The Complexity of Verbal Languages over Groups*

Sanjay Jain†
National University
of Singapore

Alexei Miasnikov‡
Stevens Institute
of Technology

Frank Stephan§
National University
of Singapore

Abstract

This paper investigates the complexity of verbal languages and pattern languages of automatic groups. For non-Abelian free groups with finitely many generators, the following is shown: the level of a language in the Chomsky hierarchy is independent of the automatic representation; the context-free verbal languages are only the full group and the language of the empty word; the regular pattern languages are either a singleton or the full group; verbal languages are, however, indexed languages by a result of Ciobanu, Diekert and Elder. For some groups, it depends on the exactly chosen automatic representation whether a pattern language is regular, context-free or context-senstive.

Keywords: verbal languages; pattern languages; automatic groups; free groups; Chomsky hierarchy.

1 Introduction

Pattern languages are languages given by a pattern (which is a string of variables and constants) generating the language. The members of a pattern language are obtained by assigning values to the variables and replacing each of them consistently by the values assigned; usually these values are strings over fixed finite alphabet. Angluin [2] introduced the pattern languages in the realm of learning theory and subsequent investigations dealt with the question when such a pattern language is regular or context-free. Reidenbach [28] solved a long-standing open problem by showing that pattern languages are not learnable from positive data for certain alphabet sizes and deepened the research in that field. Reidenbach [29] asked whether there is a pattern language which is context-free but not regular; Jain, Ong and

*A preliminary version of this paper has been presented at the conference LICS 2012.

†Department of Computer Science, National University of Singapore, Singapore 117417, Republic of Singapore. Email: sanjay@comp.nus.edu.sg. Supported in part by NUS grants R146-000-181-112, R252-000-420-112 and C252-000-087-001. Furthermore, the work was supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2016-T2-1-019 / R146-000-234-112.

‡Department of Mathematical Sciences, Schaefer School of Engineering & Science, Steven's Institute of Technology, New Jersey, USA. Email: amiasnik@stevens.edu. The current work was started when A. Miasnikov visited the NUS on invitation by the Institute of Mathematical Sciences at the NUS within the IMS programme "Automata Theory and Applications".

§Department of Mathematics and Department of Computer Science, National University of Singapore, Singapore 119076, Republic of Singapore. Email: fstephan@comp.nus.edu.sg. Supported in part by NUS grants R146-000-181-112 and R252-000-420-112. Furthermore, the work was supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2016-T2-1-019 / R146-000-234-112.

Stephan [17] showed that this depends on the alphabet size — if the alphabet size is 2 or 3, then there are such languages, while for alphabet of size 1 or of size at least 4 such languages do not exist.

For this paper, Chomsky hierarchy refers to the hierarchy formed by the four classes of languages: regular languages, context-free languages, context-sensitive languages and recursively enumerable languages. Pattern languages, in particular the verbal languages which are generated by patterns without constants, also play an important role in group theory [18, 20, 23, 26, 27]. Gilman [12] studied the lower levels of the Chomsky hierarchy for sets in groups. Verbal sets for groups played an important role in the recent solution of Tarski's problem which showed the decidability of the first order theory of the free groups [19]; Makanin [22, 23, 24] had already shown that pattern languages are decidable in free groups, but their exact complexity in terms of the Chomsky hierarchy was left open.

The present work continues these investigations and studies which verbal languages, and more generally pattern languages, belong to which level of the Chomsky hierarchy in various automatic groups. automatic groups are interesting for computer science due to the high degree of efficiency in their group operations and the word problems. The Chomsky hierarchy is a well-accepted classification of the complexity of languages of words which has been studied thoroughly and extended to many interesting language classes during decades of investigations in theoretical computer science. The present work addresses some fundamental questions on pattern languages and verbal languages for automatic groups. It also extends the result of Miasnikov and Romankov [26] that nontrivial verbal languages in noncommutative free groups are not regular up to the next level by showing that they are also not context-free.

In the following the notation and terminology are explained. $\mathbb{Z}$ denotes the set of integers and $\mathbb{N}$ denotes the set of natural numbers $\{0, 1, 2, \ldots\}$. For an integer i, $|i|$ denotes the absolute value of i. Σ denotes a finite alphabet. A string (word) is a member of Σ^*. For a string w, $|w|$ denotes the length of w. A string v is a subword (or factor) of string w, if there are strings u and u' such that $uvu' = w$. Consider a group H generated by a non-empty finite set of generators Σ. A string over $\Sigma \cup \Sigma^{-1}$ denotes the group member formed by doing the group operation instead of concatenation in the string. Symbols $\cdot$ and $\circ$ will be used to denote the group operation. In this paper only groups generated by non-empty finite set of generators are considered. For any set A, let $A^@$ denote strings over $A \cup \{x^{-1} : x \in A\}$. For a string w, let $w^@$ denote $\{w^i : i \in \mathbb{Z}\}$ and w^* denote $\{w^i : i \in \mathbb{Z}$ and $i \geq 0\}$. The empty string ε represents the neutral element of the group. For words w, w' over $\Sigma \cup \Sigma^{-1}$, $w \sim w'$ denotes that w and w' represent the same group element in H. Similarly, if Σ' is another set of generators of the group H, and $v \in (\Sigma')^@$ and $w \in \Sigma^@$, then $v \sim w$ iff they represent the same group element over Σ' and Σ respectively.

A word w is said to be reduced if it does not contain aa^{-1} or $a^{-1}a$ as a substring for any $a \in \Sigma$. Let $red(w)$ denote a reduced word w' such that $w \sim w'$.

Two languages L, L' over $\Sigma \cup \Sigma^{-1}$ are said to be equivalent (written $L \equiv L'$) if as a set of elements of the group H, they are equal.

A set G (over $\Sigma \cup \Sigma^{-1}$) is a said to be a set of representatives of H if for all $w \in H$, there exists exactly one $v \in G$ such that $v \sim w$; this unique v is said to be the representative of w in G. Note that the representative in G, for any given $v \in H$, is unique. In this paper, unless otherwise specified, the set of representatives will always be assumed to be regular.

A function $f : G \to G$ is said to be automatic [5, 6, 14, 15, 21] iff there is a finite automaton reading pairs of strings $x, y \in G$ in parallel, in each cycle one symbol from x and one from y, such that the automaton accepts at the end iff $y = f(x)$; if the length of x and y are different then the special symbol $\#$ is used to pad up the shorter input to make both the inputs of the same length. While Hodgson [14, 15] considered automatic structures like groups where the full group operation is automatic but the representatives of the group elements can be any strings, Epstein, Cannon, Holt, Levy, Paterson and Thurston [11] favoured an approach which is taylormade for representing many finitely generated groups by a set of selected normal forms of words over generators. They defined that a group H is *automatic* iff there exists a regular set G of unique representatives of the group elements of H such that, for all $a \in G$, there exists an automatic function $f : G \to G$ satisfying $f(x) \sim ax$. Here the set G is said to be an automatic presentation / representation of the group H. For groups and semigroups, automaticity, in this paper, always refers to the model of automatic groups of Epstein, Cannon, Holt, Levy, Paterson and Thurston [11] and not to that of Hodgson [14, 15] and Khoussainov and Nerode [21]. Note that Epstein, Cannon, Holt, Levy, Paterson and Thurston [11] actually allow in their definition also multiple representatives, provided that the equality $\sim$ as group elements is an automatic relation on G. Then, the set $G' = \{x \in G : \forall y <_{ll} x\, [y \in G \Rightarrow y \not\sim x]\}$ is a regular subset of G which consists of unique representatives; here $y <_{ll} x$ denotes that y is length-lexicographically strictly less than x. Thus every set of representatives G has a subset G' of unique representatives and it is assumed here that one directly works with such a set of unique representatives. In here, note that there is an automatic function from G to G' which maps every representative to the length-lexicographically least one in G and one can obtain the mappings $x \mapsto x \circ a$ for G' from those for G by concatenating them with the function finding the least representative. The representatives of the group elements in G / G' are also known as their "normal forms".

A *pattern* π is a string over variables $X = \{x_1, \ldots, x_n, \ldots\}$, their inverses $x_1^{-1}, \ldots, x_n^{-1}, \ldots$, the generators Σ and their inverses Σ^{-1}. The members of Σ and Σ^{-1} are called constants. A substitution is a mapping from the set of variables, X, to strings over $(X \cup \Sigma)^{@}$. The string generated by π using the substitution sub, denoted $sub(\pi)$, is the string obtained by replacing each variable x in the pattern π by $sub(x)$ and replacing x^{-1} by $sub(x)^{-1}$. Note that in $sub(\pi)$, neighbouring inverses such as aa^{-1} are not cancelled (though the neighbouring aa^{-1} do get cancelled when congruent reduced words to $sub(\pi)$ are considered). A reduced word w is generated in a *cancellation-free* way by the pattern π if, for some substitution sub, $sub(\pi) = w$. A pattern π *generates* an element w of H, if for some substitution sub, $w \sim sub(\pi)$. $Lang(\pi) = \{w \in \Sigma^{@}$: for some substitution sub, $w = sub(\pi)\}$. For G being a set of representatives of a group H, π is said to generate a language $L \subseteq G$ iff $L \equiv Lang(\pi)$; for ease of notation, when the representation G is clear, one just says that the language L is generated by π. A pattern π is called constant-free iff it is a string over variables. A language is called *verbal* iff it is generated by a constant-free pattern.

Here are some examples. Consider patterns $\pi_1 = x_1 a x_1^{-1} bb$, $\pi_2 = x_1 x_2 x_1^{-1} x_2^{-1}$ and $\pi_3 = x_1 x_1 x_2 x_2$. The pattern π_1 contains constants, while π_2 and π_3 do not. Thus, the language generated by π_2 and π_3 are verbal languages. Suppose that the representation set G consists of the set of all reduced words over Σ. Now $bbbab^{-1}$ is generated by the pattern π_1 by using the substitution $sub(x_1) = bbb$; two occurrences of b^{-1} in x_1^{-1} cancel out with the two constants bb at the end of the pattern. Note that π_1 does not generate a verbal language (as ε does not belong to $Lang(\pi_1)$, though ε belongs to every verbal language).

Furthermore, the notion of reduced patterns is also useful which is defined as follows and which can be defined for any group (including automatic groups). Given a pattern π, the reduced form $red(\pi)$ is obtained by doing the following operations (a), (b), (c) and (d) until none of them can be done any more:

(a) One omits from π a subpattern of the form xx^{-1} or $x^{-1}x$ where x is a variable;

(b) One omits a constant with value ε from the pattern, unless π consists only of the constant ε;

(c) Two neighbouring constants are replaced by their product;

(d) If π can be split into $\pi' a \pi''$ where π' is a subpattern containing a variable and a is a constant which commutes with all group elements then π is replaced by the pattern $a\pi'\pi''$.

If several of these operations can be carried out at the same intermediate step, then one carries out that of the possible operations which produces the length-lexicographically least output. An example for red is given as follows: If c commutes with all group members, $d = b \cdot c$ and $d \neq \varepsilon$ then $red(x_1 a a^{-1} x_1^{-1} b x_2 x_3 c x_3^{-1} x_4) = d x_2 x_4$.

In Section 2 it is shown that for any finitely generated free group with automatic presentation G and any language $L \subseteq G$, the level of L in the Chomsky hierarchy does not change when translating L from the representation G to another automatic representation. Section 3 obtains the main results for finitely generated free groups having at least two generators: It is shown that, for free group with a representation G, the only context-free verbal languages are $\{\varepsilon\}$ and G. This improves a result of Miasnikov and Romankov [26] who had shown that these are the only regular verbal languages. For free groups with at least two generators and representation G, there are context-free pattern languages which are not regular, but they need constants. Example 11 below shows that the language generated by the pattern $x_1 a x_1^{-1}$ (for the set of generators $\Sigma = \{a, b\}$) is context-free. Ciobanu, Diekert and Elder [7] solved an open problem of a previous version of this paper and showed that for a free group with at least two generators, all verbal languages and all pattern languages are indexed languages and thus context-sensitive. The actual result shows that they are EDT0L-languages, which is a proper subclass of the indexed languages.

Note that if the generating set for the group is $\Sigma = \{a\}$, then all pattern languages are regular. Section 4 generalises this result to all Abelian automatic groups. Section 5 investigates to which extent the main results can be carried over to automatic groups which are sufficiently similar to the free groups; in few cases a positive answer is given, although the generalisation fails for most groups. The independence on representation chosen for the complexity of a language (in terms of being regular, context-free, context-sensitive or recursively enumerable) for free groups mentioned above does not hold for some automatic groups: Section 5 provides a verbal language L and a group in which L is either regular or properly context-free or properly context-sensitive depending on the automatic presentation of the group. Furthermore, this group has a representation in which all verbal languages are regular; this contrasts the case of the free group where the verbal languages cannot be context-free except for the two trivial cases. Section 6 complements these results by exhibiting a group in which, for a suitable representation, all pattern languages are context-free; however, for no representation, all verbal languages of this group are regular.

2 Complexity of languages over free groups in dependence of their representation

All groups considered in this section are free groups. For a free group represented uniquely by a regular set G, it does not depend on the choice of G which level a set L takes in the Chomsky hierarchy (see Theorem 4). Recall that the relation $\sim$ says when two group elements are equal. The proof will use the following auxiliary proposition, which was observed by the anonymous referees.

Proposition 1. *Let H be any automatic monoid with Σ being a non-empty finite set of generators for H. Let φ be the homomorphism from Σ^* to H which maps every string of generators to the group element* *represented by the product of these generators. Let $G \subseteq \Sigma^*$ be regular such that the restriction of φ to G is one-one. Let the set of subwords (or factors) of elements of G be denoted with*

$$Fac(G) = \{u \in \Sigma^* : \exists p, q\, [puq \in G]\}.$$

Now there is a constant s such that

$$\forall g \in H\, [|Fac(G) \cap \varphi^{-1}(g)| \leq s]$$

and this constant s is the index of the syntactic congruence of G.

Proof. For $v, w \in \Sigma^*$, define the syntactic congruence $\equiv_G$ of G by letting $v \equiv_G w$ iff $\forall p, q\, [pvq \in G \Leftrightarrow pwq \in G]$.

Since G is regular, the number of equivalence classes as above is some finite s. Now suppose by way of contradiction that $|Fac(G) \cap \varphi^{-1}(g)| > s$, for some $g \in H$. Then, there are $v, w \in Fac(G)$, such that $v \neq w$, $v \equiv_G w$ and $\varphi(v) = \varphi(w) = g$. As there exist p, q such that $pvq \in G$, it also holds, for these p, q, that $pwq \in G$. However, as $\varphi(v) = \varphi(w)$, one has $\varphi(pvq) = \varphi(pwq)$, a contradiction to φ being one-one on G. ∎

Corollary 2. *Let H be a free group with Σ being a non-empty finite set of generators for H.*

Let $G \subseteq \Sigma^{@}$ be any regular set of unique representatives of the elements of H. Then there is a constant s such that for any word $w \in G$ and any subword u of w, if $u \sim \varepsilon$, then $|u| \leq s$.

Lemma 3. *Suppose G and G' are regular sets and E is a finite set over an alphabet Σ.*

For $a_1, a_2, \ldots, a_n \in \Sigma$, let $S(a_1a_2 \ldots a_n, E) = Ea_1Ea_2 \ldots Ea_nE$. Furthermore, suppose $S(u, E) \cap S(v, E) = \emptyset$, for distinct $u, v \in G$.

Suppose for each $w \in G$, there exists exactly one $w' \in S(w, E) \cap G'$ and for each $w' \in G'$, there exists exactly one $w \in G$ such that $w' \in S(w, E)$.

Let h be a mapping from $w \in G$ to $w' \in G'$ such that $h(w) \in S(w, E)$.

Then, there is a (nondeterministic) finite state transducer f which transforms w to $h(w)$ and a finite state transducer which transforms $w' \in G'$, to $h^{-1}(w')$, where, for some constant c, $|w| \leq c|f(w)|$ and $|f(w)| \leq c|w|$.

Thus, any $L \subseteq G$ is regular (context free, context sensitive, recursively enumerable) iff $L' = G' \cap \bigcup_{w \in L} S(w, E)$ is regular (context free, context sensitive, recursively enumerable).

Proof. Note that for any $w = a_1a_2 \ldots a_n \in G$, $h(w)$ can be computed by guessing the members $e_0, e_1, \ldots, e_n \in E$ such that $e_0a_1e_1a_2 \ldots a_{n-1}e_{n-1}a_ne_n \in G'$. Thus, $h(w)$ can be computed by a finite state transducer.

Now, it is shown that $h^{-1}(w')$ for $w \in G'$ can be computed by a finite state transducer. Suppose M is a DFA to accept G and M'' is a DFA to accept E. Let $Q, Q'', F, F'', q_0, q_0'', \delta, \delta''$ be states of M and M'', accepting states of M and M'', starting states of M and M'' and transition functions for M and M'' respectively.

Then, build a transducer T as follows which maps strings from G' to G. Note that the transducer's output on non-members of G' is not handled in the following: if one wishes to reject such strings, then the transducer can be appropriately modified.

The states of T are $Q \times Q'' \times \{0, 1\}$. Intuitively, 0 in the third component denotes that M is processing some string from E which needs to be dropped from a string in G'. Correspondingly 1 in the third component denotes that T needs to preserve the corresponding input element.

Starting state of T is $(q_0, q_0'', 0)$. Accepting states of T are all states in $F \times F'' \times \{0\}$. The transition function δ_T for T is defined as follows.

- If $q'' \in F''$, then $\delta_T((q, q'', 0), \varepsilon)$ contains $(q, q_0'', 1)$ with output ε.
- For all $b \in \Sigma \cup \Sigma^{-1}$, if $\delta''(q'', b) = p''$, then $\delta((q, q'', 0), b)$ contains $(q, p'', 0)$ with output ε.
- For all $b \in \Sigma \cup \Sigma^{-1}$, if $\delta(q, b) = p$, then $\delta((q, q_0'', 1), b)$ contains $(p, q_0'', 0)$ with output b.

It is easy to verify that the above transducer maps $w' \in G'$ to $w \in G$ such that $h(w) = w'$.

The length of $h(w)$ and w are now bounded with respect to each other due to transducers being finite state machines, and the function h being one–one.

Now, the second part of the Lemma, namely that $L \subseteq G$ is regular (context free, context sensitive, recursively enumerable) iff $L' = G' \cap \bigcup_{w \in L} S(w, E)$ is regular (context free, context sensitive, recursively enumerable), is a standard result using transforming members of G to G' and vice versa using nondeterministic finite state transducers. ∎

Theorem 4. *Suppose H is a free group and Σ and Γ are two set of generators for H. Let G, G'' be regular set of unique representatives of H over Σ and Γ respectively. Let $L \subseteq G$. Let $L'' = \{w \in G'' : \exists v \in L\, [v \sim w]\}$. Then L is regular (context-free, context-sensitive, recursively enumerable) iff L'' is regular (respectively, context-free, context-sensitive, recursively enumerable).*

Proof. Suppose $\Gamma = \{c_1, c_2, \ldots, c_k\}$ is the set of generators for G'', where c_i represents the word $w_i \in \Sigma^{@}$. Suppose a is some fixed element in Σ. Suppose m is the maximum over lengths of w_h, $h = 1, 2, \ldots, k$. Let

- $f(c_h) = (a^{m+h} a^{-m-h}) w_h (a^{m+h} a^{-m-h})$ for $h = 1, 2, \ldots, k$,
- $f(c_h^{-1}) = (a^{m+h} a^{-m-h}) w_h^{-1} (a^{m+h} a^{-m-h})$,
- $f(uv) = f(u) \cdot f(v)$,
- $G' = \{f(u) : u \in G''\}$ and
- $L' = \{w \in G' : \exists v \in L''\, [v \sim w]\}$.

Note that each member of H has a unique representative in G', and G' is regular. Also, note that f is a homomorphism and that its inverse, which is defined only for the range of f, is computed by a finite state transducer. Thus L'' is regular (context-free, context-sensitive, recursively enumerable) iff $L' = f(L'')$ is regular (respectively, context-free, context-sensitive, recursively enumerable). Here, for context-sensitive case, note that for some constant c, $|w| \leq c|f(w)|$ and $|f(w)| \leq c|w|$, and thus $|w|$ and $|f(w)|$ are within constant factor of each other.

Now Theorem 4 follows from Lemma 3. ∎

Remark 5. *The anonymous referees pointed out to the authors, that a more common notion of automatic groups does not require unique representatives. Epstein, Cannon, Holt, Levy, Paterson and Thurston [11] defined that the set of normal forms (that is, representatives) is a set G of elements represented as strings over the generators giving these elements and that for all* z *which are either the neutral element or a generator or the inverse of a generator, the relation* $\{(x, y) : x, y \in G \wedge x \circ z = y\}$ *is automatic. This is equivalent to saying that the functions* $x \mapsto x \circ z$ *are automatic and the equality is automatic. So for every* $z \in G$ *there is an automatic function* f_z *which maps every member* x *of* G *to the length-lexicographically least representative inside* G *of* $x \circ z$ *and the set* G' *of such length-lexicographically minimal representatives is also regular.*

Given $L \subseteq G$ *which respects* $\sim$*, the set* $L' = L \cap G'$ *is at most as complicated in the Chomsky hierarchy as* L*. Furthermore there is a finite transducer which can translate any* $x \in G'$ *into some* $y \in G$ *with* $y \sim x$*. Thus if* L' *is regular or context-free, so is* L*. As the mapping from* $y \in G$ *to the* $x \in G'$ *with* $x \sim y$ *is automatic, one can show that if* L' *is recognised by a linear-bounded automaton, so is* L*: The new linear-bounded automaton first executes* f *and then checks whether the image is in* L*. Thus if* L *is context-sensitive so is* L*. Thus one has that for a set of group elements in the free group, the level in the Chomsky hierarchy occupied by the representatives of this set does not depend on how one represents the corresponding free group as an automatic free group.*

The ET0L languages form a full abstract family of languages and thus it is also closed under images of transducers, even if these map every element of the original language to an infinite regular set of elements; therefore the previous paragraph's argument also applies to ET0L languages [8]. As every pattern language over the free group is an ET0L language in some automatic representation [7], this is also true for all automatic representations of every noncommutative free group. The authors are grateful to the anonymous referees for pointing this out.

Note that Theorem 4 does not say that $L \subseteq G$ is regular iff $L' = \{v \in \Sigma^{@} : \exists w \in L\, [v \sim w]\}$ is regular. This is indeed false for all non-empty finite subsets $L \subset G$: Such a set is regular although the corresponding L' is not regular. Given a word $w \in L$, it cannot be decided by a DFA whether $wa^i a^{-j} \in L'$ due to the fact that a finite automaton, after having processed wa^i, does not remember the exact value of i and therefore cannot compare it to j.

3 There are only two context-free verbal languages

The main result of this section is the characterisation of verbal languages of finitely generated free groups with at least two generators. Suppose that Σ is a non-empty finite set of generators containing $\{a, b\}$ and G is the set of reduced words over $\Sigma \cup \Sigma^{-1}$.

Suppose L is a context-free language generated by a Chomsky normal form grammar (not containing any useless symbols). Call a nonterminal A to be self-producing if there exist x, y such that $xy \neq \varepsilon$ and $A \Rightarrow^+ xAy$. Note that if a $A \Rightarrow^+ xAy$ (respectively, $A \Rightarrow^+ xA$ or $A \Rightarrow^+ Ay$), for non-empty x and y, then there exist non-empty x_A, y_A of length bounded by a constant c (depending on grammar for L) such that, for all h, $A \Rightarrow^+ (x_A)^h A (y_A)^h$ (respectively, $A \Rightarrow^+ (x_A)^h A$ or $A \Rightarrow^+ A(y_A)^h$). Thus, in any derivation of a string w in L using the grammar, for any self-producing A used in the derivation leading to $A \Rightarrow^+ u$ (where u is a substring of w), one could replace the corresponding u in w by $(x_A)^h u (y_A)^h$ (respectively, $(x_A)^h u$ or $u(y_A)^h$), for any $h \geq 0$, and still get strings in the language L. Thus, using standard techniques of pumping lemma proofs, the following version of the pumping lemma can be proven.

Proposition 6. *Suppose L is a context-free language. Then there exists a constant s such that any string $w \in L$ can be partitioned into $w_1 w_2 \dots w_k w_{k+1}$ (where each w_i is a string over $\Sigma \cup \Sigma^{-1}$) such that, for some strings $v_1, v_2, \dots, v_k$ and numbers $e_1, e_2, \dots, e_k$, the following statements hold:*

(a) *$|w_i| \leq s$, for i with $1 \leq i \leq k+1$ (some of the w_i's may be ε);*

(b) *For $i = 1, \dots, k$, $1 \leq |v_i| \leq s$;*

(c) *If $e_i = e_{i'}$ and $e_j = e_{j'}$, for some i, j, i', j' with $1 \leq i < i' \leq k$ and $1 \leq j < j' \leq k$, then either $i = j, i' = j'$ or $i < i' < j < j'$ or $j < j' < i < i'$ or $i < j < j' < i'$ or $j < i < i' < j'$;*

(d) *for all $h_{e_1}, h_{e_2}, \dots, h_{e_k} \in \mathbb{N}$, $\;w_1(v_1)^{h_{e_1}} w_2 (v_2)^{h_{e_2}} \dots (v_{k-1})^{h_{e_{k-1}}} w_k (v_k)^{h_{e_k}} w_{k+1} \in L$.*

In the above proposition, if $e_i = e_{i'}$ for distinct i, i', then the portion v_i and $v_{i'}$ are said to be linked (that is, they are pumped same number of times). Part (c) says that any two different linked pairs $v_i, v_{i'}$ and $v_j, v_{j'}$ do not have a cross-over: either both members of one pair are less than both members of the other pair or they nest in the way that both members of one pair are in between the two members of the other pair.

Intuitively, the proposition says that any word w in L can be split into $w_1 w_2 \dots w_k w_{k+1}$ such that each w_i is of length at most s and between w_i and w_{i+1} one can insert v_i pumped h_{e_i} many times. Here h_{e_i} and h_{e_j} are required to be same only if $e_i = e_j$, that is, they are linked.

Note that this pumping lemma only allows to pump up, not to pump down; however, this is sufficient for the way it is used in the present work. Furthermore, the interested reader might gain some additional insight into this pumping lemma by looking at the following proof sketch.

Let a context-free grammar for L in Chomsky Normal Form be given. Call a non-terminal A in this grammar self-deriving iff one can derive from it a terminal word such that, in the derivation, A does not only appear at the start but also in some intermediate step. Now let s be so large that (a) $\log(s)$ is strictly larger than the number of non-terminals in the grammar and (b) for every self-deriving non-terminal A there exist terminal words u_A, v_A with $A \Rightarrow^+ u_A A v_A$ and $1 \leq |u_A v_A| \leq s$ and both u_A, v_A being non-empty whenever one can choose them such. Given now a derivation of a word w in the given grammar, one can for every self-deriving A in the derivation insert additional derivation steps which replace A by $u_A A v_A$ h times for some $h \geq 0$. These modifications allow to derive the desired pumped versions for any word w in the language which is longer than s.

Theorem 7. *Consider a group H generated by a non-empty finite set of generators $\Sigma \supseteq \{a, b\}$, where $a \neq b$. Let G be a regular set of representatives for H.*

A verbal language $L \subseteq G$ is context-free iff either $L \equiv \{\varepsilon\}$ or $L = G$.

Proof. The proof technique is similar to techniques applied by Merzlyakov [25]. Note that by Theorem 4, it suffices to show the theorem for G being the set of reduced words over $\Sigma \cup \Sigma^{-1}$. Clearly $\{\varepsilon\}$ and G are context-free languages, indeed they are even regular.

For a proof of the converse direction, suppose $\pi = x_{i_1}^{j_1} x_{i_2}^{j_2} \dots x_{i_\ell}^{j_\ell}$ is a pattern using only variables $x_1, x_2, \dots, x_n$ (which all occur in π), where $j_1, j_2, \dots, j_\ell \in \{-1, +1\}$. Without loss of generality it can be assumed that all neighbouring parts $x_{i_k}^{j_k} x_{i_{k+1}}^{j_{k+1}}$ satisfy $i_k \neq i_{k+1}$ or $j_k = j_{k+1}$, as every subpattern of the form $x_{i_k}^{j_k} x_{i_k}^{-j_k}$ can be omitted from the pattern without changing the language generated. Suppose $L = \{red(w) : w \in Lang(\pi)\}$. It is now shown that L is either $\{\varepsilon\}$ or G.

Let b^+ denote $\{b, b^2, b^3, \dots\}$ and b^- denote $\{b^{-1}, b^{-2}, b^{-3}, \dots\}$.

Definition 8. The following will be used in the proof below.

- Let $r = 400 \cdot (2\ell + 1)^{2\ell}$.
- For i with $1 \leq i \leq n$, let $R_i = a^i\, b\, a^{ri+1}\, b^+\, a^{ri+2}\, b^+\, a^{ri+3}\, b^+ \ldots a^{ri+(r-1)}\, b\, a^i$ and let $R_i^{-1} = a^{-i}\, b^{-1}\, a^{-r(i+1)+1}\, b^-\, a^{-r(i+1)+2}\, b^-\, a^{-r(i+1)+3}\, b^- \ldots a^{-ri-1}\, b^{-1}\, a^{-i}$. Note that, $R_i^{-1} = \{x^{-1} : x \in R_i\}$.
- For i with $1 \leq i \leq n$ and $s > 0$, let $\alpha_{i,s} = a^i\, b\, a^{ri+1}\, b^s\, a^{ri+2}\, b^s\, a^{ri+3}\, b^s \ldots a^{ri+(r-1)}\, b\, a^i$.
- Let $R \subseteq G$ be such that $R \equiv R_{i_1}^{j_1} R_{i_2}^{j_2} \ldots R_{i_\ell}^{j_\ell}$.

Note that the only cancellations which can occur when forming the reduced words in the above definition of R are those cancellations which occur when R_i and $R_{i'}^{-1}$ are neighbouring; note that in this case $i \neq i'$. Then the words in R_i are of the form a^ibuba^i and those in $R_{i'}^{-1}$ are of the form $a^{-i'}b^{-1}u'b^{-1}a^{-i'}$ which implies that the words in $R_i \cdot R_{i'}^{-1}$ are of the form $a^ibuba^{i-i'}b^{-1}u'b^{-1}a^{-i'}$ and those in $R_{i'}^{-1} \cdot R_i$ are of the form $a^{-i'}b^{-1}u'b^{-1}a^{i-i'}buba^i$. Thus the cancellations in the concatenations are always limited to the first and last a^i / a^{-i} of the R_i and R_i^{-1} of each component. It follows that the set R defined above is indeed regular.

Claim 9. *For all $s > 0$, $red(\alpha_{i_1,s}^{j_1}\alpha_{i_2,s}^{j_2} \ldots \alpha_{i_\ell,s}^{j_\ell})$ is in $L \cap R$.*

Clearly, $red(\alpha_{i_1,s}^{j_1}\alpha_{i_2,s}^{j_2} \ldots \alpha_{i_\ell,s}^{j_\ell}) \in R$. By using the substitution, $x_i = \alpha_{i,s}$, it follows that $red(\alpha_{i_1,s}^{j_1}\alpha_{i_2,s}^{j_2} \ldots \alpha_{i_\ell,s}^{j_\ell}) \in L$. Thus the claim follows.

The following claim follows using Proposition 6 and noting that $L \cap R$ is context-free and that the number of a's in any member of R is fixed, and thus pumping can happen only in the b^+ part.

Claim 10. *There is a constant $s > 0$ such that, for $w = red(\alpha_{i_1,s}^{j_1}\alpha_{i_2,s}^{j_2} \ldots \alpha_{i_\ell,s}^{j_\ell})$, the following is satisfied.*

w can be partitioned into $w_1w_2 \ldots w_kw_{k+1}$ such that $w_1, w_2, \ldots, w_k, w_{k+1}$ are of length at most s and there exist numbers $e_1, e_2, \ldots, e_k$ and constants $c_1, c_2, \ldots, c_k > 0$ such that

(a) if $e_i = e_{i'}$ and $e_j = e_{j'}$ for i, i', j, j' with $1 \leq i < i' \leq k$ and $1 \leq j < j' \leq k$, then either $i = j \wedge i' = j'$ or $i < i' < j < j'$ or $j < j' < i < i'$ or $i < j < j' < i'$ or $j < i < i' < j'$.

(b) for all non-negative integers $h_j, j \in \{e_1, e_2, \ldots, e_k\}$, the word $w_1b^{c_1 \cdot h_{e_1}}w_2b^{c_2 \cdot h_{e_2}} \ldots w_kb^{c_k \cdot h_{e_k}}w_{k+1}$ is in $L \cap R$.

By taking $h_{k'} = \tilde{c}^{k'}$ for a sufficiently large constant $\tilde{c}$, one obtains that in

$$w' = w_1b^{c_1 \cdot h_{e_1}}w_2b^{c_2 \cdot h_{e_2}} \ldots w_kb^{c_k \cdot h_{e_k}}w_{k+1},$$

the ts in any substrings of the form $ba^{ri+p}b^ta^{ri+p+1}b$ or of the form $b^{-1}a^{-ri-p-1}b^{-t}a^{-ri-p}b^{-1}$, have different absolute value unless the corresponding segments of b^+ are linked, that is the corresponding $c_j = c_{j'}$ and $e_j = e_{j'}$.

As w' is in L, there are substitutions $y_1, \ldots, y_n$ (which are reduced words) of $x_1, \ldots, x_n$ in π which generates w'. Let Y be the set of all reduced words which are congruent to some word formed by concatenating up to ℓ copies of $y_1, y_1^{-1}, \ldots, y_n, y_n^{-1}$. Note that Y has at most $(2n + 1)^\ell$ elements. The goal of the following is to show that there exists a p such that for each i, if $ba^{ri+p}b^ta^{ri+p+1}b$ or $b^{-1}a^{-ri-p-1}b^{-t}a^{-ri-p}b^{-1}$ is a substring of some y_j, then it is either completely cancelled or completely survives in w' within the substitution of y_j (that is, it is not formed by concatenating the substituted parts of several variables).

For this, one declares p and $p + 1$ to be invalid if the following occurs:

- There are words of the form uu' and $u''u'''$ in Y such that their concatenation $uu'u''u'''$ equals to uu''' after cancellations and either $ba^{ri+p}b^t a^{ri+p+1}b$ or its inverse is a substring of uu' or $u''u'''$ or uu''' in a way that it spans over both parts of that word (that is, touches u and u' or u'' and u''' or u and u''', respectively) or one of $ba^{ri+p}b$, $ba^{ri+p+1}b$, $b^{-1}a^{-ri-p}b^{-1}$, $b^{-1}a^{-ri-p-1}b^{-1}$ is a substring of uu''' in a way that it spans over both parts of uu'''.

- For some t and $v = ba^{ri+p}b^t a^{ri+p+1}b$, there are either two occurrences of v in w' or two occurrences of v^{-1} in w'.

The first case above can happen for at most $100 \cdot |Y|^2$ many p: For each pair $(uu', u''u''')$ there is only one unique way to split the first word into u and u' and the second word into u'' and u''' such that u' and u'' are the parts which cancel out. Having these, one can see that there are only constantly many p's which get invalidated by the first case, and 100 is a safe upper bound on the number of such possibilities.

The second case can happen only if there are two linked segments b^t of this form and, furthermore, these two segments must be generated by two instances of x_i or two instances of x_i^{-1} (but not by one instance of x_i and one instance of x_i^{-1}). This happens for each pair of occurrences of x_i for at most one p (as the linked segments cannot be linked in a overlapping way by Claim 10 (a)). Hence for each pair of occurrences of variables in π, there are only two values of p invalidated and so the overall number of values of p invalidated by this process is $2\ell^2$.

In total at most $100 \cdot (2n+1)^{2\ell} + 2\ell^2$ possible values of p are invalidated. Hence there is a choice of p such that neither p nor $p+1$ is invalidated and $1 \leq p < r-2$. Taking r as $400 \cdot (2\ell+1)^{2\ell}$ as done in Definition 8 gives a safe value for r to guarantee the existence of a p such that neither p nor $p+1$ have been invalidated. There are now two cases.

Case 1: For every occurrence in w' of $ba^{ri+p}b^t a^{ri+p+1}b$ there is also an occurrence of $b^{-1}a^{-ri-p-1}b^{-t}$ $a^{-ri-p}b^{-1}$ in w'.

Then the corresponding segments of b^t and b^{-t} are linked. It follows that one can link each occurrence of x_i to an occurrence of x_i^{-1} in π according to the linkage of these segments and the linkages are never overlapping in the sense that $x_{i_k}^{j_k}$ is linked to $x_{i_{k''}}^{j_{k''}}$ and $x_{i_{k'}}^{j_{k'}}$ is linked to $x_{i_{k'''}}^{j_{k'''}}$ with $k < k' < k'' < k'''$. Hence there must be two neighbouring $x_{i_k}^{j_k}$ and $x_{i_{k+1}}^{j_{k+1}}$ which are linked, that is, $i_k = i_{k+1}$ and $j_k = -j_{k+1}$ in contradiction to the assumption at the beginning of this proof. Hence Case 1 does not occur.

Case 2: Some substring $v = ba^{ri+p}b^t a^{ri+p+1}b$ or v^{-1} occurs exactly once in w'.

By the choice of p, it follows that when forming the word w' as $red(y_{i_1}^{j_1} \ldots y_{i_\ell}^{j_\ell})$, the number of occurrences of v in w' is the sum over h of j_h times the number of occurrences of v in y_{i_h}, where occurrences of v^{-1} count negatively. Hence, for $h = 1, 2, \ldots, n$, choosing z_h to be the concatenation of all occurrences of v and v^{-1} in y_h, one obtains that $red(z_{i_1}^{j_1} \ldots z_{i_\ell}^{j_\ell})$ equals v. Hence, substituting any reduced word β for v in the above definition of z_h, it follows that β is generated by π. Thus, $Lang(\pi) \equiv H$. This completes the proof of Theorem 7. ∎

Example 11. *Consider the free group generated by $\Sigma = \{a, b\}$. The set of reduced words generated by the pattern $x_1 a x_1^{-1}$ is context-free but not regular.*

For a substitution $sub(x_1) = u$, where u is reduced and does not end with a or a^{-1}, the resulting word uau^{-1} is generated in a cancellation-free way. For a substitution $sub(x_1) = ua^k$, where u is reduced and does not end in a or a^{-1} and $k \in \mathbb{Z}$, the generated word is uau^{-1}. Therefore the set of reduced words generated by $x_1 a x_1^{-1}$ is $\{uau^{-1} : u$ is reduced and does not end in a or $a^{-1}\}$. This set is generated by the following grammar with starting symbol being S:

$S \to a|aAa^{-1}|bBb^{-1}|a^{-1}Ca|b^{-1}Db$,
$A \to aAa^{-1}|bBb^{-1}|b^{-1}Db$,
$B \to aAa^{-1}|a^{-1}Ca|bBb^{-1}|a$,
$C \to a^{-1}Aa|bBb^{-1}|b^{-1}Db$,
$D \to aAa^{-1}|a^{-1}Ca|b^{-1}Db|a$.

An easy application of the pumping lemma for regular languages shows that $\{uau^{-1} : u$ does not end in a or $a^{-1}\}$ is not regular.

Theorem 12. *Suppose H is a free group generated by generators $\Sigma \supseteq \{a, b\}$, where $a \neq b$. Suppose G is any representation of H.*

A pattern language over H (in the representation G) is regular iff it is either a singleton or G.

Proof. Note that by Theorem 4, it suffices to consider L containing reduced words over $\Sigma \cup \Sigma^{-1}$.

One direction of the theorem is obvious: A pattern consisting only of constants generates a singleton language and every singleton can be generated this way. A pattern consisting of exactly one variable generates G. G as well as the singleton languages are clearly regular.

The proof for the other direction is an adjustment of the proof Theorem 7. Assume that $L \subseteq G$ is generated by pattern $\pi = x_{i_1}^{j_1} x_{i_2}^{j_2} \ldots x_{i_\ell}^{j_\ell}$ with $j_1, j_2, \ldots, j_\ell \in \{-1, +1\}$ and the $x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}$ being either variables or constants from Σ — the latter are also called x_i in this proof in order to simplify the notation. Furthermore, assume that L is regular and contains at least two strings. Then π contains at least one variable. Now Definition 8 will be adjusted as below; the main changes are that the beginnings and endings of R_i have now $a^{\ell i}b$ in place of a^i, and that r is taken to be larger by a factor $2\ell + 1$ in order to accommodate this change; furthermore, the concept of constants is incorporated.

Definition 13. Below, the following definitions are used.

- Let $r = 400 \cdot (2\ell + 1)^{2\ell+1}$.
- If x_i is a variable in π then let
  $R_i = a^{\ell i}\ b\ a^{ri+1}\ b^+\ a^{ri+2}\ b^+\ a^{ri+3}\ b^+ \ldots a^{ri+(r-1)}\ b\ a^{\ell i}$ and let
  $R_i^{-1} = a^{-\ell i}\ b^{-1}\ a^{-r(i+1)+1}\ b^-\ a^{-r(i+1)+2}\ b^-\ a^{-r(i+1)+3}\ b^- \ldots a^{-ri-1}\ b^{-1}\ a^{-\ell i}$.
  If x_i is in Σ then let $R_i = x_i$ and $R_i^{-1} = x_i^{-1}$.
- For $i = 1, \ldots, n$, x_i being a variable and $s > 0$, let
  $\alpha_{i,s} = a^{\ell i}\ b\ a^{ri+1}\ b^s\ a^{ri+2}\ b^s\ a^{ri+3}\ b^s \ldots a^{ri+(r-1)}\ b\ a^{\ell i}$;
  If x_i is a constant, then let $\alpha_{i,s} = x_i$.
- Let R be the regular language of the reduced words in $R_{i_1}^{j_1} R_{i_2}^{j_2} \ldots R_{i_\ell}^{j_\ell}$.

If x_{i_k} is a variable then the cancellations in this expressions target only the a or a^{-1} at the beginning or end of the R_i or R_i^{-1}.

As a representative example, suppose that in π non-constant x_{i_k} and $x_{i_{k'}}^{-1}$ occur with only constants between the k-th and k'-th position, $k < k'$. If $k' = k + 1$ (that is there are no constants between x_{i_k} and $x_{i_{k'}}$), then $i_k \neq i_{k'}$ and the ending $ba^{i_k \ell}$ of R_k and the start $a^{-i_{k'}\ell}b^{-1}$ of $R_{i_{k'}}^{-1}$ do not cancel each other out. If $k' > k+1$ and all symbols in between are either a or a^{-1}, then they must all be either a or all a^{-1}. Now $i_{k'} - i_k$ is strictly between $-\ell$ and ℓ and different from 0. Hence the balance of the a and a^{-1} between the last b of R_{i_k} and the first b^{-1} of $R_{i_{k'}}^{-1}$ is not void and hence only a and a^{-1} are cancelled out. If not all constants between R_{i_k} and $R_{i_{k'}}^{-1}$ are a or a^{-1}, some of them survive the cancelling and also some of the a after the last b of R_{i_k} and some of the a^{-1} before the first b^{-1} of $R_{i_{k'}}^{-1}$ survive cancellations; hence again the last b of R_{i_k} and the first b^{-1} of $R_{i_{k'}}^{-1}$ do not get cancelled. The other cases which can occur are similar and in each case it does not happen that any b or b^{-1} in any of the nonconstant R_{i_k} is involved in cancellations. This will be used in the following considerations.

Thus, the counterpart of Claim 9 holds which states that, for $s > 0$, $w = \alpha_{i_1,s}^{j_1}\alpha_{i_2,s}^{j_2}\ldots\alpha_{i_\ell,s}^{j_\ell}$ is in $R \cap L$. Choose s larger than the pumping constant of $R \cap L$ (which is a regular language). Now, by repeated pumping, the word w can be modified to obtain a word $w' \in R \cap L$ such that there is no $t > s$ for which there are two subwords of w' starting at different positions in w' and of the form ab^ta or $a^{-1}b^{-t}a^{-1}$. The above fact is a counterpart of Claim 10.

As $w' \in L$, there are $y_i \in H$ such that $y_i = x_i$ in the case that x_i is a constant and $w' = red(y_{i_1}^{j_1}\ldots y_{i_\ell}^{j_\ell})$. Let Y be the set of all reduced words which are congruent to some word formed by taking concatenations of up to ℓ many y_i. Now, one proceeds as in the proof of Theorem 7 and declares numbers $p, p+1$ $(1 \leq p \leq r-2)$ to be invalid in the case that there are words uu' and $u''u'''$ in Y such that their concatenation $uu'u''u'''$ equals to uu''' after cancellations and either $ba^{ri+p}b^ta^{ri+p+1}b$ or its inverse is a substring of uu' or $u''u'''$ or uu''' in a way that it spans over both parts of that word (that is, touches u and u' or u'' and u''' or u and u''', respectively) or one of $ba^{ri+p}b$, $ba^{ri+p+1}b$, $b^{-1}a^{-ri-p}b^{-1}$, $b^{-1}a^{-ri-p-1}b^{-1}$ is a substring of uu''' in a way that it spans over both parts of uu'''. As shown in Theorem 7 there is a p such that $1 \leq p \leq r-2$ and neither p nor $p+1$ are declared to be invalid. Fix a t such that $u_1 = ba^{ri+p}b^ta^{ri+p+1}b$ or u_1^{-1} occurs in w'. By the choice of w' there are u_0, u_2 with $w' = u_0u_1u_2$ (respectively, $u_0u_1^{-1}u_2$) and these u_0, u_2 are unique. Now, given any reduced word v, replace in y_i all occurrences of u_1 by $u_0^{-1}vu_2^{-1}$ (respectively, $u_0^{-1}v^{-1}u_2^{-1}$) to obtain a word z_i. Due to p not being invalid, $red(z_{i_1}^{j_1}z_{i_2}^{j_2}\ldots z_{i_\ell}^{j_\ell}) = v$. Thus, it follows that L is the set of all reduced words in H. ∎

Existentially first-order definable languages are given by a set of patterns $\pi_1, \ldots, \pi_n$ in variables $x_1, \ldots, x_m$ such that $w \in L$ iff there is a substitution sub for the variables such that $w \sim sub(\pi_1) \sim \ldots \sim sub(\pi_n)$. For the class of existentially first-order definable languages, the following example exhibits languages which are regular and different from singleton languages and the full group.

Example 14. *There are non-trivial regular and properly context-free existentially first-order definable languages. Examples are the languages $(aa)^@$ and $a^@b^@$ in the case of regular languages and $\{b^nab^n : n \in \mathbb{Z}\}$ in the case of properly context-free languages. This can be seen as follows:*

1. $(aa)^@ = \{w : \exists x_1\,[w \sim x_1x_1aa \wedge w \sim aax_1x_1]\}$;
2. $a^@b^@ = \{w : \exists x_1\exists x_2\,[w \sim ax_1bx_2 \wedge w \sim x_1abx_2 \wedge w \sim ax_1x_2b \wedge w \sim x_1ax_2b]\}$;
3. $\{b^nab^n : n \in \mathbb{Z}\} = \{w : \exists x_1\,[w \sim x_1bax_1b \wedge w \sim bx_1abx_1]\}$.

Remark 15. It would be interesting to study the exact relationship between indexed languages and verbal/pattern languages over groups. Aho [1] introduced indexed languages as a generalisation of context-free languages; the indexed languages are a subclass of the context-sensitive languages. The conference version of the present work asked whether in free groups with at least two generators, all verbal languages are indexed languages. Ciobanu, Diekert and Elder answered this question explicitly: They showed that in the free group with at least two generators and in the standard presentation $(G, \circ)$ by reduced words, all languages defined by equations over variables are EDT0L languages [7]; the verbal languages and the pattern languages are a special case of these and thus also EDT0L in this presentation. The EDT0L languages are a proper subclass of the ET0L languages [3, 9, 10] and the ET0L languages are closed under mappings defined by finite state transducers [3, 30]. Thus, by a proof similar to that of Lemma 3 and Theorem 4, one can show, for every given automatic unique representation G'' of a free group H and every language $L'' \subseteq G''$, that a finite state transducer translates L'' into a copy $L' \subseteq G'$ for an automatic representation G' using the standard generators of the group H. Furthermore, another finite state transducer translates this language L' into an equivalent $L \subseteq G$, where G is the set of reduced words over the standard generators. The inverses of both of the above translations can also be computed by finite state transducers. Thus L'' is an ET0L language iff L is. Holzer, Jakobi and McQuillan [16] mention that every ET0L language is an indexed language and thus every pattern language over a free group with at least two generators is an indexed language.

4 Abelian automatic groups and hyperbolic groups

In a free commutative group over two generators, the elements are often permuted in order to move them into a normal form. For example, every word may be permuted to a congruent word with least lexicographic order among all permutations of the word. Thus, for $x = a^n b^m$, xx is normalised to $a^{2n} b^{2m}$. This seems to suggest that the pattern languages in free groups might be difficult. However, the opposite is the case and the following can be shown: in every free commutative group generated by a finite set of generators, all pattern languages are regular; this result holds independently of the chosen representation.

Proposition 16. *Let H be a free Abelian group with Σ being a finite set of generators. Let G be a set of representatives of H and let $L \subseteq G$ be a pattern language. Then L is regular independently of the chosen set of representatives and the finite set of generators.*

Proof. For ease of notation assume that $\Sigma = \{a, b, c\}$. The proof can be easily generalized to any other set of generators. Note that by commutativity, the rules $ab \sim ba$, $ac \sim ca$ and $bc \sim cb$ hold.

Suppose $L \subseteq G$ is a pattern language generated by a pattern π. As the group is commutative, without loss of generality it can be assumed that π is of the form $a^i b^j c^k x_1^{\ell_1 \cdot s} x_2^{\ell_2 \cdot s} \ldots x_r^{\ell_r \cdot s}$, where i, j, k are integers, $x_1, x_2, \ldots, x_r$ are variables, and $s, \ell_1, \ldots, \ell_r$ are natural numbers with greatest common divisor of $\ell_1, \ldots, \ell_r$ being 1.

Let $\pi' = a^i b^j c^k x^s$, where x is a variable. Now it is shown that $Lang(\pi) \equiv Lang(\pi')$. To see this note that π can be obtained from π' by using the substitution $sub(x) = x_1^{\ell_1} x_2^{\ell_2} \ldots x_r^{\ell_r}$. Thus, the language generated by π is contained in the language generated by $a^i b^j c^k x^s$. On the other hand, let $m_1, m_2, \ldots, m_r$ be integers such that $\sum_{1 \leq p \leq r} m_p \ell_p = 1$ (note that there exists such $m_1, m_2, \ldots, m_r$ as the greatest common divisor of $\ell_1, \ldots, \ell_r$ is 1). Then, π' can be obtained from π by using the substitution $sub(x_p) = x^{m_p}$. Thus, $Lang(\pi) \equiv Lang(\pi')$. It follows that $L = a^i (a^s)^{@} b^j (b^s)^{@} c^k (c^s)^{@}$, which is regular.

Now suppose G' is another representation of H using generators Σ', and $L' \subseteq G'$ is such that $L' \equiv L$. For $d \in \Sigma'$, suppose $d \equiv a^{i_d} b^{j_d} c^{k_d}$. Then, $d_1 d_2 \dots d_\ell \in L'$ iff

$$i_{d_1} + i_{d_2} + \ldots + i_{d_\ell} = i \mod s;\; j_{d_1} + j_{d_2} + \ldots + j_{d_\ell} = j \mod s;\; k_{d_1} + k_{d_2} + \ldots + k_{d_\ell} = k \mod s.$$

The above can be easily checked by a DFA, and thus L' is regular. ∎

Suppose H and F are two groups. A direct product of the groups H and F, denoted by HF, is the group with members (w, e), where $w \in G$ and $e \in F$. The group operation $\circ$ is defined by $(w, e) \circ (w', e') = ((w \circ w'), (e \circ e'))$, where $\circ$ denotes the group operation in the corresponding groups.

Proposition 16 can be generalised to all Abelian automatic groups. Such a group is a direct product of a free Abelian automatic group H and a finite group $F = \{e_1, e_2, \ldots, e_h\}$. Consider a representation G of H using some finite set of generators, where it is assumed without loss of generality that G does not intersect with H. The group members of HF can be represented as strings in a representation G of H followed by one of the symbols $e_1, e_2, \ldots, e_h$. Given a pattern language L in HF, there exist pattern languages $L_1, \ldots, L_h$ over G such that $L = L_1 \cdot e_1 \cup L_2 \cdot e_2 \cup \ldots \cup L_h \cdot e_h$. By Proposition 16 it follows that each of $L_1, L_2, \ldots, L_h$ is regular; thus L is regular. Again using methods similar to that in the proof of Proposition 16, it can be shown that L is also regular in other automatic presentations of the group GF. This gives the following general theorem.

Theorem 17. *Every pattern language in an Abelian automatic group is regular.*

Theorem 4 has shown that for two representations G and G' of the same free group over a finite alphabet Σ and for each $L \subseteq G$ and $L' \subseteq G'$ with $L' = \{w \in G' : \exists v \in L\,[v \sim w]\}$, L is regular iff L' is regular and L is context-free iff L' is context-free. While pattern languages over a free Abelian group are regular over all representations, this is not true for arbitrary regular languages $L \subseteq G$. For an example, suppose G is the free commutative group with generators a, b, c and $G = a^@ b^@ c^@$ and $G' = (ab^{-1})^@ (bc^{-1})^@ c^@$. Then the regular language $L = a^@ \subseteq G$ corresponds to the properly context-sensitive language $L' = \{(ab^{-1})^n (bc^{-1})^n c^n : n \in \mathbb{Z}\}$ in G'. The next sections will investigate groups (which are not free groups) where such witnesses can be found in the form of pattern languages and verbal languages.

Hyperbolic groups were introduced by Gromov [13] and have turned out to be an important class in group theory. Though free Abelian groups with two or more generators are not hyperbolic, there are still prominent examples of hyperbolic groups for which Theorem 12 fails.

Example 18. *The free group with one generator a is always Abelian and is also a hyperbolic group. The language $(aa)^@$ generated by the pattern $x_1 x_1$ is a regular verbal language which is non-trivial (i.e., not equivalent to $\{\varepsilon\}$ or the whole group).*

The next result shows that Theorem 12 also fails for some hyperbolic groups which are non-elementary, that is, which have the free group with two generators as a subgroup [13]. Note that finite languages are regular in all representations of an automatic group and that, by Theorem 12, in the case of free group with two or more generators, all finite pattern languages contain exactly one element.

Theorem 19. *There is a non-elementary hyperbolic group in which some pattern languages consist of 2 and 3 elements, respectively.*

Proof. Consider the group H generated by three generators a, b, c, where the subgroup generated by a, b is free and where the following additional rules hold for the generator c: $c^{-1} = c^2$, $ca = ac^{-1}$, $cb = bc^{-1}$ (and thus, $ca^{-1} = a^{-1}c^{-1}, cb^{-1} = b^{-1}c^{-1}$, $c^{-1}a = ac, c^{-1}b = bc$, $c^{-1}a^{-1} = a^{-1}c, c^{-1}b^{-1} = b^{-1}c$ and $c^{-1} = cc$).

As the free group with generators $\{a, b\}$ is a subgroup of H with index 3, the group H is a non-elementary hyperbolic group [13].

Let $G = \{w, wc, wc^{-1} : w$ is a reduced word over generators $a, b\}$. It is easy to verify that G represents all possible elements of H in a unique way and that G is automatic.

Now consider the pattern language generated by $\pi = x_1cx_1^{-1}$. Consider any reduced word w over the generators $\{a, b\}$. Due to commutativity between the powers of c, $wcw^{-1} \sim (wc)c(wc)^{-1} \sim (wc^{-1})c(wc^{-1})^{-1}$. Due to the rules that moving a, a^{-1}, b, b^{-1} over c replaces c by c^{-1} and vice versa, it follows that $wcw^{-1} \sim ww^{-1}c \sim c$ for w of even length and that $wcw^{-1} \sim ww^{-1}c^{-1} \sim c^{-1}$ for w of odd length. Hence the pattern π generates the language $\{c, c^{-1}\}$, which consists of two elements.

Furthermore, from above analysis, it follows that the pattern $x_1cx_1^{-1}x_2cx_2^{-1}$ generates the language $\{c, c^{-1}\} \cdot \{c, c^{-1}\} \equiv \{\varepsilon, c, c^{-1}\}$ which consists of three elements. ∎

Thus, an interesting question is whether there are hyperbolic groups, different from the free groups, with at least two generators for which Theorem 12 holds. In the proof of Theorem 19, the property $ca = ac^{-1}$ was essential. If an extension by using a direct product with a finite commutative group is considered, then the resulting new group satisfies Theorem 12.

Theorem 20. *Assume that H is an automatic group and that F is a commutative finite group. Furthermore assume that for every automatic presentation G' of H and for every pattern π with constants from G' the following statements hold:*

1. *π generates G' iff there exists a substitution $sub(x_i) = v_iz^{j_i}w_i$ such that $red(sub(\pi)) = z$, where z is a variable and $v_i, w_i \in G'$ are constants and j_i is an integer (depending on x_i);*

2. *π generates a singleton language iff $red(\pi)$ does not contain variables;*

3. *If the language generated by π (in the representation G') is neither G' nor a singleton, then the language generated by π (in the representation G') is not regular.*

Then the corresponding statements are true for the direct product HF of H and F.

Proof. Let G be a representation of H using reduced words over a non-empty finite set of generators Σ. Without loss of generality assume that $G \cap F = \emptyset$.

The set GF consists of all words of the form we with $w \in G$ and $e \in F$ and the group operation $\circ$ is defined by $(we) \circ (w'e') = (w \circ w') \cdot (e \circ e')$, where $\circ$ denotes the group operation in the corresponding groups and $\cdot$ is the concatenation. The generators of HF is the union of the generators of G and of F. It is easy to see that HF is an automatic group using the representation for GF.

Given a pattern π, consider the two projected patterns π_G and π_F formed by deleting the constants from F and G respectively. Now, the pattern language L generated by π (in the representation GF) satisfies that $L = L_G \cdot L_F$ for the languages $L_G \subseteq G$ generated by π_G and $L_F \subseteq F$ generated by π_F. Now consider the following three cases:

Case 1: $L_G = G$: Thus, there exists a substitution $sub(x_i) = v_i z^{j_i} w_i$ of variables $x_1, x_2, \ldots$ in π such that $sub(\pi_G) = z$, where $v_i, w_i \in G$ and z is a variable. Thus, using the substitution $sub'(x_i) = y^{j_i}$, $sub'(\pi_F) = uy$, for some $u \in F$ (due to commutativity of F). Using a further substitution $sub''(y) = u^{-1}z$ for y, it follows that $sub''(sub'(\pi_F)) = z$. Thus the substitution, $sub'''(x_i) = u^{-1}v_i z^{j_i} w_i$ gives $sub'''(\pi) = z$, and hence π generates GF.

Case 2: $L_G = \{g\}$ for some $g \in G$: Then the occurrences of the variables cancel out in π_G. As F is commutative, the variables also cancel out in π_F, and thus π_F generates some singleton language $\{h\}$. Thus $L = \{gh\}$. Furthermore, the part h can be moved to the front in $red(\pi)$ and $red(\pi) = hred(\pi')$, where π' is obtained from π by replacing all constants $g'h'$ (with $g' \in G$ and $h' \in F$) by g' in π; as the variables in $red(\pi')$ cancel out, the same occurs in $red(\pi)$.

Case 3: L_G is not regular: The language L is the Cartesian product of L_G and the finite language L_F and thus not regular as well. Furthermore, this case applies if and only if there is neither a substition sub with $red(sub(\pi)) = z$ for some variable z nor $red(\pi)$ contracts to a constant.

The remaining case is to show the above properties for other automatic presentations K of GF. So, suppose such a presentation K is given and that $L_K = \{w \in K : \exists v \in L\,[w \sim v]\}$ be generated by a pattern π. As the criteria for the first two cases are on the form of π and as these are independent of the automatic representation of the group, one directly has that π generates K iff there is a substitution sub with $red(sub(\pi))$ being one single variable and π generates a singleton language iff $red(\pi)$ does not contain variables.

Now it is shown that L_K is not regular in the third case and one assumes that π is neither of the first nor the second form, that is, neither does a substitution reduce π to a single variable nor is the reduced form of π a constant. So assume by way of contradiction that L_K is regular due to the choice of K. Then a different representation G' of G is constructed and it is shown that in the representation G', $L_{G'} \equiv L_G$ would also be regular in contradiction to the assumption of the theorem, that in any automatic presentations of H, only patterns of the first or the second form can generate regular languages.

Now here are the details for the construction of this G'. Let $\min_{ll}(D)$ denote the length-lexicographical minimum of a nonempty subset D of K. Below, for ease of notation, for $w \in K$, $h \circ w$ is used to denote $h_K \circ w$, where h_K is the representative in K for h. Note that the function

$$f : w \mapsto f(w) = \min_{ll}\{h \circ w : h \in F\}$$

is an automatic function in representation K as there are only finitely many members h in F, the mappings $w \mapsto h \circ w$ is automatic for each $h \in F$ and the length-lexicographic minimum over these constantly many values computed from w by automatic functions is again an automatic function.

Let $G' = \{f(w) : w \in K\}$. Here, for $g \in G$ and $h \in F$, if $w' \in G'$ represents gh in K, then w' represents g in G'; this also holds for generators and one can see that if w' and w'' represent $g'h'$ and $g''h''$ in K then $f(w'w'')$ represents $g' \circ g''$ in G' and thus each word in G' represents the product of the elements represented by the letters under f in it. Note that, for $g \in G, h, h' \in F$, it holds that $f(gh) = f(gh')$. Thus, for each $g \in G$, there is exactly one representative of g in G'. Let $L' = \{f(w) : w \in L_K\}$.

As the function f is automatic, the set L' is also regular. Due to the choice of π, neither the first nor the second case holds. Furthermore, L' is generated by the pattern $f(\pi)$, where variables are mapped to variables, inverted variables are mapped to inverted variables and constants representing $gh \in G \circ H$

in K are mapped to an image representing g in G'. Now $f(\pi)$ is a pattern generating L' and therefore the third clause has to apply; so L' cannot be regular. Therefore the assumption that L_K is regular was wrong. ∎

Note that the first and second of the three clauses in Theorem 20 imply that in H, for all $k > 1$, $Lang(x^k) \not\equiv \{\varepsilon\}$ and $Lang(x^k) \neq\equiv H$. Without this condition, Theorem 20 becomes false. To see this, consider the finite groups G, F containing two elements and three elements, respectively. On one hand, in G, every verbal language is either $\{\varepsilon\}$ or G so that there is no non-trivial verbal language at all. On the other hand, in GF there are four different verbal languages which are all finite and thus regular. These four languages are given by the patterns x_1 generating GF, x_1x_1 generating F, $x_1x_1x_1$ generating G and ε generating $\{\varepsilon\}$.

5 When representations matter

In the previous sections it was shown that for free groups and free Abelian groups, for pattern languages being regular, context-free, context-sensitive or recursively enumerable, the automatic representation did not matter. This section shows that for non-free automatic groups, this is not the case. In particular this section provides an example where it depends strongly on the representation chosen whether certain verbal languages are regular, properly context-free or properly context-sensitive. The following definition fixes the group H and its representations G, G', G'' for this section.

Definition 21. Let H be a group with generators $\Sigma = \{a, b, c, d\}$, where the following additional constraints are put on the group operation:

(i) $a \sim a^{-1}$, $ab \sim b^{-1}a$, $ab^{-1} \sim ba$, $ac \sim ca$, $ad \sim da$ (thus, $ac^{-1} \sim c^{-1}a$, $ad^{-1} \sim d^{-1}a$);

(ii) b, c, d (and their inverses) commute with each other.

Let representations G, G', G'' be as follows:

$$G = \{\varepsilon, a\} \cdot b^{@} \cdot c^{@} \cdot d^{@},\ G' = \{\varepsilon, a\} \cdot (bcd)^{@} \cdot c^{@} \cdot d^{@} \text{ and } G'' = \{\varepsilon, a\} \cdot (bc)^{@} \cdot c^{@} \cdot d^{@}.$$

First it needs to be shown that H is indeed a group, where it suffices to show this for the representation G. It is easy to verify that each member of H has a unique representative in each of G, G', G'', with ε being the neutral element. For $h, h', h'' \in \{0, 1\}$ and $k, k', k'', m, m', m'', n, n', n'' \in \mathbb{Z}$, the product

$$(a^h b^k c^m d^n) \cdot (a^{h'} b^{k'} c^{m'} d^{n'}) \cdot (a^{h''} b^{k''} c^{m''} d^{n''})$$

is

$$a^{h+h'+h''-2hh'-2hh''-2h'h''+4hh'h''} \cdot b^{k(1-2h'-2h''+4h'h'')} \cdot b^{k'(1-2h'')+k''} \cdot c^{m+m'+m''} \cdot d^{n+n'+n''}$$

independent of the order of the group operations. Thus, the law of associativity holds. Furthermore, for $h \in \{0, 1\}$, $k, m, n \in \mathbb{Z}$, the inverse of $a^h b^k c^m d^n$ is $a^h b^{(2h-1)k} c^{-m} d^{-n}$. Thus, G satisfies the requirements for being a group.

A group is said to be biautomatic (with representation A) if for each $w \in A$, the functions $f_w : w' \mapsto w \circ w'$ as well as $g_w : w' \mapsto w' \circ w$ are automatic. If a group (with representation A) is automatic but not biautomatic, then it is called one-sided automatic.

Theorem 22. *The group H is biautomatic in representation G and one-sided automatic (but not biautomatic) in representations G' and G''.*

Proof. To see automaticity, let $x = a^h b^k c^m d^n$ with $h \in \{0,1\}$ and $k,m,n \in \mathbb{Z}$. Now the following equations gives the representatives of x and the multiples with the generators in G, G' and G'', respectively:

- $x \sim a^h b^k c^m d^n \sim a^h (bcd)^k c^{m-k} d^{n-k} \sim a^h (bc)^k c^{m-k} d^n$;
- $ax \sim a^{1-h} b^k c^m d^n \sim a^{1-h} (bcd)^k c^{m-k} d^{n-k} \sim a^{1-h} (bc)^k c^{m-k} d^n$;
- $bx \sim a^h b^{k+1-2h} c^m d^n \sim a^h (bcd)^{k+1-2h} c^{m-k-1+2h} d^{m-k-1+2h} \sim a^h (bc)^{k+1-2h} c^{m-k-1+2h} d^n$;
- $cx \sim a^h b^k c^{m+1} d^n \sim a^h (bcd)^k c^{m-k+1} d^{n-k} \sim a^h (bc)^k c^{m-k+1} d^n$;
- $dx \sim a^h b^k c^m d^{n+1} \sim a^h (bcd)^k c^{m-k} d^{n-k+1} \sim a^h (bc)^k c^{m-k} d^{n+1}$.

The multiplications with b^{-1}, c^{-1} and d^{-1} follow similar rules.

Note that multiplications from the left by a, b, c, d and their inverses moves each part ($a^@$, $b^@$, $c^@$, $d^@$, $(bcd)^@$, $(bc)^@$ in the various representations) by at most five symbols. Thus, H is automatic for each of the representations G, G', G''.

For multiplication from the right side in the representation G,

$$a^h b^k c^m d^n \cdot a \mapsto a^{1-h} b^{-k} c^m d^n, \;\; a^h b^k c^m d^n \cdot b \mapsto a^h b^{k+1} c^m d^n,$$

$$a^h b^k c^m d^n \cdot c \mapsto a^h b^k c^{m+1} d^n \text{ and } a^h b^k c^m d^n \cdot d \mapsto a^h b^k c^m d^{n+1},$$

which are all automatic.

In the representations G', G'', the multiplication from the right side is not automatic. To see this, $a^h (bcd)^k c^{m-k} d^{n-k} \cdot a \mapsto a^{1-h} (bcd)^{-k} c^{m+k} d^{n+k}$ in G' and $a^h (bc)^k c^{m-k} d^n \cdot a \mapsto a^{1-h} (bc)^{-k} c^{m+k} d^n$ in G''. Both these mappings are clearly not automatic. ∎

Theorem 23. *In the representation G all pattern languages are regular.*

Proof. Given a pattern π, let $S = \{x_1, x_2, \ldots, x_\ell\}$ be the set of variables which appear in π. For any subset S' of S, consider the pattern $\pi_{S'}$ obtained from π by using the substitution $sub(x) = ax$, for each variable x. Note that $Lang(\pi) \equiv \bigcup_{S' \subseteq S} Lang(\pi_{S'})$, where in pattern $\pi_{S'}$ only substitutions of the form $b^@ c^@ d^@$ are allowed (that is, the substitutions do not contain a); note that in π all substitutions are allowed.

Thus, it suffices to show that the language generated by each of $\pi_{S'}$ (in the representation G) is regular, where the substitutions allowed are only of the form $b^@ c^@ d^@$. So fix a pattern $\pi_{S'}$ as above. Consider any substitution of the form $sub(x_k) = b^{i_k} c^{j_k} d^{n_k}$, for x_k being a variable in $\pi_{S'}$. By shifting a to the left end, this substitution generates a string congruent to

$$\alpha b^{q_1 i_1 + q_2 i_2 + \ldots + q_\ell i_\ell} c^{m_1 j_1 + m_2 j_2 + \ldots + m_\ell j_\ell} d^{m_1 k_1 + m_2 k_2 + \ldots + m_\ell k_\ell},$$

where m_k is the number of times variable x_k appears in the pattern $\pi_{S'}$, and q_k is the difference of the number of times the occurrence of the variable x_k in $\pi_{S'}$ has even number of a's to the right in the pattern

$\pi_{S'}$, and the number of times the occurrence of the variable x_k has odd number of a's to the right in the pattern $\pi_{S'}$. The string α is obtained by concatenation of the constant parts of $\pi_{S'}$.

Thus, the language generated by the pattern $\pi_{S'}$ is equivalent to the set of strings given by the regular expression of the form:

$$\alpha \cdot b^{q_1 i_1 + q_2 i_2 + \ldots + q_\ell i_\ell} \cdot c^{m_k j_1 + m_2 j_2 + \ldots + m_\ell j_\ell} \cdot d^{m_1 k_1 + m_2 k_2 + \ldots + m_\ell k_\ell} = \alpha \cdot (b^{q'})^{@} \cdot (c^{m'})^{@} \cdot (d^{m'})^{@},$$

where q' is the greatest common divisor of $q_1, q_2, \ldots, q_\ell$ and m' is the greatest common divisor of $m_1, m_2, \ldots, m_\ell$. Note that m' is 0 iff $m_1, m_2, \ldots, m_\ell$ are all 0 and accordingly for q'. Now $\alpha = a^h b^{q''} c^{m''} d^{m'''}$, for some constants $h \in \{0,1\}$ and $q'', m'', m''' \in \mathbb{Z}$ and thus, the regular expression for the language generated by $\pi_{S'}$ can be written in the form

$$a^h \cdot (b^{q'})^{@} \cdot b^{q''} \cdot (c^{m'})^{@} \cdot c^{m''} \cdot (d^{m'})^{@} \cdot d^{m'''}$$

which is then a subset of G. As the language generated by π is a finite union of such expressions, the language generated by π (in the representation G) is regular. ∎

Theorem 24. *The language L generated by $\pi = x_1 x_2 x_1^{-1} x_2^{-1}$ has different levels in the Chomsky hierarchy in G, G' and G''.*

Proof. Consider any substitution $sub(x_1) = a^{h_1} b^{n_1} c^{i_1} d^{j_2}$ and $sub(x_2) = a^{h_2} b^{n_2} c^{i_2} d^{j_2}$, where $h_1, h_2 \in \{0,1\}$ and $n_1, n_2, i_1, i_2, j_1, j_2 \in \mathbb{Z}$. Then, the string generated by $x_1 x_2 x_1^{-1} x_2^{-1}$ using the above substitution is congruent to: $b^{-2n_1+2n_2}$, (if $h_1 = h_2 = 1$), b^{-2n_2}, (if $h_1 = 1, h_2 = 0$), b^{2n_1} (if $h_1 = 0, h_2 = 1$) and ε (if $h_1 = h_2 = 0$). Hence, $Lang(\pi) \equiv \{b^{2n} : n \in \mathbb{Z}\}$ in the representation G, $Lang(\pi) \equiv \{(bcd)^{2n} c^{-2n} d^{-2n} : n \in Z\}$ in G' and $Lang(\pi) \equiv \{(bc)^{2n} c^{-2n} : n \in Z\}$ in G''. Thus, the language corresponding to L is regular in G, context-free but not regular in G'' and not context-free in G'. ∎

Theorem 25. *All pattern languages are context-free in G''.*

Proof. As shown in Theorem 23, a given pattern language in G can be brought into a union of finitely many languages of the form

$$a^h \cdot (b^{q'})^{@} \cdot b^{q''} \cdot (c^{r'})^{@} \cdot c^{r''} \cdot (d^{r'})^{@} \cdot d^{r'''}$$

for suitable constants $h \in \{0,1\}$ and $q', q'', r', r'', r''' \in \mathbb{Z}$. Translated into G'', the above expression gives the set

$$\{a^h \cdot (bc)^{q'n+q''} \cdot c^{r'm - q'n + r'' - q''} \cdot d^{kr' + r'''} : m, n, k \in \mathbb{Z}\}$$

which is easily seen to be context-free. ∎

6 When all pattern languages are context-free

In this section it is shown that there is a group with a representation G such that every pattern language in G is context-free. Furthermore, this level is optimal: for every automatic representation G' of the same group, there is a verbal language which is not regular in G'. Throughout this section, the group H, and representations G and G' are fixed as in the definition below.

α	$(\alpha c^m d^n)^{-1}$	$\alpha c^m d^n$	$c^m d^n \alpha$	$(\alpha c^m d^n)^2$	$(\alpha c^m d^n)^4$
ε	$c^{-m}d^{-n}$	$c^m d^n$	$c^m d^n$	$c^{2m}d^{2n}$	$c^{4m}d^{4n}$
a	$ac^m d^{-n}$	$c^{-m}d^n a$	$ac^{-m}d^n$	d^{2n}	d^{4n}
b	$bc^{-n}d^{-m}$	$c^n d^m b$	$bc^n d^m$	$c^{m+n}d^{m+n}$	$c^{2m+2n}d^{2m+2n}$
ab	$bac^n d^{-m}$	$c^{-n}d^m ab$	$abc^n d^{-m}$	$ababc^{m+n}d^{-m+n}$	ε
ba	$abc^{-n}d^m$	$c^n d^{-m}ba$	$bac^{-n}d^m$	$babac^{m-n}d^{m+n}$	ε
aba	$abac^n d^m$	$c^{-n}d^{-m}aba$	$abac^{-n}d^{-m}$	$c^{m-n}d^{-m+n}$	$c^{2m-2n}d^{-2m+2n}$
bab	$babc^{-m}d^n$	$c^m d^{-n}bab$	$babc^m d^{-n}$	c^{2m}	c^{4m}
$abab$	$ababc^m d^n$	$c^{-m}d^{-n}abab$	$ababc^{-m}d^{-n}$	ε	ε

Figure 1. Table for $\alpha c^m d^m$, $c^m d^m \alpha$, $(\alpha c^m d^m)^2$ and $(\alpha c^m d^m)^4$

Definition 26. Consider a group H with the set of generators $\Sigma = \{a, b, c, d\}$. In the group H there are following additional rules:

(i) $ac \sim c^{-1}a$, $ad \sim da$, $bc \sim db$, $bd \sim cb$ and $cd \sim dc$,

(ii) $a^{-1} \sim a$ and $b^{-1} \sim b$,

(iii) For $\alpha, \beta \in \{a, b\}^@$, $\alpha \sim \beta$ iff $\forall x, y \in c^@ d^@ [\alpha x \sim y\alpha \Leftrightarrow \beta x \sim y\beta]$.

Note that $a^{-1} = a$ and $b^{-1} = b$ is consistent with rule (iii) above, as rule (iii) gives $aa \sim \varepsilon$ and $bb \sim \varepsilon$. Furthermore, it follows from the rules above that

- $baba \sim abab$, $ababa \sim bab$ and $babab \sim aba$.
- $a^{-1} = a, b^{-1} = b, (ab)^{-1} = ba, (ba)^{-1} = ab, (aba)^{-1} = aba, (bab)^{-1} = bab, (abab)^{-1} = abab.$

Congruence among other strings over $\{a, b\}^@$ can be derived similarly. It is then easy to verify that the above definition gives a group.

Furthermore, $G = F \cdot c^@ \cdot d^@$, where $F = \{\varepsilon, a, b, ab, ba, aba, bab, abab\}$ is a set of representatives of H. Note that each member of F realises a mapping when moved over an $x \in c^@ d^@$ and the members of F are multiplied according to the concatenations of these mappings. Note that H is an automatic group using the representation G.

Figure 1 gives, for various values of $\alpha \in F$ in the first column, inverses of $\alpha c^m d^n$ in the second column, congruent value of $\alpha c^m d^n$ (using the set of representatives $c^@ d^@ F$) in the third column, congruent value of $c^m d^n \alpha$ as a member of G in the fourth column, value of $(\alpha c^m d^n)^2$ in the fifth column and $(\alpha c^m d^n)^4$ in the sixth column. The last two columns are useful in verifying some of the arguments below.

Theorem 27. *Every pattern language L in G is context-free.*

Proof. Let π be a pattern in which the variables $x_1, x_2, \ldots, x_\ell$ occur and no other variables occur. A technique similar to that used in the proof of Theorem 23 is used. For each possible $\alpha = (\alpha_1, \ldots, \alpha_\ell) \in F^\ell$, consider the pattern π_α which is obtained by replacing each occurrence of x_i in π by $\alpha_i x_i$, and then considering the language generated by the pattern π_α, where the substitutions allowed are only of the form $c^@d^@$. Thus, the language corresponding to π_α is of the form

$$\{\beta \cdot c^{s_0+o_1n_1+p_1m_1+\ldots+o_\ell n_\ell+p_\ell m_\ell} \cdot d^{t_0+q_1n_1+r_1m_1+\ldots+q_\ell n_\ell+r_\ell m_\ell} : n_1, n_2, \ldots, n_\ell, m_1, m_2, \ldots, m_\ell \in \mathbb{Z}\}$$

where $\beta \in F$, and $\beta c^{s_0}d^{t_0}$ is obtained by taking the concatenation of the constants appearing in π_α and then taking a congruent member of $Fc^@d^@$. The integers o_i, p_i, q_i, r_i are obtained based on how the shifting of a's and b's appearing in π_α to the left end changes the substitution $x_i = c^{n_i}d^{m_i}$ based on the rules of multiplication in the group. Now it follows from Lemma 28 that each of L_α is context-free, and thus the pattern language L is context-free. ∎

Lemma 28. *For all* $s_0, s_1, \ldots, s_\ell, t_1, \ldots, t_\ell \in \mathbb{Z}$, $\{c^{s_0+s_1n_1+\ldots+s_\ell n_\ell}d^{t_0+t_1n_1+\ldots+t_\ell n_\ell} : n_1, n_2, \ldots, n_\ell \in \mathbb{Z}\}$ *is context-free.*

Proof. By considering the equivalent language

$$\{c^{s_0+s_1n_1-s_1n_1'+\ldots+s_\ell n_\ell-s_\ell n_\ell'}d^{t_0+t_1n_1-t_1n_1'+\ldots+t_\ell n_\ell-t_\ell n_{\ell'}} : n_1, n_2, \ldots, n_\ell, n_1', n_2', \ldots, n_\ell' \geq 0\},$$

it suffices to show that

$$\{c^{s_0+s_1n_1+\ldots+s_\ell n_\ell}d^{t_0+t_1n_1+\ldots+t_\ell n_\ell} : n_1, n_2, \ldots, n_\ell \geq 0\}$$

is context-free for all $s_0, s_1, \ldots, s_\ell, t_1, \ldots, t_\ell \in \mathbb{Z}$. Furthermore, it suffices to show that for each $h, h' \in \{1, -1\}$,

$$\{c^{s_0+s_1n_1+\ldots+s_\ell n_\ell}d^{t_0+t_1n_1+\ldots+t_\ell n_\ell} : n_1, n_2, \ldots, n_\ell \geq 0 \text{ and } h \cdot (s_0 + s_1n_1 + \ldots + s_\ell n_\ell) \geq 0 \text{ and } h' \cdot (t_0 + t_1n_1 + \ldots + t_\ell n_\ell) \geq 0\}$$

is context-free. It is now shown that

$$A = \{c^{s_0+s_1n_1+\ldots+s_\ell n_\ell}d^{t_0+t_1n_1+\ldots+t_\ell n_\ell} : n_1, n_2, \ldots, n_\ell \geq 0 \text{ and } (s_0 + s_1n_1 + \ldots + s_\ell n_\ell) \geq 0 \text{ and } (t_0 + t_1n_1 + \ldots + t_\ell n_\ell) \geq 0\}$$

is context-free. The case for $(h, h') \in \{(1, -1), (-1, 1), (-1, -1)\}$ is similar.

For any pair i, j such that $s_i > 0$ and $s_j < 0$, let $t'_{i,j} = t_i * (-s_j) + t_j * s_i$. Let $S = \{(i, j) : s_i > 0, s_j < 0\}$. Let $m = 1 + \sum_{0 \leq i \leq \ell} |s_i|$.

Let $A' = \{c^{s_0+s_1n_1+\ldots+s_\ell n_\ell}d^{t_0+t_1n_1+\ldots+t_\ell n_\ell+\sum_{(i,j)\in S} t'_{i,j}n'_{i,j}} : n_1, n_2, \ldots, n_\ell \geq 0$ and $(n'_{i,j} \geq 0$ for $(i, j) \in S)$ and $(n_j \leq m$ for $s_j < 0)$ and $(s_0+s_1n_1+\ldots+s_\ell n_\ell) \geq 0$ and $(t_0+t_1n_1+\ldots+t_\ell n_\ell+\sum_{(i,j)\in S} n'_{i,j}t'_{i,j}) \geq 0\}$.

Note that $A' \subseteq A$ as usage of $n'_{i,j}$ in A' can be equivalently achieved by increasing n_i by $-s_j * n'_{i,j}$ and n_j by $s_i * n'_{i,j}$. For $s_j < 0$, note that if $n_j > m$ in A for generation of some string $x \in A$, then there exists an i such that $s_i > 0$ and $n_i > -s_j$ (as otherwise, $(s_0 + s_1n_1 + \ldots + s_\ell n_\ell) < 0$). Thus, for generating x in A', n_j and n_i could be decreased respectively by s_i and $-s_j$, and correspondingly $n'_{i,j}$ increased by 1. Repeatedly using the above mechanism makes sure that n_j becomes smaller than m.

Thus, by considering each possible values of n_j, such that $s_j < 0$ separately it can be assumed without loss of generality that $s_1, s_2, \ldots, s_\ell \geq 0$ in the definition of A above (though s_0 maybe negative). It is then easy to construct a non-deterministic pushdown automata for A and thus A is context-free. ∎

It is now shown that for any representation G' of H, there exist non regular pattern languages.

Theorem 29. *Suppose G' is a regular set of representatives for the group H. Then there is a verbal language $L \subseteq G$ such that the corresponding language $L' = \{w \in G' : \exists v \in L\,[v \sim w]\}$ is not regular.*

Proof. In the first part of the proof a family of verbal languages $L_{i,j}$ is constructed. In the second part of the proof it is shown that for any given regular representation G' of H, at least one of these verbal languages is not regular in representation G'. The first part of the proof is done over G in order to keep the notation simple.

First part. For the proof below, it will be useful to refer to Figure 1 for values of y^2, y^4 for various $y \in G$ as well as for $\alpha c^m d^n$, for various values of $\alpha \in F$.

Note that, for all $y \in G$, y^4 is congruent to $c^{m'} d^{n'}$ for some $m', n' \in \mathbb{Z}$, where either m', n' are multiples of 4 or both are even numbers satisfying $m' = n'$ or $m' = -n'$.

Now consider the following patterns over the variables x_1, x_2, x_3.

- $\rho_1 = x_1^4 x_2^2 x_1^4 x_2^{-2}$,
- $\rho_2 = \rho_1 x_3^2 \rho_1 x_3^{-2}$,
- $\rho_3 = \rho_2 x_2 x_3 x_2 x_3 \rho_2^{-1} x_3^{-1} x_2^{-1} x_3^{-1} x_2^{-1}$,
- $\rho_4 = x_2 \rho_3 x_2^{-1} \rho_3$ and
- $\rho_5 = x_3 \rho_4 x_3^{-1} \rho_4$.

Claim 30. *(a) The language generated by ρ_i (in the representation G) is contained in $c^{@} d^{@}$.*

(b) For i, j with $1 \leq i < j \leq 5$, for any substitution sub for the variables x_1, x_2, x_3, if $sub(\rho_i) \sim \varepsilon$, then $sub(\rho_j) \sim \varepsilon$.

As $\alpha^4 \sim \varepsilon$, for all $\alpha \in F$, part (a) of the claim is easy to verify. Part (b) now follows directly from definition of ρ_i, ρ_j.

Claim 31. *The language generated by ρ_5 in the representation G is $\{c^{32k} d^{32k}, c^{32k} d^{-32k}, c^{64k}, d^{64k} : k \in \mathbb{Z}\}$.*

To see the claim, consider any substitution sub such that $sub(\rho_5) \neq \varepsilon$. Thus, $sub(\rho_i) \neq \varepsilon$, for $i = 1, 2, 3, 4, 5$. Suppose $sub(x_1^4) \sim c^{m'} d^{n'}$, $sub(x_2) \in \beta c^{@} d^{@}$ and $sub(x_3) \in \gamma c^{@} d^{@}$, where $\beta, \gamma \in F$.

Now, $\beta \notin \{ab, ba\}$, as otherwise, $sub(x_2^2) \sim ababc^i d^j$, for some $i, j \in \mathbb{Z}$ and thus $sub(\rho_1) \sim x_1^4 ababc^i d^j x_1^4 d^{-j} c^{-i} baba \sim x_1^4 abab x_1^4 baba \sim x_1^4 x_1^{-4} ababbaba \sim \varepsilon$. Similarly $\gamma \notin \{ab, ba\}$ as otherwise $sub(\rho_2) \sim \varepsilon$. Thus, it is easy to verify that $sub(\rho_1) \sim c^{2m'} d^{2n'}$ and $sub(\rho_2) \sim c^{4m'} d^{4n'}$.

Furthermore, $\beta\gamma\beta\gamma \not\sim \varepsilon$ as otherwise $sub(x_2 x_3 x_2 x_3) = c^i d^j$, for some $i, j \in \mathbb{Z}$ and thus $sub(\rho_3) \sim sub(\rho_2) c^i d^j sub(\rho_2^{-1}) d^{-j} c^{-i} \sim \varepsilon$. Thus, $\beta\gamma \in \{ab, ba\}$, as all other members of F are self-inverses. Therefore, $\beta\gamma\beta\gamma = abab$ and for some $i, j \in \mathbb{Z}$, $sub(\rho_3) \sim c^{4m'} d^{4n'} ababc^i d^j d^{-4n'} c^{-4m'} d^{-j} c^{-i} baba \sim c^{4m'} d^{4n'} c^{-i} d^{-j} d^{4n'} c^{4m'} d^j c^i ababbaba \sim c^{8m'} d^{8n'}$. Furthermore, neither β nor γ can be $abab$ or ε, as otherwise $\beta\gamma\beta\gamma \sim \varepsilon$. As the fourth power of every member of F is congruent to ε, $\beta \neq \gamma$. As $\beta\gamma \not\sim abab$ and $abab \sim baba$, it cannot be the case that $\beta = a \wedge \gamma = bab$ or $\beta = b \wedge \gamma = aba$ or vice versa. It follows that exactly one of β, γ has an odd number of a and exactly one has an odd number of b; note that it cannot happen that one of them has both, an odd number of a and an odd number of b as

β	γ	ρ_4	ρ_5
a	b	$d^{16n'}$	$c^{16n'}d^{16n'}$
a	aba	$d^{16n'}$	$c^{-16n'}d^{16n'}$
bab	b	$c^{16m'}$	$c^{16m'}d^{16m'}$
bab	aba	$c^{16m'}$	$c^{16m'}d^{-16m'}$
b	a	$c^{8m'+8n'}d^{8m'+8n'}$	$d^{16m'+16n'}$
b	bab	$c^{8m'+8n'}d^{8m'+8n'}$	$c^{16m'+16n'}$
aba	a	$c^{8m'-8n'}d^{-8m'+8n'}$	$d^{-16m'+16n'}$
aba	bab	$c^{8m'-8n'}d^{-8m'+8n'}$	$c^{16m'-16n'}$

Figure 2. Table for ρ_4 and ρ_5.

then it would be ab or ba. Thus, one of β, γ is a or bab and the other one is b or aba. Figure 2 gives the values of $sub(\rho_4)$ and $sub(\rho_5)$ for these remaining possibilities for β and γ.

Recall that m', n' can be chosen freely either as multiples of 4 or both as even numbers satisfying $m' = n'$ or $m' = -n'$. Thus, ρ_5 generates $\{c^{32k}d^{32k}, c^{32k}d^{-32k}, c^{64k}, d^{64k} : k \in \mathbb{Z}\}$. This completes the proof of Claim 31.

Now, for each $i, j \geq 0$, consider the pattern $\pi_{i,j} = (\rho_5)^i x_4 (\rho_5)^j x_4^{-1}$.

Claim 32. *The language $L_{i,j}$ generated by pattern $\pi_{i,j}$ (in the representation G) is* $(c^{32(i+j)} \cdot d^{32(i+j)})^@ \cup (c^{32(i+j)} \cdot d^{32(-i+j)})^@ \cup (c^{32(i+j)} \cdot d^{32(i-j)})^@ \cup (c^{32(i+j)} \cdot d^{32(-i-j)})^@ \cup (c^{32(i-j)} \cdot d^{32(i+j)})^@ \cup (c^{32(i-j)} \cdot d^{32(-i+j)})^@ \cup (c^{32(i-j)} \cdot d^{32(i-j)})^@ \cup (c^{32(i-j)} \cdot d^{32(-i-j)})^@ \cup (c^{64(i+j)})^@ \cup (c^{64(i-j)})^@ \cup (d^{64(i+j)})^@ \cup (d^{64(i-j)})^@ \cup (c^{64i} \cdot d^{64j})^@ \cup (c^{64i} \cdot d^{-64j})^@ \cup (c^{64j} \cdot d^{64i})^@ \cup (c^{-64j} \cdot d^{64i})^@$.

To see the claim, note that, for any substitution sub, with $sub(x_4) \in \delta c^@ d^@$, with $\delta \in F$, $sub(\pi_{i,j}) = sub(\rho_5)^i \delta sub(\rho_5)^j \delta^{-1}$. Furthermore, $\{\delta c^k \delta^{-1} : \delta \in F\} = \{\delta d^k \delta^{-1} : \delta \in F\} = \{c^k, d^k, c^{-k}, d^{-k}\}$ and $\{\delta c^k d^k \delta^{-1} : \delta \in F\} = \{\delta c^k d^{-k} \delta^{-1} : \delta \in F\} = \{c^k d^k, c^k d^{-k}, c^{-k} d^k, c^{-k} d^{-k}\}$. Claim 32 now follows immediately from Claim 31. The following claim follows from Claim 32.

Claim 33. *For any $n \neq 0$, there are only finitely many members in $L_{i,j}$ of the form $c^m d^n$.*

Second part. Let a regular G' be given in which every group element of H has a unique representative. It will be shown that, for some values of i and j, $L_{i,j}$ is non-regular (in the representation G').

Given a word $w \sim c^m d^n \alpha$ with $\alpha \in F$, let $f_c(w) = m$ and $f_d(w) = n$ and $f_F(w) = \alpha$. Furthermore, without loss of generality, assume that if the DFA recognising G' is in the same state after reading either v or w, then $f_F(v) = f_F(w)$. This can be easily ensured by padding $f_F(x)$ to the state of the automaton after reading the string x. Let the so obtained DFA be M.

Claim 34. *For all $x, u \in \Sigma^@$, if M is in the same state after reading either u or ux, then there is a word $x' \in c^@ d^@$ such that $ux^k u' \sim (x')^k uu'$ for all $u' \in \Sigma^@$ and $k \geq 0$. Furthermore, $f_c(ux)$ and $f_d(ux)$ differ from $f_c(u)$ and $f_d(u)$, respectively, by at most $|x|$.*

To see the claim, note that M being in the same state after reading either u or ux implies that $f_F(u) = f_F(ux)$. And thus, $u \sim c^i d^j \alpha$ and $ux \sim c^{i'} d^{j'} \alpha$, for some $\alpha \in F$ and $i, j, i', j' \in \mathbb{Z}$. Thus, for

$x' = c^{i'-i}d^{j'-j}$, $ux^ku' \sim (x')^k uu'$. Furthermore, as $\alpha x = x'\alpha$, it follows that $i' - i = f_c(x')$ and $|f_c(x')| \leq |x|$ and $j' - j = f_d(x')$ and $|f_d(x')| \leq |x|$ (by the definition of the group operation). Thus, $|f_c(ux) - f_c(u)| = |i' - i| \leq |x|$ and $|f_d(ux) - f_d(u)| = |j' - j| \leq |x|$. Claim follows.

Let p be the number of states of M and let $q = (128p^3)!$. Let L' be $L_{q/2,q/2-1} \cup L_{q/2+1,q/2}$ as represented in G'. It will be shown that L' is not regular by showing that any DFA for accepting L' accepts infinitely many elements congruent to elements of the form $c^m d^{q\ell}$, for some fixed ℓ (depending on the DFA accepting L'), which then contradicts Claim 33.

Consider the following two cases for any word $w \in G'$ with $w \sim c^{q^2\ell}d^{q\ell}$.

Case (a). The word w has a proper splitting $w_0w_1w_2 \ldots w_pw_{p+1}$ such that $|w_1w_2 \ldots w_p| \leq 4p^3$ and $f_d(w_0w_1 \ldots w_k)$ is the same for $k = 0, 1, \ldots, p$. Here proper splitting means that each component is different from ε.

In this case there are i, j with $0 \leq i < j \leq p$ such that M is in the same state after reading the inputs $w_0w_1 \ldots w_i$ and $w_0w_1 \ldots w_j$. Thus, $w_0 \ldots w_i(w_{i+1} \ldots w_j)^+w_{j+1} \ldots w_{p+1} \subseteq G'$ and f_d maps all these words to the same value. In the following let $u = w_0 \ldots w_i$, $x = w_{i+1} \ldots w_j$, $u' = w_{j+1} \ldots w_{p+1}$. Let $r = |f_c(u) - f_c(ux)|$. Note that, by Claim 34, $r \leq |x| \leq 4p^3$. Furthermore, $f_F(u) = f_F(ux)$ as M is in the same state after reading u and ux. It follows that $r > 0$, as otherwise uxu' and $uxxu'$ would give two representations of the same group element in G'. Let $v = ux \cdot x^{q\cdot\ell/r}u'$. Note that either $v \sim c^{(q+1)q\ell}d^{q\ell}$ or $v \sim c^{(q-1)q\ell}d^{q\ell}$. Thus, $v \in G' \cap L'$.

Case (b). The word w has a proper splitting $w_0w_1 \ldots w_{2p^2}w_{2p^2+1}$ such that $|w_1w_2 \ldots w_{2p^2}| \leq 4p^3$ and $f_d(w_0w_1 \ldots w_k)$ takes its maximum or minimum value (among $f_d(w_0w_1 \ldots w_{k'})$, for $k' \leq 2p^2$) at $k = p^2$ and $f_d(w_0w_1 \ldots w_k)$ differs from this extreme value by $|p^2 - k|$ for $k = 0, 1, \ldots, 2p^2$.

Now, by pigeonhole principle, there exist i, j with $0 \leq p^2 - i < p^2 - j \leq p^2 + j < p^2 + i \leq 2p^2$ such that M is in the same state after reading $w_0 \ldots w_{p^2-i}$ or $w_0 \ldots w_{p^2-j}$ as well as M is in the same state after reading $w_0 \ldots w_{p^2+i}$ or $w_0 \ldots w_{p^2+j}$. Let $u = w_0 \ldots w_{p^2-i}$, $x = w_{p^2-i+1} \ldots w_{p^2-j}$, $u' = w_{p^2-j+1} \ldots w_{p^2+j}$, and $y = w_{p^2+j+1} \ldots w_{p^2+i}$, $u'' = w_{p^2+i+1} \ldots w_{2p^2+1}$. Let $r = |f_c(ux) - f_c(u) + f_c(uxu'y) - f_c(uxu')|$. Note that $r > 0$, as otherwise there are two different representations $uxu'yu''$ and $uu'u''$ for the same group element in G'.

Now, the following hold for all $k > 0$:

- $f_d(ux^ku'y^ku'') = f_d(uxu'yu'')$;
- $f_c(ux^ku'y^ku'') = f_c(uxu'yu'') + (k-1)r$ or $f_c(ux^ku'y^ku'') = f_c(uxu'yu'') - (k-1)r$;
- $f_F(ux^ku'y^ku'') = \varepsilon$.

Note that here $u' = \varepsilon$ if $j = 0$. It follows that $v = uxx^{q\ell/r}u'yy^{q\ell/r}u''$ satisfies either $v \sim c^{(q+1)q\ell}d^{q\ell}$ or $v \sim c^{(q-1)q\ell}d^{q\ell}$. Thus, $v \in L'$ as represented by G'.

For each ℓ, either Case (a) or Case (b) holds. Assume by way of contradiction that both cases do not hold. Call a prefix u of w extremal, if $|f_d(u)| \geq p^2$ and $|f_d(u)| > |f_d(u')|$ for any prefix u' of u. Now, as Case (a) and (b) do not hold, for any extremal prefix u of w with $|u| \leq |w| - 2p^3$, there exists another extremal prefix u' of w such that $|u| < |u'| \leq |u| + 2p^3$; otherwise, $|f_d(u)| \geq |f_d(u'')|$, for each u'' of length at most $|u| + 2p^3$, and thus by Case (a) not holding, there exists a splitting $w_0w_1 \ldots w_{2p^2+1}$ of w such that, for $i \leq 2p^2$, $w_0 \ldots w_{p^2} = u$, $|f_d(w_0w_1 \ldots w_i)| = |f_d(u)| - |i - p^2|$, $|w_0| \geq |u| - 2p^3$ and $|w_0w_1 \ldots w_{2p^2}| \leq |u| + 2p^3$ and thus Case (b) holds. Note that by Case (a) not holding, there exists an extremal prefix u of w of length at most $4p^3$. Thus, $|f_d(w)| \geq -4p^3 + \frac{|w|-6p^3}{4p^3}$, which contradicts $f_d(w) = q\ell$, as $|w| \geq q^2\ell$.

Now suppose by way of contradiction that in the representation G' the language $L' = L_{q/2,q/2-1} \cup L_{q/2+1,q/2}$ is regular. Let s be the number of states of a DFA M' recognising L' in the representation G'.

Let $\ell > s!$. Let v as defined above in the case analysis be $uxx^{q\ell/r}u'$ in case (a) and $uxx^{q\ell/r}u'yy^{q\ell/r}u''$ in case (b). In both cases $v \in L'$. Furthermore, M' is in the same state after reading any of (i) $uxx^{q\ell/r+ts!}$, for different values of $t = 0, 1, 2, \ldots$ (in case (a)), and (ii) $uxx^{q\ell/r+ts!}u'yy^{q\ell/r+ts!}$, for different values of $t = 0, 1, 2, \ldots$ (in case (b)).

Thus, for every $t > 0$ the words $\beta_t = uxx^{q\ell/r+ts!}u'$ in case (a) and the words $\beta_t = uxx^{q\ell/r+ts!}u'y$ $y^{q\ell/r+ts!}u''$ in case (b) are accepted by the M' and satisfy $f_d(\beta_t) = f_d(v)$. There are infinitely many different such β_t, which all represent different members of G'. However, there are only finitely many elements of L' which have the same nonzero value $q\ell$ of f_d (see Claim 33). This contradiction gives that L' cannot be regular. Hence one of the verbal languages $L_{q/2,q/2-1}$ and $L_{q/2+1,q/2}$, as represented in G' cannot be regular. ∎

Remark 35. Arguing as in Theorem 4, it can be shown that Theorem 29 holds even if generators other than a, b, c, d are used. Hence there is no automatic presentation of G where all verbal sets are regular.

7 Conclusions

The starting point for the study was to look at verbal languages in free groups with finitely many and at least two generators. For any given automatic representation of the free group, these languages are not context-free unless they are a singlton consisting of the neutral element of the group, or they are the full group. Similarly, it was shown that for the free groups, exactly the full group and singleton pattern languages are regular and no other pattern languages are regular. These two results are based on the theorem which showed that in free groups, with at least two generators, it does not depend on the automatic presentation chosen whether a language is regular, context-free, context-sensitive or recursively enumerable. The remainder of the paper was dedicated to explore the extent to which these results can be carried over to other groups similar to the free groups or, more generally, to all automatic groups. The answers are in general negative, although in a few cases generalisation was obtained. For free Abelian groups with two or more generators, the above mentioned independence result (for languages being regular, context-free, context-sensitive or recursively enumerable inpdenent of the representation chosen) breaks down. However still, for Abelian automatic groups, all verbal languages and all pattern languages are regular independently of the chosen representation of the group. The result that only trivial pattern languages (the whole group or singletons) are regular and only trivial verbal languages are context-free can be carried over to some, but not all, hyperbolic groups. Furthermore, for automatic groups in general, examples were provided where the level of a pattern language or verbal language in the Chomsky hierarchy depends on the automatic presentation, and thus the nice characterisation obtained for the free groups does not hold in general. The following two questions remain open.

Question 36. *Is every pattern language in an automatic group an indexed language?*

Question 37. *Is there a non-trivial context-free verbal language in some non-elementary hyperbolic group?*

Note that the corresponding version of Question 37 for pattern languages and regularity is answered positively by Theorem 19.

Acknowledgments. A preliminary version of this paper was presented at the conference LICS 2012. The authors would like to thank the anonymous referees for the careful proofreading and useful comments and for providing Proposition 1 which simplified some proofs. The authors would also like to thank Volker Diekert and Murray Elder for correspondence.

References

[1] Alfred Aho. Indexed grammars — an extension of context-free grammars. *Journal of the Association for Computing Machinery*, 15(4):647–671, 1968.

[2] Dana Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.

[3] Pater R.J. Asveld. *A characterization of ET0L and EDT0L.* Technical Report, Department of Applied Mathematics, Twente University of Technology, 1976. http://doc.utwente.nl/70588/1/1976TWmemo129.pdf

[4] Jean Berstel. *Transductions and Context-Free Languages*. Teubner-Verlag, 1979. http://www-igm.univ-mlv.fr/~berstel/LivreTransductions/LivreTransductions14dec2009.pdf

[5] Achim Blumensath. *Automatic structures.* Diploma thesis, Department of Computer Science, RWTH Aachen, 1999.

[6] Achim Blumensath and Erich Grädel. Automatic structures. *Fifteenth Annual IEEE Symposium on Logic in Computer Science*, LICS 2000, pages 51–62, 2000.

[7] Laura Ciobanu, Volker Diekert and Murray Elder. Solution sets for equations over free groups are EDT0L languages. *Automata, Languages and Programming, Forty-second International Colloquium, ICALP 2015*, Lecture Notes in Computer Science, Volume 9134, pages 134–145, Springer 2015.

[8] Laura Ciobanu, Murray Elder and Michal Ferov. Applications of L systems to group theory. *International Journal of Algebra and Computation*, 28(2):309–330, 2018.

[9] Andrzej Ehrenfeucht and Grzegorz Rozenberg. Some context-free languages that are not deterministic ET0L languages. *R.A.I.R.O. Informatique Théorique*, 11(4):273–291, 1977.

[10] Andrzej Ehrenfeucht, Grzegorz Rozenberg and Sven Skyum. A relationship between ET0L and EDT0L languages. *Theoretical Computer Science* 1:325–330, 1976.

[11] David B.A. Epstein, James W. Cannon, Derek F. Holt, Silvio V.F. Levy, Micheal S. Paterson and William P. Thurston. *Word Processing in Groups.* Jones and Bartlett Publishers, Boston, 1992.

[12] Robert Gilman. Formal languages and infinite groups. *Geometric and Computational Perspectives on Infinite Groups* (Minneapolis, MN and New Brunswick, NJ, 1994), *DIMACS Series Discrete Mathematics and Theoretical Computer Science*, 25:27–51, 1996.

[13] Mikhail Gromov. Hyperbolic groups. *Essays in group theory*, pages 75–263, Springer, New York, 1987.

[14] Bernard R. Hodgson. *Théories décidables par automate fini.* Ph.D. thesis, Département de mathématiques et de statistique, Université de Montréal, 1976.

[15] Bernard R. Hodgson. Décidabilité par automate fini. *Annales des sciences mathématiques du Québec*, 7(1):39–57, 1983.

[16] Markus Holzer, Sebastian Jakobi and Ian McQuillan. Generalized derivations with synchronised context-free grammars. *Developments in Language Theory* – Sixteenth International Conference, DLT 2016, Taipei, Taiwan, August 14–17, 2012. Proceedings. *Springer LNCS* 7410:109–120, 2012.

[17] Sanjay Jain, Yuh Shin Ong and Frank Stephan. Regular patterns, regular languages and context-free languages. *Information Processing Letters* 110:1114–1119, 2010.

[18] Olga Kharlampovich, Bakhadyr Khoussainov and Alexei Miasnikov. From automatic structures to automatic groups. *Groups, Geometry and Dynamics*, 8(1):157–198, 2014.

[19] Olga Kharlampovich and Alexei Myasnikov. Elementary theory of free non-Abelian groups. *Journal of Algebra*, 302(2):451–552, 2006.

[20] Olga Kharlampovich and Alexei Myasnikov. Definable subsets in a hyperbolic group. *International Journal of Algebra and Computation*, 23(1):91–110, 2013.

[21] Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. *Logical and Computational Complexity*, (International Workshop LCC 1994). Springer LNCS 960:367–392, 1995.

[22] Gennady S. Makanin. The problem of solvability of equations in a free semigroup. *Sbornik: Mathematics*, 32(2):129–198, 1977.

[23] Gennady S. Makanin. Equations in a free group. *Izvestiya Akademii Nauk SSSR Seriya Matematicheskaya* 46(6):1199–1273, 1982. Translated to English, 21(3):483–546, 1983.

[24] Gennady S. Makanin. Decidability of the universal and positive theories of a free group. *Izvestiya Akademii Nauk SSSR Seriya Matematicheskaya*, 48(4):735–749, 1984. Translated to English, 25(1):75–88, 1985.

[25] Yuri I. Merzlyakov. Positive formulae over free groups. *Algebra i Logika*, 5:25–42, 1966.

[26] Alexei Myasnikov and Vitaly Romankov. On rationality of verbal subsets in a group. *Theory of Computing Systems*, 52(4):587–598, 2013.

[27] Nikolay Nikolov. *Algebraic properties of profinite groups.* Technical report, 2012. `http://arxiv.org/abs/1108.5130`

[28] Daniel Reidenbach. A non-learnable class of E-pattern languages. *Theoretical Computer Science*, 350:91–102, 2006.

[29] Daniel Reidenbach. The ambiguity of morphisms in free monoids and its impacts on algorithmic properties of pattern languages. PhD Thesis, University of Kaiserslautern, Germany, 2006.

[30] Grzegorz Rozenberg. Extensions of tabled 0L-systems and languages. *International Journal of Computer and Information Sciences* 2(4):311–336, 1973.