The Complexity of Verbal Languages over Groups

Sanjay Jain
Department of Computer Science
National University of Singapore
Singapore 117417
Email: sanjay@comp.nus.edu.sg

Alexei Miasnikov
Department of Mathematical Sciences
Steven's Institute of Technology
Hoboken, New Jersey, USA
Email: amiasnik@stevens.edu

Frank Stephan
Department of Mathematics and
Department of Computer Science
National University of Singapore
Singapore 117543
Email: fstephan@comp.nus.edu.sg

***Abstract*—This paper investigates the complexity of verbal languages and pattern languages of Thurston automatic groups in terms of the Chomsky hierarchy. Here the language generated by a pattern is taken as the set of representatives of all strings obtained when chosing values for the various variables. For noncommutative free groups, it is shown that the complexity of the verbal and pattern languages (in terms of level on the Chomsky hierarchy) does not depend on the Thurston automatic representation and that verbal languages cannot be context-free (unless they are either the empty word or the full group). They can however be indexed languages. Furthermore, it is shown that in the general case, it might depend on the exactly chosen Thurston automatic representation which level a verbal language takes in the Chomsky hierarchy. There are examples of groups where, in an appropriate representation, all pattern languages are regular or context-free, respectively.**

***Index Terms*—Thurston Automatic Groups; Free Groups; Verbal Languages; Chomsky Hierarchy;**

I. 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 [12] 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 [13] asked whether there is a pattern language which is context-free but not regular; Jain, Ong and Stephan [5] showed that this depends on the alphabet size — if the alphabet size is 2 or 3 then there are such languages while for alphabet size of at least 4 such languages do not exist.

Pattern languages, in particular the verbal languages which are generated by patterns without constants, also play an important role in group theory [6], [8], [9], [10], [11]. Gilman [4] studied the low 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 [7]; it follows from this result that verbal sets are always decidable, 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 Thurston automatic groups. Thurston 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 novel language classes during many decades of investigations in theoretical computer science. The present work wants to answer fundamental questions on pattern languages and verbal languages for automatic groups; it also brings the result of Miasnikov and Romankov [10] 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 terminology and notation is explained more in detail. Let $\mathbf{Z}$ denote the set of integers. Given a set Σ of generators and Σ^{-1} of the inverses of the generators, one says that a group is Thurston automatic iff there is a regular subset G of words over $\Sigma \cup \Sigma^{-1}$ containing exactly one representative for each group element such that the mappings realising the group operation $x \mapsto ax$ for each fixed $a \in G$ is an automatic function. A function $f : G \to G$ is automatic iff there is an 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 provided to the automaton in those cycles where one of x, y is already read completely and the other one not. The members of G are called the set of representatives of the group; formally, representatives are defined as follows: one says $v \sim w$ if v and w represent the same group word in the set of all words over $\Sigma \cup \Sigma^{-1}$; then for each v there is a unique word $w = repr_G(v)$ such that $w \in G \wedge w \sim v$. The empty string ε represents the neutral element of the group.

For the languages of groups, a pattern π is a string over variables $x_1, \ldots, x_n$, their inverses $x_1^{-1}, \ldots, x_n^{-1}$, the generators and their inverses. The pattern $\pi = \alpha_0 x_{i_1}^{j_1} \alpha_1 x_{i_2}^{j_2} \alpha_2 \ldots x_{i_\ell}^{j_\ell} \alpha_\ell$ over variables $x_1, \ldots, x_n$ with all $i_k \in \{1, \ldots, n\}$ and all $j_k \in \{-1, +1\}$ and $\alpha_0, \alpha_1, \ldots, \alpha_\ell \in G$ generates the language

$$L = \{repr_G(\alpha_0 y_{i_1}^{j_1} \alpha_1 \ldots y_{i_\ell}^{j_\ell} \alpha_\ell) : y_1, \ldots, y_n \in G\}.$$

Furthermore, the pattern π is called constant-free iff $\alpha_0 = \varepsilon \wedge \ldots \wedge \alpha_\ell = \varepsilon$ and a language is called verbal iff it is generated by a constant-free pattern.

Now some example of patterns: $\pi_1 = x_1 a x_1^{-1} bb$, $\pi_2 = x_1 x_2 x_1^{-1} x_2^{-1}$ and $\pi_3 = x_1 x_1 x_2 x_2$. Here π_1 contains constants while π_2 and π_3 do not. Now $bbbab^{-1}$ is generated by the pattern π_1 when assigning bbb to x_1; two occurrences of b^{-1} in x_1^{-1} cancel out with the two constants bb at the end of the pattern. So π_2 and π_3 generate verbal languages but π_1 does not generate a verbal language (as ε is not in the language generated by π_1).

In the following, to simplify the notation for groups, given any finite subset A of the generators Σ of a group, A^* always denotes the strings over all members from $A \cup A^{-1}$. More generally, when talking about an arbitrary subset $A \subseteq G$, A^* is the set of all w which are formed as a word over $A \cup A^{-1}$; without loss of generality, such a w is reduced, that is, it does not contain substrings of the form vv^{-1}.

Example 1: Let a group have the generators $\Sigma = \{a, b, c, d\}$ with $a = a^{-1}$, $ab = b^{-1}a$, $ab^{-1} = ba$ $ac = ca$, $ad = da$, $bc = cb$, $bd = db$, $cd = dc$ and consider $\pi_0 = x_1 x_2 x_1^{-1} x_2^{-1}$. If one represents the group by $G = \{\varepsilon, a\} \cdot b^* \cdot c^* \cdot d^*$ then the language generated by π_0 is the regular language $\{bb\}^*$. If one choses the representation $G' = \{\varepsilon, a\} \cdot (bcd)^* \cdot c^* \cdot d^*$ then the resulting language is in G' of the form $\{(bcd)^{2n} c^{-2n} d^{-2n} : n \in \mathbf{Z}\}$ and therefore not context-free.

A proof for the claims in the above example will be given below in Section V. For free groups, this problem does not arise as the level of the Chomsky hierarchy of a language is independent of the group representation; this is shown explicitly in Section II. Therefore one can fix the representatives as the reduced words in Sections III and IV; here a word w is reduced iff it does not contain any empty subword of the form vv^{-1} for $v \in \Sigma \cup \Sigma^{-1}$. Note that, if one chooses $\Sigma = \{a\}$ then all pattern languages are of the form

$$L_{i,j} = \{w \in G : \exists n \in \mathbf{Z}\, [\mathbf{w} \sim \mathbf{a^{i+j \cdot n}}]\}.$$

To see this, note that the group is commutative and therefore one can write the pattern in the form $a^i x_1^{j_1} x_2^{j_2} \ldots x_k^{j_k}$ with $j_1, \ldots, j_k > 0$ and variables whose powers add up to 0 being omitted; then a further normalisation permits to choose j as the greatest common divisor of $j_1, \ldots, j_k$ and $j = 0$ if there are no variables. The sets $L_{i,j}$ are all regular for $j > 0$ as one can just count (modulo j) the number of occurrences of a positively and those of a^{-1} negatively and accept iff the corresponding counter is i and the word is in G. For $j = 0$ they are singletons as each group element has only one representative in G and therefore the sets are again regular. Therefore Sections III and IV deal only with finitely generated free groups having at least two generators.

For the case that G is the set of representatives of a free group with finitely many and at least 2 generators, there are context-free pattern languages in the group G which are not regular, but they need constants. Example 9 below shows that the language generated by the pattern $x_1 a x_1^{-1}$ is context-free.

The main result on finitely generated noncommutative free groups in the present work is the following: The only context-free verbal languages are $\{\varepsilon\}$ and G. This improves a result of Miasnikov and Romankov [10] who had shown that these are the only regular verbal languages. The proof is given in Section III. Section IV shows that many nontrivial verbal languages are context-sensitive; actually they are shown to be indexed languages. In Section II it is shown that for any finitely generated free group with Thurston 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 Thurston automatic representation. This independence result does not hold for all automatic groups: Example 1 and Section V provide a verbal language L and a group in which L is either regular or properly context-free or properly context-senstive 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 VI 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.

II. Complexity of Languages over Free Groups in Dependence of Their Representation

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. This is proven in Theorem 3 below for the lower levels of this hierarchy. Recall that $\sim$ says when two group elements are equal. The proof will use the following auxiliary proposition.

Proposition 2: Let Σ be a finite non-empty set of generators and Σ^* be the set of all words over $\Sigma \cup \Sigma^{-1}$. Let G be the set of reduced words over $\Sigma \cup \Sigma^{-1}$ and G' be any other regular set such that for each $w \in \Sigma^*$ there is exactly one $v \in G'$ with $v \sim w$. Then there is a constant s such that for any word $w \in G$ with $w = w_1 w_2 \ldots w_\ell$ (each w_k consisting of one symbol in $\Sigma \cup \Sigma^{-1}$) there are words $u_0, u_1, \ldots, u_s$ satisfying the following: $v = u_0 w_1 u_1 w_2 u_2 \ldots w_\ell u_\ell \in G'$ and $|u_k| \leq s \wedge u_k \sim \varepsilon$ for each $k \in \{0, 1, \ldots, \ell\}$.

Proof: The proof mainly uses the following: There is a maximal number s such that some $w \in G'$ contains some v of length s as a subword with $v \sim \varepsilon$. For this let n be the number of states of the automaton. First it is claimed that there does not exist a $w \in G'$ satisfying

$$w = \alpha_0 \alpha_1 \ldots \alpha_{n^2} \alpha_{n^2+1} \beta_{n^2+1} \beta_{n^2} \ldots \beta_1 \beta_0 \text{ and}$$
$$\alpha_k \beta_k \neq \varepsilon \wedge \alpha_0 \alpha_1 \ldots \alpha_k \beta_k \ldots \beta_1 \beta_0 \sim w$$

for each $k \in \{0, 1, \ldots, n^2 + 1\}$. Suppose otherwise, that there exists such a w. There are two different $k, k' \in \{0, 1, \ldots, n^2\}$ such that the deterministic finite automaton recognising G' after processing $\alpha_0 \alpha_1 \ldots \alpha_k$ is in the same state as after processing $\alpha_0 \alpha_1 \ldots \alpha_{k'}$ and after processing $\alpha_0 \alpha_1 \ldots \alpha_{n^2} \alpha_{n^2+1} \beta_{n^2+1} \beta_{n^2} \ldots \beta_{k+1}$ the automaton is in the same state as after processing

$\alpha_0\alpha_1 \ldots \alpha_{n^2}\alpha_{n^2+1}\beta_{n^2+1}\beta_{n^2} \ldots \beta_{k'+1}$. This follows from the fact that there are only n^2 possible pairs for the states for each k but n^2+1 possible values for k; so such k, k' can be found; without loss of generality $k < k'$. At least one of the parts $\alpha_{k+1}\alpha_{k+2} \ldots \alpha_{k'}$ and $\beta_{k'} \ldots \beta_{k+2}\beta_{k+1}$ is different from ε (otherwise, G' contains two representatives for w: $\alpha_0\alpha_1 \ldots \alpha_{n^2}\alpha_{n^2+1}\beta_{n^2+1}\beta_{n^2} \ldots \beta_1\beta_0$ and $\alpha_0\alpha_1 \ldots \alpha_k\alpha_{k'+1} \ldots \alpha_{n^2+1}\beta_{n^2+1} \ldots \beta_{k'+1}\beta_k \ldots \beta_1\beta_0$). Now the string

$$v = \alpha_0\alpha_1 \ldots \alpha_k \ \alpha_{k'+1}\alpha_{k'+2} \ldots \alpha_{n^2}\alpha_{n^2+1}$$
$$\beta_{n^2+1}\beta_{n^2} \ldots \beta_{k'+2}\beta_{k'+1} \ \beta_k \ldots \beta_1\beta_0$$

is a member of G'. As $\alpha_0\alpha_1 \ldots \alpha_{k'}\beta_{k'} \ldots \beta_1\beta_0 \sim w$ it follows that

$$\alpha_{k'+1}\alpha_{k'+2} \ldots \alpha_{n^2}\alpha_{n^2+1}\beta_{n^2+1}\beta_{n^2} \ldots \beta_{k'+2}\beta_{k'+1} \sim \varepsilon$$

and $v \sim w$. So v, w are two different elements in G' representing the same group element and such do not exist by assumption.

For the next step one assumes that there is a $w \in G'$, $w = \gamma_0\gamma_1 \ldots \gamma_s\gamma_{s+1}$ with $\gamma_1, \ldots, \gamma_s$ consisting each of one symbol in $\Sigma \cup \Sigma^{-1}$ and $\gamma_1\gamma_2 \ldots \gamma_s \sim \varepsilon$. It is shown that $s \leq (n+1) \cdot (2|\Sigma|)^{n^2+1}$.

Let $\delta_k = repr_G(\gamma_1\gamma_2 \ldots \gamma_k)$ for $k = 1, 2, \ldots, s$ and note that $\delta_0 = \varepsilon$ and $\delta_s = \varepsilon$. Furthermore, either $\delta_{k+1} = \delta_k\gamma_{k+1}$ or $\delta_k = \delta_{k+1}\gamma_{k+1}^{-1}$ for all $k \in \{0, 1, \ldots, s-1\}$.

Assume by way of contradiction that there is a k with $|\delta_k| > n^2$. Now choose $\alpha_0, \alpha_1, \ldots, \alpha_{n^2+1}$ and the corresponding $\beta_0, \beta_1, \ldots, \beta_{n^2+1}$ as follows: Let i_m be the first index where δ_{i_m} consists of the first m symbols of δ_k and let j_m be the last index where δ_{j_m} consists of the first m symbols of δ_k; note that $i_0 = 0$ and $j_0 = s$. Now let $\alpha_0 = \gamma_0$ and $\beta_0 = \gamma_{s+1}$ and, for $m \in \{1, 2, \ldots, n^2\}$, let $\alpha_m = \gamma_{i_{m-1}+1}\gamma_{i_{m-1}+2} \ldots \gamma_{i_m}$ and $\beta_m = \gamma_{j_m+1}\gamma_{j_m+2} \ldots \gamma_{j_{m-1}}$. Let $\alpha_{n^2+1} = \gamma_{i_{n^2}+1}\gamma_{i_{n^2}+2} \ldots \gamma_{j_{n^2}}$ which has positive length and $\beta_{n^2+1} = \varepsilon$. One can now see that these α_m and β_m would satisfy the conditions laid out at the beginning of the proof and hence they cannot exist. Therefore a k as stated does not exist.

Assume now by way of contradiction that there are an u and $n+1$ numbers $h_0, h_1, \ldots, h_n$ such that $\delta_{h_0} = u \wedge \delta_{h_1} = u \wedge \ldots \wedge \delta_{h_n} = u$. Then one can find two different numbers $k, k' \in \{0, 1, \ldots, n\}$ with $k < k'$ and the automaton recognising G' being in the same state after processing $\gamma_0\gamma_1 \ldots \gamma_{h_k}$ and $\gamma_0\gamma_1 \ldots \gamma_{h_{k'}}$. It follows that $v = \gamma_0\gamma_1 \ldots \gamma_{h_k} \ \gamma_{h_{k'}+1}\gamma_{h_{k'}+2} \ldots \gamma_{s+1}$ is an element of G' as well. Furthermore, as $\delta_{h_k} = \delta_{h_{k'}}$, it holds that $\gamma_{h_k+1}\gamma_{h_k+2} \ldots \gamma_{h_{k'}} \sim \varepsilon$ and therefore $v \sim w$. This again would contradict the fact that every group element has a unique representative in G'.

Hence there are at most n indices k for which the corresponding reduced word δ_k is the same and furthermore each δ_k is a word of length up to n^2 over $\Sigma \cup \Sigma^{-1}$. This permits to estimate that $s \leq n \cdot (2|\Sigma|)^{n^2+1}$.

Now let $w \in G$ and $v \in G'$ be given such that $w \sim v$. Then $w = repr_G(v)$. Let $w_1, w_2, \ldots, w_\ell$ be the symbols making up w. As $w = repr_G(v)$ there are $u_0, u_1, \ldots, u_\ell \sim \varepsilon$ with $v = u_0w_1u_1w_2u_2 \ldots w_\ell u_\ell$. As shown above, each u_k has at most length s for some constant $s \leq n \cdot (2|\Sigma|)^{n^2+1}$. ∎

Theorem 3: Let Σ be a finite set of generators and Σ^* be the set of all words over $\Sigma \cup \Sigma^{-1}$. Let G be the set of reduced words over $\Sigma \cup \Sigma^{-1}$ and G' be any other regular set such that for each $w \in \Sigma^*$ there is exactly one $v \in G'$ with $v \sim w$. Let $L \subseteq G$ and $L' = \{v \in G' : \exists w \in L\,[v \sim w]\}$. Then L is context-free iff L' is context-free.

Proof: By Proposition 2 there is a constant s such that for every $v \in L'$ there is a $w = repr_G(v)$ such that $w = w_1w_2 \ldots w_\ell$ for some ℓ and there are $u_0, u_1, u_2, \ldots, u_\ell \sim \varepsilon$ each of length at most s with $v = u_0w_1u_1w_2u_2 \ldots w_\ell u_\ell$. There is a non-deterministic automaton U which is designed to identify the significant symbols $w_1, w_2, \ldots, w_\ell$ by having a set Q of specific states such that the following holds: In each run of U on a $v \in G'$ ending in an accepting state, the subsequence of symbols from v on which U was after parsing the corresponding symbol in a state from Q is exactly w_1, w_2, …, w_ℓ. Furthermore, U accepts exactly the members of G'. Note that the actual choice of the $u_0, u_1, \ldots, u_\ell$ might depend on the nondeterministic run: if $w = a$ and $v = aa^{-1}a$ then one can either have $u_0 = \varepsilon \wedge u_1 = a^{-1}a$ or $u_0 = aa^{-1}$ and $u_1 = \varepsilon$. So essentially what U does is telling which symbols have to be preserved (by being in a state inside Q after having parsed them) and which have to be omitted (by being in a state outside Q after having parsed them). Such a non-deterministic automaton U exists, as it has only to verify three things that it can guess at suitable positions: First each string of symbols marked to be omitted (by not being in a state in Q after passing through them) belongs to the fixed list of strings α with $\alpha \sim \varepsilon$ and $|\alpha| \leq s$; second each two consecutive symbols marked to be kept (by being in a state in Q after passing over them) are not inverting each other; third, the overall word is in G'. The verification that such a non-deterministic finite automaton U exists is left to the reader.

Assume now that L' is context-free. Given a context-free grammar in Chomsky normal form for the language L', one constructs the context-free grammar for L as follows. For each non-terminal A in the grammar for L', one has triples (q, A, r) as non-terminals in the grammar for L, where q and r are states from the non-deterministic automaton U described above; intuitively, q represents the state the automaton is in before reading the word generated by A and r describes the state the automaton is in after reading the word produced by A.

For a production $A \to BC$ in the context-free grammar for L', one has the following productions in the grammar for L: $(q, A, r) \to (q, B, s)(s, C, r)$, for all states q, s, r of the automaton U.

For each production of the form $A \to a$ in the grammar for L', one has the following productions in the grammar for L:

(i) all productions of the form $(q, A, r) \to \varepsilon$, where the non-deterministic automaton U goes from state q to r on input a and $r \notin Q$

(ii) all productions $(q, A, r) \to a$ where the non-deterministic automaton U can go from q to r on

a and $r \in Q$.

Furthermore, the start symbols (they could easily be made unique if one requires this) of the new grammar for L are all triples (q, S, r) where S is a start symbol of the grammar for L' and q is a starting state of U and r is an accepting state of U.

It can easily be seen that the resulting grammar generates L as it preserves the symbols contained in the reduced word and replaces the other ones by ε in the generated word. The conditions on the terminals generated ensure that the choices for omitting or preserving generated symbols are consistent with a run of the non-deterministic automaton U.

On the other hand, if one has a context-free grammar for L then one can first replace every symbol c generated by a non-terminal C which generates all sequences of the form $\alpha c \beta$ with $\alpha \sim \varepsilon$, $\beta \sim \varepsilon$ and $|\alpha|, |\beta| \leq s$. Furthermore, if ε in L then one adds a fixed production from the start symbol to produce the corresponding word in L'. The so constructed new language L'' contains for every word $w \in L$ some $v \in G'$ with $w \sim v$. Furthermore, for each word $v \in L''$ it holds that $red(v) \in L$. Hence $L' = G' \cap L''$ and L' is context-free. ∎

The above theorem can be improved in two directions. First, one might ask what happens if instead of the generators in $\Sigma \cup \Sigma^{-1}$ one uses some other set $\{c_1, c_2, \ldots, c_k\}$ of generators representing words $w_1, w_2, \ldots, w_k$ in Σ^* and takes G'' to be a regular set of the words over $\{c_1, c_2, \ldots, c_k\}$ such that each group element has exactly one representation in G''. For this case too one can show that a subset $L \subseteq G$ is context-free iff the corresponding $L'' = \{w \in G'' : \exists v \in L\, [v \sim w]\}$ is context-free. This can be done as follows. Let $f(c_h) = (aa^{-1})^h bb^{-1} w_h$ for $h = 1, 2, \ldots, k$, $f(uv) = f(u) \cdot f(v)$, $G' = f(G'')$ and $L' = \{w \in G' : \exists v \in L''\, [v \sim w]\}$. Now one can show that G' is regular and L is context-free iff L' is context-free iff L'' is context-free. Hence one can generalise Theorem 3 as below; furthermore, one can show that the result holds not only for context-free languages but also for all levels of the Chomsky hierarchy.

Theorem 4: Let G and G' be Thurston automatic presentations of the same free group and let $L \subseteq G$. Let $L' = \{w \in G' : \exists v \in L\, [v \sim w]\}$. Then L and L' belong to the same levels in the Chomsky hierarchy.

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 $w \in L$ one cannot decide whether $wa^i a^{-j} \in L'$ due to the fact that a finite automaton having processed wa^i does not remember the exact value of i and therefore cannot compare it to j.

III. 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 $\{a, b\} \subseteq \Sigma$ and G is the set of reduced words over $\Sigma \cup \Sigma^{-1}$. The empty string ε represents the neutral element of G.

Theorem 5: A verbal language L over G is context-free iff either $L = \{\varepsilon\}$ or $L = G$.

Proof: Clearly $\{\varepsilon\}$ and G are context-free languages, indeed they are even regular. So, for a proof of the converse direction, let L be a context-free verbal language.

In the following let L be a context-free verbal language generated by a pattern $\pi = x_{i_1}^{j_1} x_{i_2}^{j_2} \ldots x_{i_\ell}^{j_\ell}$ over variables $x_1, x_2, \ldots, x_n$ (which all occur) where $j_1, j_2, \ldots, j_\ell \in \{-1, +1\}$. Without loss of generality one can assume 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 obviously 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. Furthermore, let b^+ denote $\{b, b^2, b^3, \ldots\}$ and b^- denote $\{b^{-1}, b^{-2}, b^{-3}, \ldots\}$.

Definition 6: Let $r = 101 \cdot (2\ell + 1)^{2\ell}$. For each x_i in π, 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 $R_i^{-1} = \{x^{-1} : x \in R_i\} = a^{-i}\ b^{-1}\ a^{-r(i+1)+1}\ b^-\ a^{-r(i+1)+2}\ b^-\ a^{-r(i+1)+3}\ b^-\ \ldots\ a^{-ri-1}\ b^{-1}\ a^{-i}$. Furthermore, let R be the regular language of the reduced words in $R_{i_1}^{j_1} R_{i_2}^{j_2} \ldots R_{i_\ell}^{j_\ell}$.

The only cancellations which can occur when forming the reduced words are those 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 $a^i buba^i$ 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 $a^i buba^{i-i'} b^{-1} u' b^{-1} a^{-i'}$ and those in $R_{i'}^{-1} \cdot R_i$ are of the form $a^{-i'} b^{-1} u' b^{-1} a^{i-i'} buba^i$. This illustrates then that 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.

Claim 7: For each constant s, if one replaces in each R_i each expression b^+ by b^s and in each R_i^{-1} the expression b^- by b^{-s} then the resulting word is in $L \cap R$. This is easily seen by taking for each x_i the value $a^i ba^{ri+1} b^s a^{ri+2} b^s a^{ri+3} b^s \ldots a^{ri+(r-1)} ba^i$ and then considering the word obtained by evaluating π.

Assume that w is a word in a context-free language and that in the derivation one has $S \Rightarrow vAw \Rightarrow vxAyw \Rightarrow vxzyw$. Then one can also generate every word of the form $vx^k zy^k w$. This can also be done by pumping at multiple parts in the language. Therefore one can get the following more general pumping lemma: For each context-free language H there is a constant s with the following properties: each word $w \in H$ longer than s has a partition into $v_1 v_2 \ldots v_k v_{k+1}$ such that no v_j is of length longer than s and that there are non-empty words $u_1, u_2, \ldots, u_k$ and an indexing of exponents $e_1, e_2, \ldots, e_k$ such that for all numbers $h_1, h_2, \ldots, h_k$ the word $v_1 u_1^{h_{e_1}} v_2 u_2^{h_{e_2}} \ldots v_k u_k^{h_{e_k}} v_{k+1}$ is in H and if $e_d = e_{d'}$ and $e_{d''} = e_{d'''}$ for d, d', d'', d''' with $d < d'$ and $d'' < d'''$ then either $d = d'' \wedge d' = d'''$ or $d < d' < d'' < d'''$ or $d'' < d''' < d < d'$ or $d < d'' < d''' < d'$ or $d'' < d < d' < d'''$. As the pumped parts in $R \cap L$ cannot contain any a or a^{-1} (their number is fixed), one can obtain

the following pumping lemma for $R \cap L$.

Claim 8: Given $L \cap R$ there is a constant s such that when one replaces b^+ by b^s and b^- by b^{-s} then one obtains a word w which can be partitioned into $v_1 v_2 \ldots v_k v_{k+1}$ such that no part is longer than s and that there are constants $c_1, c_2, \ldots, c_k > 0$ and an indexing of exponents $e_1, e_2, \ldots, e_k$ such that for all numbers $h_1, h_2, \ldots, h_k$ the word $v_1 b^{c_1 \cdot h_{e_1}} v_2 b^{c_2 \cdot h_{e_2}} \ldots v_k b^{c_k \cdot h_{e_k}} v_{k+1}$ is in $L \cap R$ and if $e_d = e_{d'}$ and $e_{d''} = e_{d'''}$ for d, d', d'', d''' with $d < d'$ and $d'' < d'''$ then either $d = d'' \wedge d' = d'''$ or $d < d' < d'' < d'''$ or $d'' < d''' < d < d'$ or $d < d'' < d''' < d'$ or $d'' < d < d' < d'''$. Furthermore, by taking $h_{k'} = \tilde{c}^{k'}$ for a sufficiently large constant $\tilde{c}$, one obtains that any two intervals of s symbols b or b^{-1} (with a or a^{-1} on either side of the interval) are made so long that they become different unless they are pumped by the same occurrences of $b^{c_{k'} \cdot h_{k'}}$ or respectively $b^{-c_{k'} \cdot h_{k'}}$ in it (that is, either these are pumped into both intervals or into none of them). Such intervals are called linked. Let w' refer to the corresponding word.

As w' is in L, there are values $y_1, \ldots, y_n$ of the variables $x_1, \ldots, x_n$ which generate w'. Let Y be the set of all words obtained 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. Now one declares k and $k+1$ 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 $ba^{ri+k}b^t a^{ri+k+1}b$ 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 $ba^{ri+k}b$, $ba^{ri+k+1}b$, $b^{-1}a^{-ri-k}b^{-1}$, $b^{-1}a^{-ri-k-1}b^{-1}$ is a substring of uu''' in a way that it spans over both parts of uu'''.
- For some t and $v = ba^{ri+k}b^t a^{ri+k+1}b$, there are either two occurrences of v in w' or two occurrences of v^{-1} in w'.

The first can happen for at most $100 \cdot |Y|^2$ many k: 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 only constantly many ways are there that a k gets invalid and that the k is determined as being a number for which one of $ba^{ri+k}b$ or $ba^{ri+k+1}b$ or $ba^{ri+k-1}b$ or the inverse appears over the middle border in one of the words $uu', u''u''', uu'''$ or is separated from the middle border only by some word in b^*. The factor 100 is a safe upper bound of the optimal constant for this inequality.

The second case can also happen only if there are two linked intervals of this form and they must both refer to two instances of x_i or two instances of x_i^{-1} but not in a mixed way; furthermore, this happens for each pair of occurrences of x_i only once as the linked intervals cannot be linked in a overlapping way; hence for each pair of occurrences of variables there are only two values of k invalidated and so the overall number of values of k invalidated by this process is $2\ell^2$.

In total at most $100 \cdot (2n+1)^{2\ell} + 2\ell^2$ possible values of k are invalidated. Hence there is a choice of k such that k is not invalidated and $1 \leq k < r-2$. Taking r as $101 \cdot (2\ell+1)^{2\ell}$ as done in Definition 6 gives a safe value for r to guarantee the existence of a k which is not invalidated. There are now two cases.

The first case is that for every occurrence in w' of $ba^{ri+k}b^t a^{ri+k+1}b$ there is in w' also an occurrence of $b^{-1}a^{-ri-k-1}b^{-t}a^{-ri-k}b^{-1}$. Then the corresponding intervals 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 intervals 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 section. Hence the first case does not occur.

The second case is that there is an unlinked occurrence. Then one has that some substring $v = ba^{ri+k}b^t a^{ri+k+1}b$ occurs exactly once in w'. By the choice of k one has that when forming the word w' as $y_{i_1}^{j_1} \ldots y_{i_\ell}^{j_\ell}$, the number of occurrences of v in w' is the sum over all j_h times the occurrences of v in y_{i_h} where occurrences of v^{-1} count negatively. Hence, one can for $h = 1, 2, \ldots, n$ choose z_h to be the concatenation of all occurrences of v and v^{-1} in y_h and then obtains that $z_{i_1}^{j_1} \ldots z_{i_\ell}^{j_\ell}$ equals v; hence, one can, by substitution of v by any other reduced word w'', generate w'' with the pattern π. This means that the corresponding verbal language is the full group. This completes the proof of Theorem 5. ∎

Example 9: The language generated by $x_1 a x_1^{-1}$ is context-free but not regular.

Proof: Taking any value u of x_1 which does not end with a or a^{-1}, it is clear that the resulting word uau^{-1} is generated in a cancellation-free way. Furthermore, in the case that $x_1 = ua^k$ one gets again the same value uau^{-1}. Therefore the language given by $x_1 a x_1^{-1}$ is, for the case of $\Sigma = \{a, b\}$, generated by the following productions starting with S:

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

An easy application of the pumping lemma shows that this language is not regular. ∎

It is open whether there are any pattern languages (over some G) which are regular, besides singletons like $\{w\}$ for $w \in G$ and G itself. Such examples can be obtained for the more general class of first-order definable languages where the first-order definition can use constants from G. 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

are values for the variables such that every pattern generates w with these values.

Example 10: There are non-trivial regular and properly context-free existentially first-order definable sets. Examples are the sets $(aa)^*$ and a^*b^* in the case of regular languages and $\{b^nab^n : n \in \mathbf{Z}\}$ in the case of properly context-free languages.

Proof: One can define the non-trivial regular set $(aa)^*$ as $(aa)^* = \{w : \exists x_1[w \sim x_1x_1aa \wedge w \sim aax_1x_1]\}$. One can define a^*b^* as $a^*b^* = \{w : \exists x_1 \exists x_2[w \sim ax_1bx_2 \wedge w \sim x_1abx_2 \wedge w \sim ax_1x_2b \wedge w \sim x_1ax_2b]\}$.

The language of all b^nab^n is properly context-free and first-order defined by $w \in \{b^nab^n : n \in \mathbf{Z}\}$ iff $\exists x_1[w \sim x_1bax_1b \wedge w \sim bx_1abx_1]$. ∎

IV. Verbal Sets and Indexed Languages

Many verbal languages fall into the class of indexed languages which are a proper subclass of the context-sensitive languages. These languages are generated by indexed grammars. Indexed grammars are like context-free grammars except that there are two kinds of nonterminals. There are ordinary nonterminals, which play the same role as they do in context-free grammars; and there are so-called indices, which occur in sentential forms only to the right of ordinary nonterminals.

Definition 11 (Aho [1]): An indexed grammar consists of a number of pairwise disjoint finite nonempty sets together with a designated start symbol. The sets are as follows.

1. A set of terminal symbols $\Sigma = \{a, b, c, \ldots\}$;
2. A set of ordinary nonterminals $\Delta = \{A, B, C, \ldots\}$;
3. A set of special nonterminals called indices $\Theta = \{f, g, h, i, j, k, \ldots\}$;
4. A set of productions $\mathcal{P} \subset \Delta \times (\Sigma + \Delta)^* + \Delta \times \Delta\Theta + \Delta\Theta \times (\Sigma + \Delta)^*$.

The start symbol is a fixed nonterminal in Δ.

The start symbol will be S unless said otherwise and productions are written as $\alpha \to \beta$ instead of (α, β). Below are examples of the three types of productions. Notice that indices are written as subscripts.

$A \to BaC$	a context-free production.
$A \to B_i$	a production which produces an index i.
$A_j \to BaC$	a production which consumes an index j.

Productions are applied to sentential forms to yield other sentential forms by direct derivation. Sentential forms are strings of terminals and nonterminals in which indices occur only to the right of ordinary nonterminals. Terminal letters do not have indices following them. In other words sentential forms are elements of $(\Sigma+\Delta\Theta^*)^*$. The direct derivation works as follows.

$$\left.\begin{array}{l} A \to BaC \\ A \to B_i \\ A_j \to BaC \end{array}\right\} \text{ applied to } \left\{\begin{array}{l} DB_{jk}aC_{jk}E_{mn} \\ DB_{ijk}E_{mn} \\ DB_kaC_kE_{mn} \end{array}\right. \text{ yields}$$

The reflexive transitive closure of direct derivation is called derivation. Furthermore, let $\gamma \to \gamma'$ denote a direct derivation and $\gamma \overset{*}{\to} \gamma'$ a derivation in general. The language generated by a grammar is the set of words over the terminal alphabet derivable from the start symbol.

Example 12: The language $L = \{a^{2^n} \mid n \geq 0\}$ is generated by the indexed grammar

$$S \to T_i, \quad T \to T_j, \quad T \to A, \quad A_j \to AA, \quad A_i \to a.$$

Indeed it is clear that all derivations begin with $S \overset{*}{\to} A_{j^m i}$ and a straightforward induction on m shows that a^{2^m} is the unique word in the terminal alphabet derivable from $Aj^m i$.

Theorem 13 (Aho [1]): Indexed languages are closed under the rational operations of union, product and generation of submonoid. They are also closed under intersection with regular languages.

Proposition 14: Let Σ be a finite alphabet with formal inverses Σ^{-1} and let ρ be a pattern consisting only of variables and possibly their negations. Then the language L' obtained by chosing for any variable x_i a (possibly empty) string $u_{i,1}u_{i,2}\ldots u_{i,j_i}$ over $\Sigma \cup \Sigma^{-1}$ and for x_i^{-1} the corresponding inverse string $u_{i,j_i}^{-1} \ldots u_{i,2}^{-1}u_{i,1}^{-1}$ put together by concatenation (without doing cancellations) has an indexed grammar.

Proof: The basic idea is the following: Suppose the variables used in the pattern are $x_0, x_1, \ldots, x_n$. One first generates, for the start symbol S, an index which contains, for each variable in the pattern, its value in a coded way. Each of these values are separated by a special character y and the end of the coding is marked by $\#$. In the next step, the start symbol S derives a string representing the pattern, where each symbol stands for one of the variables in the pattern or their inverses or the constants. Each symbol representing a variable or its inverse receives a full copy of the index and operates from now on independently of the other parts of the derivation.

If a symbol represents x_k, then the derivation from it skips all the symbols in the index until y is skipped k times; if the symbol stands for x_0, this skipping phase is void. Once the above is done, the derivation transforms each current index symbol to the corresponding group generator / inverted group generator at the left side of itself until another y or the end of the index is reached. Afterwards the rest of the index is ignored and the variable symbol is transformed into ε.

If the symbol stands for x_k^{-1}, then the derivation again skips all symbols in the index until y has been skipped k times. Then the derivation transforms each current index symbol to the inverse of the corresponding group generator / inverted group generator at the right side of the symbol representing the variable. If the end or another y is reached, the remaining part of the index is ignored and the variable symbol is transformed into ε.

Note that in both cases, one employs several nonterminals in order to code the information about how many y have been skipped so far.

Now, the construction is shown in detail for the following illustrative case given by $\Sigma = \{a, b\}$, $\rho = XYX^{-1}Y^{-1}$ and $L' = \{wvw^{-1}v^{-1} \mid w, v \text{ are words over } \Sigma \cup \Sigma^{-1}\}$. It is shown that L' is generated by the following indexed grammar.

1. Terminals $\Sigma \cup \Sigma^{-1} = \{a, b\} \cup \{a^{-1}, b^{-1}\}$;

2. Ordinary nonterminals $\{S, T, U, X, Y, Z, X^{-1}, Y^{-1}, Z^{-1}\}$ with start symbol S;
3. Indices $\{a, b, a^{-1}, b^{-1}, y, \#\}$;
4. Productions
   (a) $S \to T_\#$
   (b) $T \to T_a \mid T_b \mid T_{a^{-1}} \mid T_{b^{-1}} \mid U_y$
   (c) $U \to U_a \mid U_b \mid U_{a^{-1}} \mid U_{b^{-1}} \mid XYX^{-1}Y^{-1}$
   (d) $X_i \to iX$ for all indices $i = a, b, a^{-1}, b^{-1}$
   (e) $X_y \to \varepsilon$ (the empty word)
   (f) $X_i^{-1} \to X^{-1}i^{-1}$ for all indices $i = a, b, a^{-1}, b^{-1}$
   (g) $X_y^{-1} \to \varepsilon$
   (h) $Y_i \to Y$ for all indices $i = a, b, a^{-1}, b^{-1}$
   (i) $Y_y \to Z$
   (j) $Z_i \to iZ$ for all indices $i = a, b, a^{-1}, b^{-1}$
   (k) $Z_\# \to \varepsilon$ (the empty word)
   (l) $Y_i^{-1} \to Y^{-1}$ for all indices $i = a, b, a^{-1}, b^{-1}$
   (m) $Y_y^{-1} \to Z^{-1}$
   (n) $Z_i^{-1} \to Z^{-1}i^{-1}$ for all indices $i = a, b, a^{-1}, b^{-1}$
   (o) $Z_\#^{-1} \to \varepsilon$ (the empty word)

Productions (a)–(c) derive precisely the sentential forms of $X_\sigma Y_\sigma X_\sigma^{-1} Y_\sigma^{-1}$ where σ is a string of indices $wyv\#$, and w and v are arbitrary words over $\Sigma \cup \Sigma^{-1}$. Examination of productions (d)–(g) shows that X_σ derives w and X_σ^{-1} derives w^{-1}. From productions (h), (i) it follows that Y_σ derives $Z_{v\#}$ and productions (j), (k) show that $Z_{v\#}$ derives v. Likewise productions (l)–(o) show that $Z_{v\#}^{-1}$ derives v^{-1}. ∎

Definition 15: Call a pattern language L cancellation-free iff there are patterns $\pi_1, \pi_2, \ldots, \pi_n$ such that each pattern generates a subset of L and for each word $w \in L$ there is a pattern π_k, $1 \leq k \leq n$, and an assignment of values to the variables such that π_k generates w in a cancellation-free way with this assignment.

Theorem 16: Every cancellation-free pattern language $L \subseteq G$ is an indexed language.

Proof: Recall that G is the regular set of all reduced words over $\Sigma \cup \Sigma^{-1}$. By Proposition 14 above, the language L_m of all strings obtained by simply concatenating the values for the variables in π_m (and not doing any cancellations) is generated by an indexed grammar. The language L equals to $G \cap (L_1 \cup L_2 \cup \ldots \cup L_n)$; as each L_m is an indexed language, so is their union and also the intersection of that union with the regular set G. ∎

As examples of cancellation-free pattern language is ax_1x_1 generated by $\pi_1 = ax_2x_1x_1x_2^{-1}$, $\pi_2 = x_2x_1x_1x_2^{-1}a$ and $\pi_3 = x_1a^{-1}x_1$. Furthermore, the verbal language generated by $x_1x_1x_1$ is a cancellation-free pattern language as witnessed by $\pi_1 = x_2x_1x_1x_1x_2^{-1}$. However, not every pattern language is cancellation-free.

Example 17: Assume that $a, b, c, d \in \Sigma$. The language L generated by $x_1ax_1^{-1}bx_1cx_1^{-1}$ is not cancellation-free.

Proof: Assume that a pattern ρ generating a sublanguage of L is generating infinitely many words of the form $c^nac^{-n}bc$ in a cancellation-free way. Each variable x_i in ρ must occur as often uninverted as inverted; the reason is that otherwise one could assign to x_i the value d and to all other variables the value ε and would then obtain a word in which the occurrences of d and d^{-1} are not balanced although L does not have such a word. Thus a and b must appear as constants, as otherwise a variable containing a or b in its value would just occur exactly once without its inverse occurring in ρ. So each variable takes as value an element of c^*. Now one changes in the values of the variables all c to d and all c^{-1} to d^{-1}. The resulting word y satisfies that before a one has only occurrences of c and d, between a and b only occurrences of c^{-1} and d^{-1} and after the b there is either one c or one d. As the variables are not void (for n being sufficiently large), there occur at least some d. Assume now that $x_1ax_1^{-1}bx_1cx_1^{-1}$ generates y and that the value of x_1 is of the form uvw where v is the part from the first to the last occurrence of d or d^{-1} in the value of x_1. Note that the parts waw^{-1}, $u^{-1}bu$ and wcw^{-1} are of odd length and do therefore not vanish in the word $y = uvwaw^{-1}v^{-1}u^{-1}buvwcw^{-1}v^{-1}u^{-1}$; hence the parts v and v^{-1} are all separted by nonvoid members of $\{a, b, c\}^*$ and do not cancel out. Therefore, after cancellations, the resulting word is in $\{a, b, c\}^* v \{a, b, c\} v^{-1} \{a, b, c\}^* v \{a, b, c\}^* v^{-1} \{a, b, c\}^*$ and there are at least three alternations between d and d^{-1} in the word. Hence the resulting word differs from y. This contradiction shows that no pattern ρ generates a subset of L and in addition produces cancellation-free infinitely many words of the form $c^nac^{-n}bc$. Hence L cannot be a cancellation-free pattern language. ∎

It is unknown whether every verbal language is a cancellation-free language.

V. When Representations Matter

While the previous sections dealt with a free group having at least two and at most finitely many generators, this section investigates the complexity of verbal sets in a group where the representation is crucial. The following definition fixes G, G', G'' for this section.

Definition 18: Let a group $G = \{\varepsilon, a\} \cdot b^* \cdot c^* \cdot d^*$ with generators $\Sigma = \{a, b, c, d\}$ have the group operations be obtained from concatenation by taking the equations $a = a^{-1}$, $ab = b^{-1}a$, $ab^{-1} = ba$, $ac = ca$, $ad = da$, $bc = cb$, $bd = db$ and $cd = dc$ into account. Let $G' = \{\varepsilon, a\} \cdot (bcd)^* \cdot c^* \cdot d^*$ and $G'' = \{\varepsilon, a\} \cdot (bc)^* \cdot c^* \cdot d^*$ be alternative representations of G.

Theorem 19: The group is Thurston biautomatic in representation G and Thurston one-sided automatic (but not biautomatic) in representations G' and G''.

Proof: First one has to see that G is a group (which needs of course only to be done in one representation). For this, note that that the product of $(a^h b^k c^m d^n) \cdot (a^{h'} b^{k'} c^{m'} d^{n'}) \cdot (a^{h''} b^{k''} c^{m''} d^{n''})$ is $a^{h+h'+h''-2hh'-2hh''-2h'h''+4hh'h''} b^{k(1-2h'-2h''+4h'h'')} \cdot b^{k'(1-2h'')+k''} c^{m+m'+m''} d^{n+n'+n''}$ independent of the order of the group operations and so the law of associativity holds (where $h, h', h'' \in \{0, 1\}$ and $k, k', k'', m, m', m'', n, n', n'' \in \mathbf{Z}$). Furthermore, the inverse of $a^h b^k c^m d^n$ is $a^h b^{(2h-1)k} c^{-m} d^{-n}$ and all elements in G are unique group elements with ε being the neutral element.

To see the Thurston automaticity, let $x = a^h b^k c^m d^n$ with $h \in \{0,1\}$ and $k, m, n \in \mathbf{Z}$. 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. One can see that all operations move each part of the word only by at most 5 positions (note that a, b, c, d, b^{-1}, c^{-1} and d^{-1} all occupy one position in a word). For the multiplication from the other side in the representation G, only the case $x \mapsto xa$ is not standard, but this one follows the rule $a^h b^k c^m d^n \mapsto a^{1-h} b^{-k} c^m d^n$ and so the positions shift only by one and b, b^{-1} become interchanged. In G', G'' this is not the case as $a^h (bcd)^k c^{m-k} d^{n-k} \mapsto a^{1-h} (bcd)^{-k} c^{m+k} d^{n+k}$ and correspondingly also for G''; both mappings are not automatic. ∎

Theorem 20: In the presentation G all pattern languages are regular.

Proof: Given a pattern π, one can replace each variable x_i by either y_i or ay_i where y_i is then only of the form $b^{k_i} c^{m_i} d^{n_i}$. This gives a finite set of patterns, S, with a restricted range for the variables. Now, one can use that $ab^k = b^{-k}a$ and that the y_i all commute with each other and with b, c, d in order to move the constant parts to the front and sort the variables. This gives, for each language generated by a pattern in S, a regular expression of the form

$$\alpha \cdot b^{q_1 k_1 + q_2 k_2 + \ldots + q_\ell k_\ell} \cdot c^{r_1 m_1 + r_2 m_2 + \ldots + r_\ell m_\ell} \cdot d^{r_1 n_1 + r_2 n_2 + \ldots + r_\ell n_\ell} = \alpha \cdot (b^{q'})^* \cdot (c^{r'})^* \cdot (d^{r'})^*,$$

where α consists of all constants in the expression and q' is the greatest common divisor of $q_1, q_2, \ldots, q_\ell$ and r' is the greatest common divisor of $r_1, r_2, \ldots, r_\ell$. Note that r' is 0 iff $r_1, r_2, \ldots, r_\ell$ are 0 and accordingly for q'. Now $\alpha = a^h b^{q''} c^{r''} d^{r'''}$ for some constants $h \in \{0,1\}$ and $q'', r'', r''' \in \mathbf{Z}$ and one can bring the expression into the form

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

which is then a subset of G. Then the language generated by π is the finite union of such expressions. ∎

Theorem 21: The language L generated by $x_1 x_2 x_1^{-1} x_2^{-1}$ has different levels in the Chomsky hierarchy in G, G' and G''.

Proof: All occurrences of c, d in x_1, x_2 cancel out and a word in L is non-empty in the case that the values of x_1 and x_2 do not commute, that is, that at least one of them contains an a. Now $ab^n \cdot ab^m \cdot b^{-n} a \cdot b^{-m} a = b^{2m-2n}$, $ab^n \cdot b^m \cdot b^{-n} a \cdot b^{-m} = b^{-2m}$ and $b^n \cdot ab^m \cdot b^{-n} \cdot b^{-m} a = b^{2n}$; hence L equals to the set of all b^{2n} in G, L equals to the set of all $(bcd)^{2n} c^{-2n} d^{-2n}$ in G' and L equals to the set of all $(bc)^{2n} c^{-2n}$ in G''. So the language corresponding to L is regular in G, properly context-free in G'' and not context-free in G'. ∎

Theorem 22: All pattern languages are context-free in G''.

Proof: As shown in Theorem 20, a given pattern language in G can be brought into 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 $q', q'', r', r'', r''' \in \mathbf{Z}$. Translated into G'', this expression gives the set

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

and it is easy to see that this set is context-free. ∎

VI. 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 Thurston-automatic representation G' of the same group, there is a verbal language which is not regular in G'. Throughout this section, G and G' are fixed as in the definition below.

Definition 23: Let $G = H \cdot c^* \cdot d^*$ where $H = \{\varepsilon, a, b, ab, ba, aba, bab, abab\}$ and $ac = c^{-1}a$, $ad = da$, $bc = db$, $bd = cb$, $cd = dc$. Furthermore, for $\alpha, \beta, \gamma \in H$ it holds that $\alpha\beta = \gamma \Leftrightarrow \forall x, y \in c^* d^* [\alpha\beta x = y\alpha\beta \Rightarrow \gamma x = y\gamma]$. All other multiplications between members of G are derived from these rules.

Note that each member of H realises a mapping when moved over an $x \in c^* d^*$ and the members of H are multiplied according to the concatenations of these mappings.

Remark 24: G with the above defined operation is a Thurston automatic group.

Theorem 25: 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. As in Theorem 20 one replaces every variable x_i by an expression of the form $\alpha_i c^{n_i} d^{m_i}$ and then considers for each $(\alpha_1, \ldots, \alpha_\ell) \in H^\ell$ the corresponding pattern ρ which can be brought into the form

$$\alpha \cdot c^{o_1 n_1 + p_1 m_1 + \ldots + o_\ell n_\ell + p_\ell m_\ell} \cdot d^{q_1 n_1 + r_1 m_1 + \ldots + q_\ell n_\ell + r_\ell m_\ell}$$

where α can then be brought into the form $\beta c^s d^t$ with $\beta \in H$ and $s, t \in \mathbf{Z}$. Now one can make the following context-free grammar for this word (with the additional constraint that $b^s c^t$ go into the right place): $S \to \beta O_1$, $O_1 \to c^{o_1} O_1 d^{q_1} | P_1$, $P_1 \to c^{p_1} P_1 d^{r_1} | O_2$, …, $O_\ell \to c^{o_\ell} O_\ell d^{q_\ell} | P_\ell$, $P_\ell \to c^{p_\ell} P_\ell d^{r_\ell} | c^s d^t$; where S is the starting symbol and $O_1, \ldots, O_\ell$ and $P_1, \ldots, P_\ell$ are the other non-terminals. The language generated by π is the finite union of such context-free languages. ∎

Note that in the following result the choice of the pattern depends on the presentation G' and it can be assumed that this cannot be avoided.

Theorem 26: Assume that G' is regular and contains for every element of $v \in G$ exactly one member $w \in G'$ with

$w \sim v$. Then there is a verbal language L in 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 is created and in the second part of the proof it is shown that given any regular representation G', one of these verbal languages is not regular in G'. The first part is done over G in order to keep notation simple.

First part. For every word x_1, note that x_1^4 is a member of the subgroup represented as c^*d^*. This mainly follows from the fact that the elements in H have this property and that the members of c^*d^* remain in c^*d^* when a member of H is moved over them. Note that $aa \sim \varepsilon$ and $bb \sim \varepsilon$. Hence aba and bab are also self-inverse. $abab \sim baba$ as both send $c^m d^n$ to $c^{-m}d^{-n}$. Hence $abab$ is self-inverse and $(ab)^4 \sim (ba)^4 \sim \varepsilon$. Note that due to the permutations of c and d and possible negations involved, $(\alpha c^m d^n)^4$ is mapped to either $c^{4m}d^{4n}$ or $c^{2m+2n}d^{2m+2n}$ or $c^{2m-2n}d^{2n-2m}$ or c^{4m} or d^{4n} or ε.

For the next step one introduces two new variables x_2, x_3 and forms several patterns which map x_1^4, which is equivalent to $c^{m'}d^{n'}$ derived from m, n as indicated above, as follows: $\rho_1 = x_1^4 x_2^2 x_1^4 x_2^{-2}$, $\rho_2 = \rho_1 x_3^2 \rho_1 x_3^{-2}$, $\rho_3 = \rho_2 x_2 x_3 x_2 x_3 \rho_2^{-1} x_3^{-1} x_2^{-1} x_3^{-1} x_2^{-1}$, $\rho_4 = x_2 \rho_3 x_2^{-1} \rho_3$ and $\rho_5 = x_3 \rho_4 x_3^{-1} \rho_4$.

Note that each pattern $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5$ takes as values only members from c^*d^*; furthermore, let $\beta, \gamma \in H$ be such that x_2 is in βc^*d^* and $x_3 \in \gamma c^*d^*$, note that the form of the patterns make only β and γ be relevant for further investigation. In the case that one of $\rho_1, \rho_2, \ldots, \rho_5$ is ε then every subsequent one of these patterns takes the same value.

Assume now that x_1, x_2, x_3 are chosen such that ρ_5 is different from ε. Then $\rho_1 \neq \varepsilon$ and $\beta^2 \neq abab$ as $abab x_1^4 (abab)^{-1} \sim x_1^{-4}$. Hence $\beta \notin \{ab, ba\}$. Similarly $\gamma \notin \{ab, ba\}$ as otherwise ρ_2 would take the value ε. Furthermore, for $\rho_3 \neq \varepsilon$ it is needed that $\beta\gamma\beta\gamma \neq \varepsilon$. So $\beta\gamma$ is either ab or ba, as all others are self-inverses. So $\beta\gamma\beta\gamma = abab$. Furthermore, neither β nor γ can be $abab$ or ε, as otherwise again $\beta\gamma\beta\gamma = \varepsilon$. As the fourth power of every member of H is ε, $\beta \neq \gamma$. As $\beta\gamma \neq abab$ and $abab = baba$, it cannot be 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 then it would be ab or ba. So one has that one of them is a or bab and the other one is b or aba. Furthermore, $\rho_3 = c^{8m'}d^{8n'}$. For ρ_4 and ρ_5, see the following table.

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	$c^{8m'-8n'}d^{-8m'+8n'}$	$c^{16m'-16n'}$

Recall that one could choose m', n' either freely as multiples of 4 or both as even numbers satisfying $m' = n' \vee m' = -n'$. So ρ_5 generates $\{c^{32k}d^{32k}, c^{32k}d^{-32k}, c^{64k}, d^{64k} : k \in \mathbf{Z}\}$.

Now let $\pi_{i,j} = (\rho_5)^i x_4 (\rho_5)^j x_4^{-1}$ where $x_4 \in \delta c^*d^*$ for some $\delta \in H$. Note that the value of $\pi_{i,j}$ only depends on the value of ρ_5 and the $\delta \in H$. Here $\delta cd\delta^{-1} = cd$ for $\delta \in \{\varepsilon, b\}$, $\delta cd\delta^{-1} = cd^{-1}$ for $\delta \in \{ba, bab\}$, $\delta cd\delta^{-1} = c^{-1}d$ for $\delta \in \{a, ab\}$ and $\delta cd\delta^{-1} = c^{-1}d^{-1}$ for $\delta \in \{aba, abab\}$; furthermore, $\delta cd^{-1}\delta^{-1} = cd$ for $\delta \in \{ab, bab\}$, $\delta cd^{-1}\delta^{-1} = cd^{-1}$ for $\delta \in \{\varepsilon, aba\}$, $\delta cd^{-1}\delta^{-1} = c^{-1}d$ for $\delta \in \{b, abab\}$ and $\delta cd^{-1}\delta^{-1} = c^{-1}d^{-1}$ for $\delta \in \{a, ba\}$; furthermore, $\delta c\delta^{-1} \in \{c, c^{-1}, d, d^{-1}\}$ and $\delta d\delta^{-1} \in \{c, c^{-1}, d, d^{-1}\}$. Let $L_{i,j}$ be the language generated by $\pi_{i,j}$; $L_{i,j} = (c^{32(i+j)} \cdot d^{32(i+j)})^* \cup (c^{32(i+j)} \cdot d^{32(-i+j)})^* \cup (c^{32(i+j)} \cdot d^{32(i-j)})^* \cup (c^{32(i+j)} \cdot d^{32(-i-j)})^* \cup (c^{32(i-j)} \cdot d^{32(i+j)})^* \cup (c^{32(i-j)} \cdot d^{32(-i+j)})^* \cup (c^{32(i-j)} \cdot d^{32(i-j)})^* \cup (c^{32(i-j)} \cdot d^{32(-i-j)})^* \cup (c^{64(i+j)})^* \cup (c^{64(i-j)})^* \cup (d^{64(i+j)})^* \cup (d^{64(i-j)})^* \cup (c^{64i} \cdot d^{64j})^* \cup (c^{64i} \cdot d^{-64j})^* \cup (c^{64j} \cdot d^{64i})^* \cup (c^{-64j} \cdot d^{64i})^*$.

Second part. Now let a regular G' be given in which every group element of G has a unique representative. Given a word $w \sim c^m d^n \alpha$ with $\alpha \in H$, let $f_c(w) = m$ and $f_d(w) = n$ and $f_H(w) = \alpha$. Furthermore, assume that whenever an automaton recognising G' after reading v and w is in the same state then $f_H(v) = f_H(w)$; this can easily be obtained by increasing the number of states. Let p be the number of states of the so obtained automaton and let $q = (128p^3)!$.

One important property is that, for all x, whenever the automaton is in the same state after u and ux then there is a word $x' \in c^*d^*$ such that $ux^k u' \sim x'^k uu'$ for all u, u' and k. Furthermore, $f_c(ux)$ and $f_d(ux)$ differ each from $f_c(u)$ and $f_d(u)$, respectively, at most by the number of symbols in x. Now one of the following two cases holds 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 $w_0 w_1 w_2 \ldots w_p w_{p+1}$ such that $|w_1 w_2 \ldots w_p| \leq 4p^3$ and $f_d(w_0 w_1 \ldots w_k)$ 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 the state of the automaton recognising G' is the same after the inputs $w_0 w_1 \ldots w_i$ and $w_0 w_1 \ldots w_j$; due to the state repetition one has that $w_0 \ldots w_i (w_{i+1} \ldots w_j)^+ w_{j+1} \ldots w_{p+1} \subseteq G'$ and the value f_d of all these words is the same; in the following abbreviate this language as ux^+u'. Now note that $r \leq 4p^3$ for $r = |f_c(u) - f_c(ux)|$. Furthermore, $f_H(u) = f_H(ux)$ as the automaton for G' is in the same state before and after x; therefore $r > 0$ as otherwise uxu' and $uxxu'$ would represent the same group element in G'. Let $v = ux \cdot x^{q \cdot \ell/r} u'$; either $v \sim c^{(q+1)q\ell}d^{q\ell}$ or $v \sim c^{(q-1)q\ell}d^{q\ell}$.

Case (b). The word w has a proper splitting $w_0 w_1 \ldots w_{2p^2} w_{2p^2+1}$ such that $|w_1 w_2 \ldots w_{2p^2}| \leq 4p^3$ and $f_d(w_0 w_1 \ldots w_k)$ takes its maximum or minimum value (among $f_d(w_0 w_1 \ldots w_{k'}$, for $k' \leq 2p^2$) at $k = p^2$ and $f_d(w_0 w_1 \ldots w_k)$ differs from this extreme value by $|p^2 - k|$ for $k = 0, 1, \ldots, 2p^2$.

Now there are i, j with $0 \leq p^2 - i < p^2 - j \leq p^2 + j < p^2 + i \leq 2p^2$ such that the state of the

automaton recognising G' is the same after $w_0 \dots w_{p^2-i}$ and $w_0 \dots w_{p^2-j}$ as well as the same after $w_0 \dots w_{p^2+i}$ and $w_0 \dots w_{p^2+j}$. Hence taking $u = w_0 \dots w_{p^2-i}$, $x = w_{p^2-i+1} \dots w_{p^2-j}$, $u' = w_{p^2-j+1} \dots w_{p^2+j}$, $y = w_{p^2+j+1} \dots w_{p^2+i}$, $u'' = w_{p^2+i+1} \dots w_{2p^2+1}$ gives the following properties: $f_d(ux^k u' y^k u'') = f_d(uxu'yu'')$ for all $k > 0$, $f_c(ux^k u' y^k u'')$ is $f_c(uxu'yu'') + (k-1)r$ or $f_c(uxu'yu'') - (k-1)r$ for some r with $0 < r \leq 4p^3$ and all $k > 0$; $f_H(ux^k u' y^k u'') = \varepsilon$ for all $k > 0$. Note that here $u' = \varepsilon$ if $j = 0$. It follows that $v = uxx^{q\ell/r} u' y y^{q\ell/r} u''$ satisfies either $v \sim c^{(q+1)q\ell} d^{q\ell}$ or $v \sim c^{(q-1)q\ell} d^{q\ell}$.

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 $w_0 w_1 \dots w_{2p^2+1}$ of w such that, for $i \leq 2p^2$, $w_0 \dots w_{p^2} = u$, $|f_d(w_i)| = |f_d(u)| - |i - p^2|$, $|w_0| \geq |u| - 2p^3$, and $|w_{2p^2}| \geq |u| - 2p^3$ 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 assume by way of contradiction that the G'-version of the verbal languages $L_{q/2,q/2-1}$ and $L_{q/2+1,q/2}$ are both regular. Then so is their union. Let s be the number of states of an automaton recognising the G'-version L' of $L_{q/2,q/2-1} \cup L_{q/2+1,q/2}$. Note that $w \in L$.

Taking now $\ell > s!$ one has that the v made is of the form $uxx^{q\ell/r}u'$ in the case of (a) and of the form $uxx^{q\ell/r}u'yy^{q\ell/r}u''$ in case of (b). One can verify that in both cases $v \in L'$. Whenever the automaton recognising L' goes over at least $s!$ repetitions of x, it is in the same state after reading $s!$ further x. Hence, for $\ell > s!$ and for every t the word of the form $\tilde{w}_t = uxx^{q\ell/r+ts!}u'$ in case (a) and the word of the form $\tilde{w}_t = uxx^{q\ell/r+ts!}u'yy^{q\ell/r+ts!}u''$ in case (b) is accepted by the automaton and satisfies $f_d(\tilde{w}_t) = f_d(w)$.

There are infinitely many different such $\tilde{w}_t$ which all represent different elements but there are only finitely many elements of L' which have the same nonzero value of f_d.

This contradiction gives that L' cannot be regular. Hence one of the G'-versions of the two verbal languages $L_{q/2,q/2-1}$ and $L_{q/2+1,q/2}$ cannot be regular. ∎

Remark 27: One can use the same argumentation as in Theorem 4 in order to show that the result of Theorem 26 holds if one uses other generators than a, b, c, d; hence there is no Thurston automatic presentation of G where all verbal sets are regular.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for the careful proofreading and useful comments. 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". Sanjay Jain was supported in part by NUS grants R252-000-420-112 and C252-000-087-001. Frank Stephan was supported in part by NUS grant R252-000-420-112.

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] 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.

[4] 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.

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

[6] Olga Kharlampovich, Bakhadyr Khoussainov and Alexei Miasnikov. *From automatic structures to automatic groups.* Technical Report, 2011. http://arxiv.org/abs/1107.3645

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

[8] Olga Kharlampovich and Alexei Myasnikov. *Definable subsets in a hyperbolic group.* Technical Report, 2012. http://arxiv.org/abs/1111.0577

[9] Gennady S. Makanin. Equations in a free group. *Izvestiya Akademii Nauk SSSR Seriya Matematicheskaya* 46(6):1199–1273, 1982.

[10] Alexei Myasnikov and Vitaly Romankov. *On rationality of verbal subsets in a group.* Technical report, 2011. http://arxiv.org/abs/1103.4817

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

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

[13] 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.