G-AUTOMATA, COUNTER LANGUAGES AND THE CHOMSKY HIERARCHY

MURRAY ELDER

ABSTRACT. We consider how the languages of G-automata compare with other formal language classes. We prove that if the word problem of G is accepted by a machine in the class $\mathcal{M}$ then the language of any G-automaton is in the class $\mathcal{M}$. It follows that the so called *counter languages* (languages of $\mathbb{Z}^n$-automata) are context-sensitive, and further that counter languages are indexed if and only if the word problem for $\mathbb{Z}^n$ is indexed.

1. INTRODUCTION

In this article we compare the languages of G-automata, which include the set of *counter languages*, with the formal language classes of context-sensitive, indexed, context-free and regular. We prove in Theorem 5 that if the word problem of G is accepted by a machine in the class $\mathcal{M}$ then the language of any G-automaton is in the class $\mathcal{M}$. It follows that the counter languages are context-sensitive. Moreover it follows that counter languages are indexed if and only if the word problem for $\mathbb{Z}^n$ is indexed.

The article is organized as follows. In Section 2 we define G-automata, linear-bounded automata, nested-stack, stack, and pushdown automata, and the word problem for a finitely generated group. In Section 3 we prove the main theorem, and give the corollary that counter languages are indexed if and only if the word problem for $\mathbb{Z}^n$ is indexed for all n.

2. DEFINITIONS

If G is a group with generating set $\mathcal{G}$, we say two words u, v are equal in the group, or $u =_G v$, if they represent the same group element. We say u and v are identical if the are equal in the free monoid, that is, they are equal in $\mathcal{G}^*$.

Definition 1 (Word problem)**.** The *word problem* of a group G with respect to a finite generating set $\mathcal{G}$ is the set of words $\{w \mid w =_G 1\}$. Note that

---

Date: February 2, 2008.

2000 *Mathematics Subject Classification.* 20F65, 20F10, 68Q45.

Key words and phrases. G-automaton; counter language; word problem for groups; Chomsky hierarchy.

Supported by EPSRC grant GR/S53503/01.

the word problem for a finitely generated group is a language over a finite alphabet.

Definition 2 (G-automaton)**.** Let G be a group and Σ a finite set. A (non-deterministic) G-*automaton* A_G over Σ is a finite directed graph with a distinguished *start vertex* q_0, some distinguished *accept vertices*, and with edges labeled by $(\Sigma^{\pm 1} \cup \{\epsilon\}) \times G$. If p is a path in A_G, the element of $(\Sigma^{\pm 1})$ which is the first component of the label of p is denoted by $w(p)$, and the element of G which is the second component of the label of p is denoted $g(p)$. If p is the empty path, $g(p)$ is the identity element of G and $w(p)$ is the empty word. A_G is said to *accept* a word $w \in (\Sigma^{\pm 1})$ if there is a path p from the start vertex to some accept vertex such that $w(p) = w$ and $g(p) =_G 1$.

Definition 3 (Counter language)**.** A language is k-*counter* if it is accepted by some $\mathbb{Z}^k$-automaton. We call the (standard) generators of $\mathbb{Z}^k$ *counters.* A language is *counter* if it is k-counter for some $k \geq 1$.

These definitions are due to Mitrana and Striebe [9]. Note that in these counter automata, the values of the counters is not accessible until the final accept/fail state. For this reason they are sometimes called *blind.* Elston and Ostheimer [1] proved that the word problem of G is deterministic counter with an extra "inverse" property if and only if G is virtually abelian. Recently Kambites [8] has shown that the inverse property restriction can be removed from this theorem.

It is easy to see that the word problem for G is accepted by a G-automaton.

Lemma 4. *The word problem for a finitely generated group G is accepted by a deterministic G-automaton.*

Proof: Construct a G-automaton with one state and a directed loop labeled by (g, g) for each generator g. The state is both start and accept. A word in the generators is accepted by this automaton if and only if it represents the identity, by definition. □

Recall the definitions of the formal language classes of recursively enumerable, decidable, context-sensitive, indexed, stack, context-free, and regular. Each of these can be defined as the languages of some type of restricted Turing machine as follows.

Consider a machine consisting of finite *alphabet* Σ, a finite *tape alphabet* Γ, a *finite state control* and an infinite *work tape*, which operates as follows. The finite state control is a finite graph with a specified *start node*, some specified *accept nodes*, and edges labeled by an alphabet letter and an instruction for the work tape. The instructions in general are are the form `read, write, move left, move right` and one reads/writes letters from Γ on the tape. The tape starts out blank.

One inputs a finite string in Σ^* one letter at a time, read from left to right. For each letter $x \in \Sigma$, the finite state control performs some instructions on the work tape corresponding to an edge who label is $(x, \texttt{instructions})$, and moves to the target node of the edge. One starts at the start node, and *accepts* the string if there is some path from the start node to an accept node labeled by the letters of the string. The *language* of a machine is the set of strings in Σ^* which the machine accepts. If the finite state control includes edges of the form $(\epsilon, \texttt{instructions})$ then one can work on the work tape without reading input, and if a machine has such edges, or two edges with the same letter in the first coordinate of the edge label, then such machines (and their languages) are called *non-deterministic*.

If we allow no further restrictions on how this machine operates, then we have a *Turing machine*. By placing increasingly strict restrictions on the machine, we obtain a hierarchy of languages corresponding to the machines. If the Turing machine halts on accepted strings then the language is *recursively enumerable* or *r.e.*, and if it halts both accepted and rejected strings the language is *decidable*. If we restrict the number of squares of tape that can be used to be a constant multiple of the length of the input string, then we obtain a *linear bounded automaton*, and the languages of these are called *context-sensitive*. If we make the tape act as a *nested stack* (see [4]) then the language of such a machine is called *indexed*.

If the tape is a stack (first in last out) where the pointer may read but not write on any square, then the machine and its languages are called *stack*. If the tape is a stack such that the pointer can only read the top square, we have a *pushdown automaton*, the languages of which are *context-free*. Finally, if we remove the tape altogether we are left with just the finite state control, which we call a *finite state automaton*, languages of which are *regular*.

For more precise definitions see [4] for nested stack automata, [5] for stack, [12] for regular and context-free and [7] for these plus linear bounded automata. Two good survey articles are [11] and [3].

3. Main theorem

Theorem 5. *Let $\mathcal{M}$ be a formal language class: (regular, context-free, stack, indexed, context-sensitive, decidable, r.e.) and let G be a finitely generated group. The word problem for G is in $\mathcal{M}$ if and only if the language of every G-automaton is in $\mathcal{M}$.*

Proof: By Lemma 4 the word problem of G is accepted by a G-automaton, so one direction is done.

Let L be a language over an alphabet Σ accepted by a G-automaton P. Fix a finite generating set $\mathcal{G}$ for G which includes all elements of G that are the second coordinate of an edge label in P. Let N be an $\mathcal{M}$-automaton which accepts the word problem for G with respect to this generating set.

Construct a machine M from the class $\mathcal{M}$ to accept L as follows. The states of M are of the form (p_i, q_j) where p_i is a state of P and q_i is a state of N. The start state is (p_S, q_S), and accept states are (p_A, q_A) where p_S, p_A, q_S, q_A are start and accept states in P and N.

The transitions are defined as follows. For each edge (x, g) in P from p to p', and for each edge $(g, \texttt{instruction})$ in N from q to q', add an edge from (p, q) to (p', q') in M labeled $(x, \texttt{instruction})$.

Then M accepts a string in Σ^* if there is a path in M corresponding to paths in P from the start state to an accept state such that the labels of the second coordinate of the edges give a word w in $\mathcal{G}^*$, and a path in N from a start state to an accept state labeled by w which evaluates to the identity of G and respects the tape instructions.

Note that M is deterministic if both N and P are deterministic and no state in P has two outgoing edges with the same group element in the first coordinate of the edge labels. □

It is easy to see that if G is a finite group, then the construction describes a finite state automaton (we don't need any tape). Herbst [6] showed that the word problem for a group is regular if and only if the group is finite. If G is virtually free then Muller and Schupp [10] proved that G has a context-free word problem, so the language of every G-automaton for G virtually free is context-free.

Since the word problem for G virtually abelian is context-sensitive, we get the following corollary.

Corollary 6. *Counter languages are context-sensitive.*

More generally, Gersten, Holt and Riley [2, Corollary B.2] have shown that finitely generated nilpotent groups of class c have context-sensitive word problems, so the language of every G-automaton for G finitely generated nilpotent is context-sensitive.

Figure 1 shows how counter languages fit into the hierarchy.

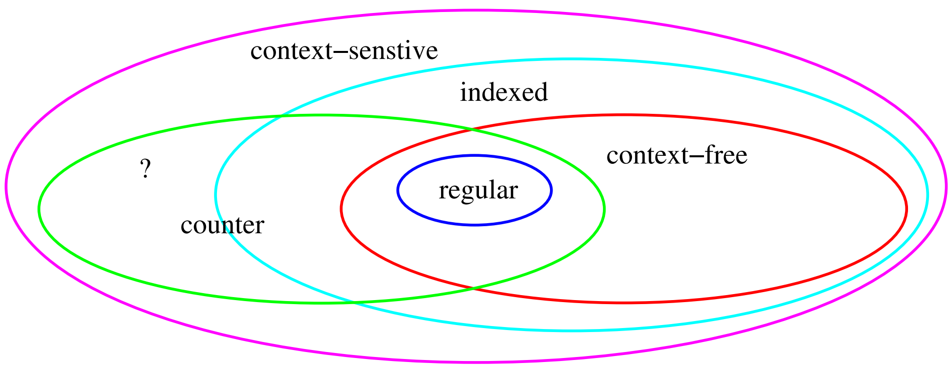

FIGURE 1. How counter languages fit into the hierarchy

Corollary 7. *The word problem for $\mathbb{Z}^n$ is indexed for all $n \geq 1$ if and only if every counter language is indexed.*

Proof: By Theorem 5 if the word problem for $\mathbb{Z}^n$ is indexed then the language of every $\mathbb{Z}^n$-automaton is indexed, proving one direction.

Since the word problem of $\mathbb{Z}^n$ is an n-counter language then if it is not indexed then not every counter language is indexed. □

By Gilman and Shapiro, if the word problem of G is accepted by a nested stack automaton with some extra restrictions, then it is virtually free. We still do not know whether word problem of $\mathbb{Z}^n$ is accepted by a nested stack automaton without extra restrictions.

Conjecture 8. The word problem for $\mathbb{Z}^2$ is not indexed.

To motivate this conjecture and suggest a possible proof strategy, define $L_n(y, z)$ to be the set of words in $\{y, z\}^*$ with n ys and n zs, and consider the language $L = \{x^n w_n \mid n \in \mathbb{N}, w_n \in L_n(y, z)\}$. This language is the word problem $\mathbb{Z}^2$ with respect to the generators a, b, a^{-1}, b^{-1}, intersected with the regular language $\{(ab)^n w_n \mid w_n \in L_n(a^{-1}, b^{-1})\}$. So if one could prove L is not indexed then one would prove the conjecture.

4. Acknowledgments

Thanks to Ray Cho, Andrew Fish, Bob Gilman, Claas Rover, Sarah Rees and Gretchen Ostheimer for many fruitful discussions about counter, context-sensitive and indexed languages and word problems.

References

[1] Gillian Elston and Gretchen Ostheimer. On groups whose word problem is solved by a counter automaton. *Theoret. Comput. Sci.*, 320 175–185, 2004.
[2] Steve Gersten, Derek Holt and Tim Riley. *Geometric And Functional Analysis* 13, 795-814, 2003.
[3] Robert Gilman. *Formal languages and infinite groups*, Geometric and computational perspectives on infinite groups. DIMACS Ser. Discrete Math. Theoret. Comput. Sci.(25) 1996
[4] Robert Gilman and Michael Shapiro. *On groups whose word problem is solved by a nested stack automaton.* arXiv: math.GR/9812028.
[5] Derek Holt and Claas Rover. Groups with indexed co-word problem. Submitted.
[6] Thomas Herbst. On subclasses of context-free groups. *Theoret. Informatics Appl.* 25, 255–272, 1991.
[7] John Hopcroft and Jeffery Ullman. *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.
[8] Mark Kambites. *Word problems recognizable by deterministic blind monoid automata.* arXiv: math.GR/0506137.
[9] Victor Mitrana and Ralf Stiebe. The accepting power of finite automata over groups, in *New trends in formal languages*, Lecture Notes in Comput. Sci. 1218, 39–48, 1997.
[10] David Muller and Paul Schupp. Groups, the theory of ends and context-free languages. *J. Comput. System Sci.*, 26, 295–310, 1983.
[11] Sarah Rees. Hairdressing in groups: a survey of combings and formal languages. *Geometry and Topology Monographs* 1, 493–509, 1998.
[12] Michael Sipser. *Introduction to the Theory of Computation*, PWS Publishing Co., 1997.

School of Mathematics and Statistics, University of St Andrews, KY16 9SS Scotland
E-mail address: murrayelder@gmail.com, http://me.id.au/