{a}{a}^i\binom{b}{b}$ the value of the counter is set to 2, and the automaton is in state q_1.
- For each $\binom{b}{b}$ read the counter is increased by 1, and for each $\binom{\#}{\#}$ the counter is decremented by 1, so that after reading a prefix of a string from L_∞ containing p copies of the symbol $\binom{\#}{\#}$, the value of the counter is equal to
$$1 + f_0 + f_1 + \cdots + f_{p-1} - p.$$
- If the counter returns to zero while at state q_1 after having read p copies of the symbol $\binom{\#}{\#}$, we are not in Case 2.1 and the input is rejected, since the edge leaving q_1 for state q_2 checks that the counter is positive.
- Once the automaton has read the string $\otimes\ a^{r_0}b^{s_0}\gamma, a^{r_0}b^{s_0}\gamma)$, the value of the counter is
$$1 + f_0 + f_1 + \cdots + f_n - M = R - M$$
since we have read all the $\binom{b}{b}$ letters in $\otimes\ a^{r_0}b^{s_0}\gamma, a^{r_0}b^{s_0}\gamma)$ and M $\binom{\#}{\#}$ letters. Recall that $M = \max\{i_m, j_n\}$. From here the input is accepted precisely if the remaining letters to be read are $\otimes\ \varepsilon, \#^{R-M}b)$, as verified by the automaton.

Then $L_{R>M} = L' \cap \otimes\ L_\infty, L_\infty)$ is a deterministic non-blind 1-counter automaton which accepts exactly those strings from Case 2.1 which satisfy $R > M$.

If $R = M$, which can occur if and only if $R = i_m$, we claim that the infinite normal form for gx_1^{-1} is either

$$x_{i_0}^{e_0}x_{i_1}^{e_1}\cdots x_{i_m}^{e_m-1}x_{j_n}^{-f_n}\cdots x_{j_2}^{-f_2}x_{j_1}^{-f_1}x_0^{-f_0} \quad \text{if } e_m > 1, \tag{3.3}$$

or

$$x_{i_0}^{e_0}x_{i_1}^{e_1}\cdots x_{i_{m-1}}^{e_{m-1}}x_{j_n}^{-f_n}\cdots x_2^{-f_2}x_{j_1}^{-f_1}x_0^{-f_0} \quad \text{if } e_m = 1, \tag{3.4}$$

that is, in (3.4) we have $i_{m-1} = j_n$ and hence there is no additional cancelation of terms through application of relations from $\mathcal{P}_{\text{inf}}$. We justify this statement as follows.

If $i_{m-1} = j_n$, as we began with a reduced expression in the infinite normal form for g, we must have $i_m = i_{m-1} + 1 = j_n + 1$. In Case 2.1, it is always true that

$$1 + f_0 + \cdots + f_b \geq j_{b+1} + 1$$

for all $0 \leq b \leq n$. So

$$1 + f_0 + f_1 + \cdots + f_{n-1} \geq j_n + 1$$

and it follows that

$$R = 1 + f_0 + f_1 + \cdots + f_n \geq j_n + 2 = i_m + 1,$$

in which case $R = i_m$. Thus $i_{m-1} = j_n$ and (3.4) is the infinite normal form for gx_1^{-1}.

gain letting $\overline{u}x_1^{-1} =_F \overline{v}$, in this case we have

- $u = \gamma\#^s a^{e_m}$, where γ ends in a or b,
- $v = \gamma\#^s a^{e_m-1}$ when $e_m > 1$ and $v = \gamma$ otherwise.

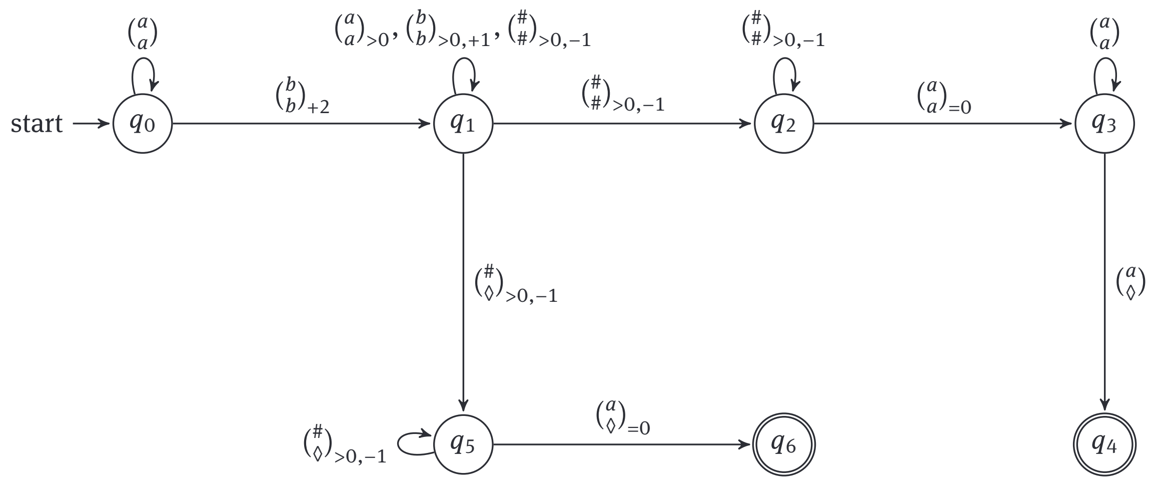

Figure 3. deterministic non-blind 1-counter automaton used in Case 2.1 of the proof of Proposition 3.5 when $R = M$. The start state is q_0 and accept states are q_4 and q_6. The top path is followed when $e_m > 1$ and the bottom when $e_m = 1$.

The 1-counter automaton in Figure 3 accepts exactly those convolutions of elements of the above forms with $y \in \{a, b, \#\}^*$. The top line of states and transitions is followed when $e_m > 1$ and the bottom line when $e_m = 1$. Note that the value of the counter must equal 0 when the difference in the words is detected by the automaton. Intersecting the language accepted by this machine with $\otimes L_\infty, L_\infty)$ yields a language $L_{R=M}$ consisting exactly of those convolutions of strings accepted in Case 2.1 with $R = M$.

If $R < M$, then we must have $M = i_m$ and we consider three scenarios for the strings $u, v \in L_\infty$.

(1) The generator x_R does not appear in the infinite normal form for g, and hence we have
- $u = y\#a^{r_{R-1}}b^{s_{R-1}}\#\#\eta$,
- $v = y\#a^{r_{R-1}}b^{s_{R-1}}\#b\#\eta$.

(2) The generator x_R does appear in the infinite normal form for g, as does x_{R+1} and hence we have
- $u = y\#a^{r_R}\#a^{r_R\ 1}\#\eta$,
- $v = y\#a^{r_R}b\#a^{r_R\ 1}\#\eta$.

(3) The generator x_R does appear in the infinite normal form for g but x_{R+1} does not, and hence we have
- $u = y\#a^{r_R}\#\#\eta$,
- $v = y\#a^{r_R-1}\#\eta$

as in Case 1.3.

In all three cases above, $\eta \subset \{a, \#\}^*$.

finite state machine accepting convolutions of the above strings must check that the difference in the strings comes at the correct position; for this we require a single counter.

We now build a 1-counter machine which accepts the language of all strings $\otimes\ u, v)$ as above with y and η replaced with an arbitrary string from $\{a, b, \#\}^*$. Each $\otimes\ u, v)$ in this language is the concatenation of a string from a prefix language with a string from a suffix language. The prefix language is a non-blind deterministic 1-counter language based on the same counter instructions as in Figure 3, and below we explain why the counter must have value 0 to transition between the two languages. The suffix languages, one for each type above, are all regular according to Lemma 2.5. The prefix and suffix languages are given in the following table, where y and η denote words in $\{a, b, \#\}^*$.

Type of pair	Prefix language	Suffix language	FS accepting suffix language
(1)	$\{\otimes\ ya^nb^k\#\#, ya^nb^k\#b)\}$ with $n, k \geq 0$	$\{\otimes\ \eta, \#\eta)\}$	M_1
(2)	$\{\otimes\ y\#a^n\#, y\#a^nb)\}$ with $n \geq 1$	$\{\otimes\ a^k\#\eta, \#a^k\#\eta)\}$ with $k > 0$	M_2
(3)	$\{\otimes\ y\#a^n, y\#a^n)\}$with $n \geq 0$	$\{\otimes\ a\#\#\eta, \#\eta)\}$	M_3

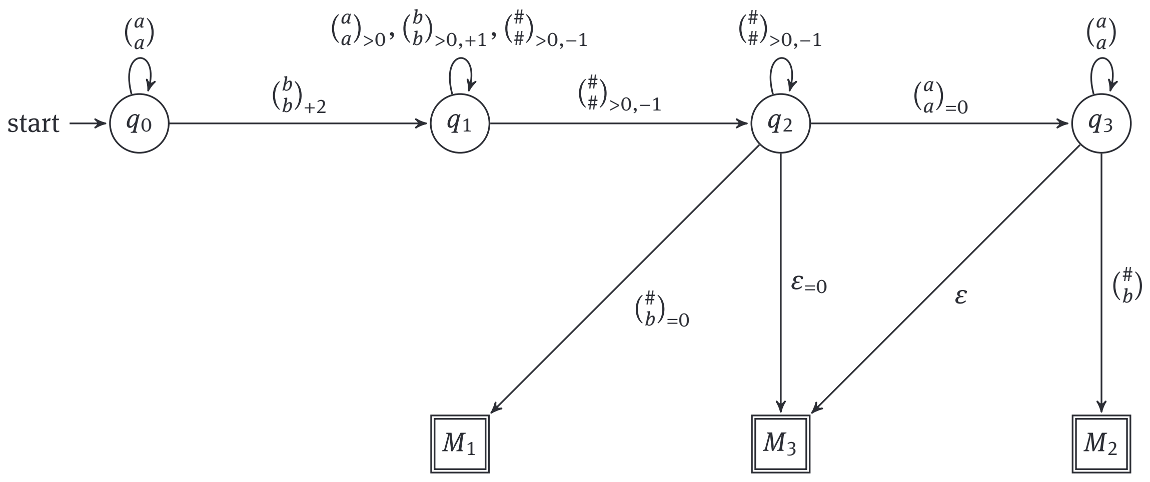

Figure 4. deterministic non-blind 1-counter automaton which verifies that the change in u and v from the case $R < M$ occurs at the correct position in the word. The start state is q_0 and accept states are the accept states from the finite state machines M_1, M_2 and M_3. n arrow terminating at one of these machines is understood to terminate at the start state of the machine.

Figure 4 presents a 1-counter automaton which accepts the language of convolutions of concatenations of a prefix and suffix from the above table. ny arrow terminating at a state labeled M_i for $i = 1, 2, 3$ is assumed to terminate at the start state of that machine. The ε edges can be removed but are used to give the simplest depiction of the machine.

This automaton initially operates with the same counter instructions as in Figures 2 and 3, that is, $\binom{b}{b}_{+2}$ initially followed by $\binom{b}{b}_{>0,+1}$ and $\binom{\#}{\#}_{>0,-1}$, as in Figures 2 and 3. Note that after reading a prefix in which there are p copies of the symbol $\binom{\#}{\#}$, the value of the counter is equal to

$$1 + f_0 + f_{i_1} + \cdots + f_{i_{p-1}} - p.$$

In the case $R < M$, we have $1 + f_0 + f_1 + \cdots + f_{p-1} > p$ for all p, and we have $R = 1 + f_0 + f_1 + \cdots + f_n$. If $\otimes\ u, v)$ arises in the case $R < M$, the value of the counter will first equal 0 when R symbols of the form $\binom{\#}{\#}$ have been read. If we divide $\otimes\ u, v)$ into a prefix p and suffix s using the above prefix and suffix languages, the counter will first equal zero after the final $\binom{\#}{\#}$ in p has been read. Hence we verify in the machine in Figure 4 that the value of the counter has value 0 after the final $\binom{\#}{\#}$ in the prefix word before transitioning to the suffix word.

Let $L_{R<M}$ denote the intersection of the language accepted by the machine in Figure 4 with $\otimes\ L_\infty, L_\infty)$. Then $L_1 = L_{R<M} \cup L_{R=M} \cup L_{R>M}$ is the language consisting exactly of those convolutions of strings described in Case 2.1.

Case 2.2. For the remaining case, suppose that $1 + f_0 + f_1 + \cdots + f_{t-1} = t$ for some $t \le j_n$. Beginning with the expression in (3.2) for the infinite normal form of g, we can write the infinite normal form for gx_1^{-1} as

$$x_0^{e_0} \cdots x_{i_m}^{e_m} x_{i_n}^{-f_n} \cdots x_{j_{w\ 1}}^{-f_{w\ 1}} x_t^{-1} x_{j_w}^{-f_w} \cdots x_0^{-f_0}. \tag{3.5}$$

We again consider subcases, depending on whether or not the expression in (3.5) is the infinite normal form for gx_1^{-1}. In each case, we show that the language $\{\otimes\ u, v)\}$ of accepted words in that case is the concatenation of a prefix language and a suffix language. The suffix language is always a regular language. The prefix language is always a deterministic non-blind 1-counter language. While it first seems like the language of all possible prefixes is also regular, we must use a counter to ensure that the difference between the strings u and v occurs at the proper place in the string. We again use the counter instructions $\binom{b}{b}_{+2}$ followed by $\binom{b}{b}_{>0,+1}$ and $\binom{\#}{\#}_{>0,-1}$, as in Figures 2 and 3. Note that after reading a prefix in which there are p copies of the symbol $\binom{\#}{\#}$, the value of the counter is equal to

$$1 + f_0 + f_{i_1} + \cdots + f_{i_{p-1}} - p.$$

When $1 + f_0 + f_{i_1} + \cdots + f_{i_t} - t = 0$, we are in a position to read the additional x_t letter introduced by permuting

$$\begin{tikzcd}
\text{start} \arrow[r] & q_0 \arrow[loop above, "\binom{a}{a}"] \arrow[r, "\binom{b}{b}_{+2}"] & q_1 \arrow[loop above, "{\binom{a}{a}_{\geq 0}, \binom{b}{b}_{>0,+1}, \binom{\#}{\#}_{>0,-1}}"] \arrow[r, "\binom{a}{\#}_{=0}"] & \boxed{M}
\end{tikzcd}$$

Figure 5. deterministic non-blind 1-counter automaton which accepts exactly those strings from Case 2.2.1 with $\gamma, \eta \in \{a, b, \#\}^*$. The start state is q_0 and the accept states are the accept states of M. ny arrow terminating at M is assumed to terminate at the start state of M.

the x_1^{-1} past generators with smaller indices using the group relations. The prefix string always terminates with the difference resulting from the newly introduced generator x_t^{-1}. In Case 2.2.1, this difference is the removal of an “a” symbol to obtain the reduced expression for gx_1^{-1}. In Case 2.2.2, the prefix string always terminates with the symbol $\binom{\#}{b}$ where the “b” corresponds to the x_t^{-1}.

Case 2.2.1. If (3.5) is not the infinite normal form for gx_1^{-1}, then $t = j_{w+1}$, there is an index $p \leq m$ so that $i_p = t$, and x_{t+1} is not present in the normal form for g to any nonzero power, that is, $i_{p+1} = t + 1$ and $j_{w+1} = t + 1$. This case is analogous to Case 1.3 above, and we can write

- $u = \gamma a^{r_t}\#\#\eta$,
- $v = \gamma a^{r_t - 1}\#\eta$.

Note that each $\otimes\ u, v)$ of the above form can be written as a prefix $\otimes\ \gamma a, \gamma\#)$ and a suffix from the language $\{\otimes\ \#\#\eta, \eta)\}$. If we replace γ and η with any strings from $\{a, b, \#\}^*$, then it follows from Lemma 2.5 that the language of all possible suffixes is regular. Let M denote the finite state machine accepting these suffixes. s explained above, we require that the value of the counter be 0 to transition from a prefix word to a suffix word.

Figure 5 contains a deterministic non-blind 1-counter automaton accepting concatenations of prefix and suffix words of this form. Let $L_{2.2.1}$ be the intersection of the language accepted by this machine with $\otimes\ L_\infty, L_\infty)$. Then $L_{2.2.1}$ is exactly the set of convolutions described in Case 2.2.1.

Case 2.2.2. Now suppose that equation (3.5) is the infinite normal form for gx_1^{-1}. This occurs in three ways:

(1) If x_t^{-1} is already in the normal form of g, then
   - $u = \gamma\#a^{r_t}b^{s_t}\#\eta$ with $s_t = 0$,
   - $v = \gamma\#a^{r_t}b^{s_t+1}\#\eta$.

(2) If x_t^{-1} is not in the infinite normal form for g, but x_t and either x_{t+1} or x_{t+1}^{-1} are present, then
   - $u = \gamma\#a^{r_t}\#a^{r_t\ 1}b^{s_t\ 1}\#\eta$ with $r_t > 0$ and $r_{t+1} + s_{t+1} > 0$,
   - $v = \gamma\#a^{r_t}b\#a^{r_t\ 1}b^{s_t\ 1}\#\eta$.

(3) If both x_t and x_t^{-1} are not present in the infinite normal form for g, then
   - $u = \gamma\#a^{r_{t-1}}b^{s_{t-1}}\#\#\eta$,
   - $v = \gamma\#a^{r_{t-1}}b^{s_{t-1}}\#b\#\eta$.

We now claim that the set of all strings of types 1), 2) and 3) above form a non-blind deterministic 1-counter language. We again build a machine which accepts the language of such strings with γ and η replaced by any string in $\{a, b, \#\}^*$. In each case we note that the set of such words can be divided into a prefix language which is a 1-counter language and a suffix language which is regular, and the value of the counter must be 0 to transition between the two languages. The prefix and suffix languages for each type of pair $\otimes\ u, v)$ listed in Case 2.2.2 are given in the following table.

Type of pair	Prefix language	Suffix language	FS accepting suffix language
(1)	$\{\otimes\ \gamma\#a^n b^k\#, \gamma\#a^n b^{k+1})\}$ with $n \geq 0$, $k \geq 1$	$\{\otimes\ \eta, \#\eta)\}$	M_1
(2)	$\{\otimes\ \gamma\#a^n\#, \gamma\#a^n b)\}$ with $n \geq 1$	$\{\otimes\ a\eta, \#a\eta)\} \cup \{\otimes\ b\eta, \#b\eta)\}$	M_2
(3)	$\{\otimes\ \gamma\#\#, \gamma\#b)\}$	$\{\otimes\ \eta, \#\eta)\}$	M_1

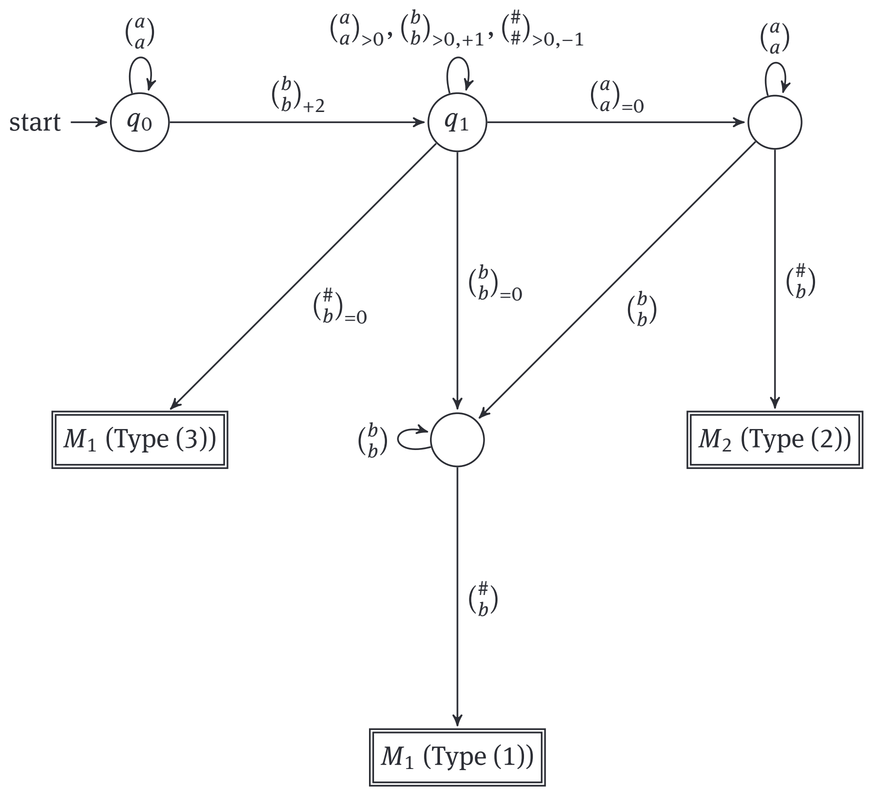

Figure 6. deterministic non-blind 1-counter automaton used in Case 2.2.2 of the proof of Proposition 3.5. The start state is q_0; any arrow terminating at M_1 or M_2 is assumed to end at the start state of the appropriate machine. ccept states of this machine are the accept states of M_1 and M_2.

The reasoning given at the beginning of Case 2.2 implies that if $\otimes\ u, v)$ is a pair in this case, then the value of the counter at the end of the prefix string is equal to zero This value is verified immediately after the t-th $\binom{\#}{\#}$ symbol is read by the machine in the prefix word.

Figure 6 contains a deterministic non-blind 1-counter language accepting concatenations of prefix and suffix words in this case, with arbitrary strings γ and η. Let $L_{2.2.2}$ be the intersection of the language accepted by the machine in Figure 6 with the regular language $\otimes\ L_\infty, L_\infty)$. Then the language of all $\otimes\ u, v)$ accepted in Case 2.2 is exactly $L_2 = L_{2.2.1} \cup L_{2.2.2}$ which is a deterministic non-blind 1-counter language.

Let $L_{s_0=0} = L_1 \cup L_2$, and it follows that $L_{x_1^{-1}} = L_{s_0=0} \cup L_{s_0=0}$ is a deterministic non-blind 1-counter language, as required. □

We have now proven the following theorem.

Theorem 3.6. *Thompson's group F is deterministic non-blind* 1*-counter graph automatic with respect to the generating set* $X = \{x_0^{\pm 1}, x_1^{\pm 1}\}$ *and symbol alphabet* $\{a, b, \#\}$.

Since 1-counter languages are (strictly) contained in the class of context-free languages, we obtain the corollary that F is context-free graph automatic. We now show that we cannot improve upon this result using the language L_∞ if normal forms for elements of F.

Lemma 3.7. *Thompson's group F is not graph automatic with respect to the normal form L_∞ and alphabet* $\{a, b, \#\}$ *given above.*

Proof. Suppose $L_{x_1^{-1}}$ was a regular language, with pumping length p. The string $\otimes\ b^p, b^p\#^{p+1}b)$ is in the language, representing
- $u =_F x_0^{-p}$,
- $v =_F x_{p+1}^{-1}x_0^{-p} =_F x_0^{-p}x_1^{-1}$.

By the Pumping Lemma for Regular Languages, $\otimes\ b^p, b^p\#^{p+1}b)$ can be partitioned into xyz with $|xy| \leq p$, $|y| > 0$ and $xy^iz \in L_{x_1^{-1}}$ for all $i \in \mathbb{N}$. However, the convolution $\otimes\ b^{p+m}, b^{p+m}\#^{p+1}b)$, for any $m > 0$, does not lie in $L_{x_1^{-1}}$ as the second word does not have the correct number of # symbols, that is, $x_0^{-\ p+m)}x_1^{-1} = x_{1+p}^{-1}x_0^{p+m}$. Hence $L_{x_1^{-1}}$ is not a regular language. □

cknowledgment: The authors wish to thank Sean Cleary, Bob Gilman, lexei Miasnikov and especially Sharif Younes for many helpful discussions during the writing of this paper.

Funding: The first author is supported by ustralian Research Council grant FT110100178. The second author acknowledges support from National Science Foundation grant DMS-1105407 and Simons Foundation grant 31736 to Bowdoin College.

References

[1] D. Berdinsky and B. Khoussainov, On automatic transitive graphs, in: *Developments in Language Theory*, Lecture Notes in Comput. Sci. 8633, Lecture Notes in Comput. Sci. (2014), 1–12.
[2] K. S. Brown and R. Geoghegan, n infinite-dimensional torsion-free FP_∞ group, *Invent. Math.* **77** (1984), no. 2, 367–381.
[3] J. Burillo, Quasi-isometrically embedded subgroups of Thompson's group *F*, *J. lgebra* **212** (1999), no. 1, 65–78.
[4] J. W. Cannon, W. J. Floyd and W. R. Parry, Introductory notes on Richard Thompson's groups, *Enseign. Math. (2)* **42** (1996), no. 3–4, 215–256.
[5] S. Cleary, S. Hermiller, M. Stein and J. Taback, Tame combing and almost convexity conditions, *Math. Z.* **269** (2011), no. 3–4, 879–915.
[6] S. Cleary and J. Taback, Seesaw words in Thompson's group *F*, in: *Geometric Methods in Group Theory*, Contemp. Math. 372, merican Mathematical Society, Providence (2005), 147–159.
[7] M. Elder, context-free and a 1-counter geodesic language for a Baumslag–Solitar group, *Theoret. Comput. Sci.* **339** (2005), no. 2–3, 344–371.
[8] M. Elder and J. Taback, -graph automatic groups, *J. lgebra* **413** (2014), 289–319.
[9] D. B. . Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson and W. P. Thurston, *Word Processing in Groups*, Jones and Bartlett, Boston, 1992.
[10] S. . Greibach, Remarks on blind and partially blind one-way multicounter machines, *Theoret. Comput. Sci.* **7** (1978), no. 3, 311–324.
[11] V. S. Guba and M. V. Sapir, The Dehn function and a regular set of normal forms for R. Thompson's group *F*, *J. ust. Math. Soc. Ser.* **62** (1997), no. 3, 315–328.
[12] O. Kharlampovich, B. Khoussainov and . Miasnikov, From automatic structures to automatic groups, *Groups Geom. Dyn.* **8** (2014), no. 1, 157–198.
[13] V. Shpilrain and . Ushakov, Thompson's group and public key cryptography, in: *pplied Cryptography and Network Security*, Lecture Notes in Comput. Sci. 3531, Springer, Berlin (2005), 151–163.
[14] J. Taback and S. Younes, Tree-based language complexity of Thompson's group *F*, *Groups Complex. Cryptol.* **7** (2015), no. 2, 135–152.