Decision Problems in Invertible Automata

Evan Bergeron & Klaus Sutner

May 5, 2017

Abstract

We consider a variety of decision problems in groups and semigroups induced by invertible Mealy machines. Notably, we present proof that, in the Abelian case, the automorphism membership problem is decidable in these semigroups. In addition, we prove the undecidability of a Knapsack variant. A discussion of iteration and orbit rationality follows.

Contents

1 Introduction

The word problem is a classic group-theoretic decision problem. Given a finitely generated group G, and a word w over the generators (and their inverses), the word problem asks "is $w \in G$." The word problem is known to be undecidable in surprisingly small classes of groups - see [2] and [3] for background.

The invertible Mealy machines we consider here give rise to a class of semigroups (and sometimes groups) for which the word problem is decidable. The computability picture here is rather nuanced, however. Similarly important decision problems, among them the conjugacy problem, and the isomorphism problem are known to be undecidable - see [20] and TODO for details.

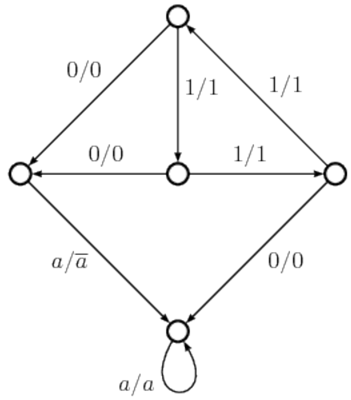


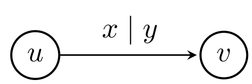
Figure 1: Grigorchuk's 5 state machine

In this paper, we present proof that, for the Abelian case, automorphism membership testing is decidable in this class of semigroups.

Serre first suggested the study of subgroups of the full automorphism group Aut 2^* of the infinite binary tree $\mathbf{2}^*$ in [18]. This notion has been usefully applied across group theory; a classic result here is Grigorchuk's group of intermediate growth, generated by the 5 state invertible machine shown in figure 1.

2 Background

An *automaton* is formally a triple (Q, Σ, δ), where Q is some finite state set, Σ is a finite alphabet of *symbols*, and δ is a transformation on $Q \times \Sigma$. Automata are typically viewed as directed graphs with vertex set Q and an edge labeled $x \mid y$ between u, v if $(u, x)\delta = (v, y)$.



One interprets this as if $\mathcal{A}$ is in state u and reads symbol x, then $\mathcal{A}$ transitions to state v and outputs symbol y. A computation within $\mathcal{A}$ may then start at some state q_0, and on input $\alpha_0\alpha_1 \dots \alpha_k$, output $\beta_0\beta_1 \dots \beta_k$, where $(q_i, \beta_i) = (q_{i-1}, \alpha_i)\delta$ for all $i = 0 \dots k$.

As in the above case, where δ outputs exactly one character for every transition, we call the automaton $\mathcal{A}$ *alphabetic*. An automaton is called *invertible* when every state in Q has some bijection

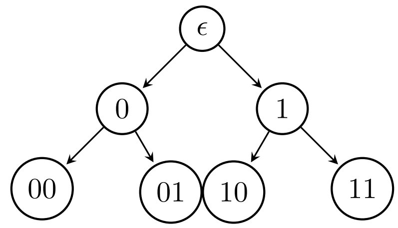


Figure 2: $\mathbf{2}^*$ interpreted as the infinite binary tree

π on Σ such that $(u, x)\delta = (v, \pi(x))$. A state in $\mathcal{A}$ is a *copy state* if π is the identity permutation and is a *toggle state* otherwise. The present paper is concerned only with invertible automata.

2.1 Actions on the infinite tree

We may identify the set Σ^* with an infinite, regular tree of degree $|\Sigma|$. The root is labelled with the empty string ϵ, and a vertex labelled w has the child wa for each $a \in \Sigma$. Out of convenience, we will frequently conflate a vertex with its label.

Each state $q \in Q$ acts on the corresponding tree, sending vertex w to wq. Moreover, if $\alpha\alpha' q = \beta\beta'$, then $\alpha q = \beta$, for any $\alpha, \alpha', \beta, \beta' \in \Sigma^*$. Which is to say, q's action on the tree is an adjacency-preserving map and is thus an endomorphism on the tree. Additionally, q is length-preserving, and thus preserves levels of the tree (and thus is an automorphism of the tree).

We extend the action of Q on Σ^* to words $q = q_1 \ldots q_n$ over Q^+ by

$$wq = (\ldots((wq_1)q_2)\ldots q_n).$$

This computation corresponds with running $\mathcal{A}$ starting at state q_1, then taking that output and running it through the machine starting at state q_2, and so on. We adopt the convention of applying functions from the right here. In this way, function composition corresponds naturally with string concatenation.

So there is a natural homomorphism $\phi : Q^+ \to \operatorname{Aut} \mathbf{2}^*$, where $\operatorname{Aut} \mathbf{2}^*$ denotes the semigroup of automorphisms of the tree $\mathbf{2}^*$. We denote the image of ϕ by $\Sigma(\mathcal{A})$.

Semigroup theory

A *semigroup* is a set S paired with a binary operation $f : S \times S \to S$ such that S is closed under f and f is associative over S. Any set of endofunctions forms a semigroup under composition.

A semigroup is called *Abelian* when its corresponding binary operation is commutative.

For an automaton $\mathcal{A}$, we denote by $S(\mathcal{A})$ the semigroup generated by Q under composition. $\mathcal{A}$ is said to be *commutative* or *Abelian* when $S(\mathcal{A})$ is Abelian. We write $G(\mathcal{A})$ for the group generated by the elements of Q and their inverses.

One may also speak about $S(\mathcal{A})$ and $G(\mathcal{A})$ without explicit reference to an automaton $\mathcal{A}$, yielding the corresponding definition:

Definition 1. *We call a semigroup S an* automaton semigroup *if there is some automaton $\mathcal{A}$ with $S \simeq \Sigma(\mathcal{A})$. Similarly, a group G is called an* automaton group *if $G \simeq \Sigma(\mathcal{A})$ for some automaton $\mathcal{A}$.*

Wreath Recursions

Any automorphism f of Σ^* can be written in the recursive form:

$$f = (f_{\alpha_1}, f_{\alpha_2}, \ldots, f_{\alpha_n})\tau$$

where $n = |\Sigma|$ and each f_α is an automorphism of a subtree of the root. Here, τ is some permutation on Σ. In the case where $\Sigma = \{0, 1\}$, we have $f = (f_0, f_1)\sigma$ where σ denotes transposition. If $f = (f_0, f_1)\sigma$, f is said to be *odd.* If $f = (f_0, f_1)$, f is said to be *even.* That is to say, automorphisms may be classified as even or odd depending on their action on the first level of the tree.

The set of even automorphisms form a subgroup H of index 2 in $G(\mathcal{A})$. Moreover, the residuation maps are group homomorphisms when restricted to H.

The automorphism semigroup of Σ^* decomposes into a recursive wreath product

$$\mathrm{Aut}\,\mathbf{2}^* = \mathrm{Aut}\,\mathbf{2}^* \wr \tau_\Sigma$$

where τ_Σ is the tranformation semigroup on Σ. Which is to say,

$$\operatorname{Aut}\mathbf{2}^* = \underbrace{(\operatorname{Aut}\mathbf{2}^* \times \ldots \times \operatorname{Aut}\mathbf{2}^*)}_{n \text{ times}} \rtimes \tau_\Sigma.$$

Definition 2. *Define* residuation maps $\partial_a : S(\mathcal{A}) \to S(\mathcal{A})$ *that map* $f = (f_0, f_1)\sigma$ *to* f_a*, and a* parity map par *such that* $\mathsf{par}(f) = \sigma$*.*

Note that a subgroup G of $\operatorname{Aut}(2^*)$ need not be closed under residuation; if it is, we call it *self-similar* or *state-closed*. In this case, the wreath characterization in the full automorphism group carriers over and we have $G \cong (G \times G) \rtimes \tau_{\mathbf{2}}$.

3 Decision Problems

Automaton semigroups exhibit many interesting and nuanced computability properties. While it is an easy result that the WORD PROBLEM is solvable in such semigroups, similar group-theoretic problems such as the CONJUGACY PROBLEM and FINITENESS PROBLEM have been shown to be undecidable (see [20], and [6], respectively).

Various other semigroup theoretic decision problems have recently been considered for small classes of semigroups by Cain in [3]. We consider a subset of his distinguished properties in the automaton semigroup case here.

3.1 IsAbelian is polynomial time

For a binary invertible automaton $\mathcal{A}$, define the *gap* of an automorphism $f \in G(\mathcal{A})$ to be $\gamma_f = (\partial_0 f)(\partial_1 f)^{-1}$.

The following result is adapted from [16].

Lemma 1. *$\mathcal{A}$ is Abelian if and only if all even automorphisms in $S(\mathcal{A})$ have gap I and odd automorphisms have constant gap.*

Proof. Suppose $\mathcal{A}$ is Abelian; so $fg = gf$ for all f, g in $S(\mathcal{A})$. If f and g are both odd, simply residuate both sides to get

$$(\partial_a f)(\partial_{\bar{a}} g) = \partial_a(fg) = (\partial_a gf) = (\partial_a g)(\partial_{\bar{a}} f)$$

which yields $\gamma_f = \gamma_g$. If f is even and g odd, without loss of generality, we have

$$(\partial_0 f)(\partial_0 g) = \partial_0(fg) = \partial_0(gf) = (\partial_0 g)(\partial_1 f)$$

which, with algebraic manipulation, yields $\gamma_f = I$.

Conversely, first suppose f and g are both odd. Then $fg = (\partial_0 f \partial_1 g, \partial_1 f \partial_0 g)$ and $gf = (\partial_0 g \partial_1 f, \partial_1 g \partial_0 f)$. Since $\gamma_f = \gamma_g$, these wreath recursions are the same. If f is even and g odd, $fg = (\partial_0 f \partial_0 g, \partial_1 f \partial_1 g)\sigma$ and $gf = (\partial_0 g \partial_1 f, \partial_1 g \partial_r f)\sigma$.

If f and g are both even, the claim follows by induction. $\square$

Definition 3. *For an automaton $\mathcal{A}$ with states $s_1 \dots s_n$, the* inverse automaton *of $\mathcal{A}$, denoted $\mathcal{A}^{-1}$, has state set $t_1, \dots t_n$ and transitions $\partial_a t_i = \partial_{\bar{a}} s_i$, with t_i a toggle state if and only if s_i is as well.*

It is easy to verify by induction that $t_i = s_i^{-1}$ for all i.

Definition 4. *For an automaton $\mathcal{A} = (Q, \Sigma, \delta)$, the* acceptor of $\mathcal{A}$ at t, *denoted $\mathcal{A}(t)$, is a partial DFA with state set Q, input alphabet $Q \times Q$, and transitions $s \xrightarrow{a \times b} s'$ for each transition $t \xrightarrow{a|b} t'$ in $\mathcal{A}$. Every state is accepting.*

Lemma 2. *The language of the acceptor $\mathcal{A}(t)$ is*

$$\{(x_1, y_1)(x_2, y_2), \dots, (x_n, y_n) \mid y_1 \dots y_n = (x_1, \dots x_n)t\}$$

Proof. By induction on the length of the input string. $\square$

Definition 5. *For an automaton $\mathcal{A} = (Q, \Sigma, \delta)$, the* product automaton $\mathcal{A} \times \mathcal{A}$ *is a machine with state set $Q \times Q$ and transition function defined by $\partial_a(s_1, s_2) = (\partial_a s_1, \partial_{as_1} s_2)$.*

We can see by induction that each state (s_1, s_2) in the product automaton corresponds to the word $s_1 s_2 \in S(\mathcal{A})$.

Theorem 1. *There is a polynomial time algorithm to check if an automaton $\mathcal{A}$ is Abelian.*

Proof. On input automaton $\mathcal{A}$, build the inverse automaton $\mathcal{A}^{-1}$. Construct the product automaton $\mathcal{A} \times \mathcal{A}^{-1}$. Then for each toggle state t_i of $\mathcal{A}$, for the state $s_i = (\partial_1 t_i, \partial_1 t_i^{-1})$ in $\mathcal{A} \times \mathcal{A}^{-1}$, construct the acceptor DFA $(\mathcal{A} \times \mathcal{A}^{-1})(s_i)$. Verify all the constructed DFAs are equivalent. $\square$

The reader may be interested to note that this product automaton construction also provides proof that the word problem for automaton semigroups is decidable.

3.2 Automorphism Membership

This section considers the subsemigroup $S(\mathcal{A})$ of Aut $\mathbf{2}^*$ generated by the associated autormophisms of an invertible binary transducer. We assume minimality throughout this section.

We provide proof that the automorphism membership question is decidable in the Abelian case, and discuss partial work toward the general case. Some necessary background from [15] is outlined below.

3.2.1 Linear algebraic background

Theorem 2. *If $\mathcal{A}$ is Abelian, then $G(\mathcal{A})$ is isomorphic to either a finite Boolean group or to $\mathbb{Z}^m$ for some $m \geq 1$. In the latter case, there is an isomorphism $\phi : G(\mathcal{A}) \to \mathbb{Z}^m$ satisfying the following recursion*

$$\phi^{-1}(v) = \begin{cases} (\phi^{-1}(A \cdot v), \phi^{-1}(A \cdot v)) & \textit{if } \phi^{-1} \textit{ is even} \\ (\phi^{-1}(A \cdot v - r), \phi^{-1}(A \cdot v + r)) & \textit{otherwise} \end{cases}$$

where $A \in GL(m, \mathbb{Q})$ and $v \in \mathbb{Q}^m$. Additionally, for all $v \in \mathbb{Z}^m$, $A \cdot v \in \mathbb{Z}^m$ or $A \cdot v \pm r \in \mathbb{Z}^m$.

We call the matrix A above the *residual matrix* of A. The vector r is referred to as the *residual vector*. Put differently, this theorem specifies that when $\mathcal{A}$ is Abelian, residuation is an affine map.

We have the following properties of A:

Theorem 3. *If $G(A) \cong \mathbb{Z}^m$ and A is its associated residual matrix, A satisfies the following properties:*

1. *A is contracting; its spectral radius is less than 1*

2. *A is 1/2-integral, meaning that A^{-1} is a subgroup of index 2 in $\mathbb{Z}^m$. Therefore A be represented as*

$$\begin{bmatrix} \frac{a_{1,1}}{2} & a_{1,2} & \cdots & a_{1,m} \\ \frac{a_{2,1}}{2} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{m,1}}{2} & a_{m,2} & \cdots & a_{m,m} \end{bmatrix}$$

where all $a_{i,j}$ are integers.

3. *The characteristic polynomial $\chi_A(x)$ is irreducible over $\mathbb{Q}$ and has the form*

$$\chi_A(x) = x^m + \frac{1}{2}g(x)$$

for some $g \in \mathbb{Z}[x]$ of degree $m-1$. In particular, the constant term is $+-\frac{1}{2}$.

4. *A is invertible and the characteristic polynomial $\chi_{A^{-1}}(x)$ is integral and irreducible over $\mathbb{Q}$. From property 2, Laplace expansion yields that A^{-1} is an integral matrix that is similar to the companion matrix of $\chi_{A^{-1}}(x)$ over $\mathbb{Q}$.*

The Latimer and MacDuffee theorem states that:

Theorem 4. *If $p(x) \in \mathbb{Z}[x]$ is monic and irreducible, the $GL(m,\mathbb{Z})$ similarity classes of integral matrices whose characteristic polynomial coincides with $p(x)$ is in one-to-one correspondance with ideal classes of the ring $\mathbb{Z}[\theta]$, where θ is any root of $p(x)$.*

Property 1 of theorem 3 provides a bound on the coefficients of $\chi_A(x)$. When combined with property 4 and the above theorem, it can be shown that, for fixed m, there exist only finitely many possibilities of A, up to $GL(m,\mathbb{Z})$ similarity.

Definition 6. *Take G to be some self-similar group. We may construct the* complete group automaton *(occasionally abbreviated as the* complete automaton*) for G, written $\mathcal{C}_G$, as follows: the automaton has G's carrier set as state set with transitions $f \xrightarrow{a|af} \partial_a f$.*

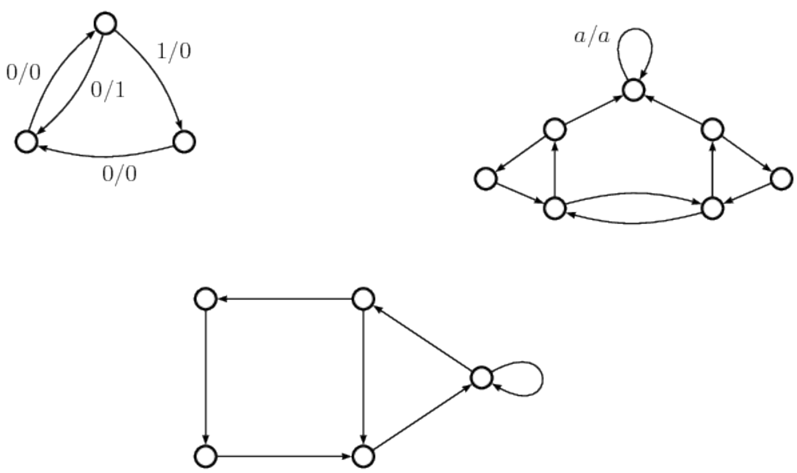
Of course in general, this invertible automaton will be infinite, but certainly $\mathcal{S}(\mathcal{C}_G)$ is a group and ismorphic to G. The more interesting case is when G may be represented in terms of a finite automaton. Toward this end, call G *finite-state* if for all $f \in G$, the number of residuals $\partial_w f$ is finite. If G is self-similar, finite-state, and finitely generated, we can construct the *group automaton* $\mathcal{A}_G$, a binary Mealy automaton, just like the complete group automaton, but with state set restricted to the collection of all residuals of the generators of G. Of course, the group generated by $\mathcal{A}_G$ is isormorphic to G. One need to be careful, however; the semigroup may be different. Pleasantly, $\mathcal{A}_G$ is minimal by construction.

Lemma 3. *Any complete automaton defined by (A, r) has only finitely many subautomata, each of which have finitely many states.*

Example 1. *The following automata represent the subautomata of the complete automaton generated with residual matrix*

$$A = \begin{bmatrix} -1 & 1 \\ -\frac{1}{2} & 0 \end{bmatrix}$$

and residual vector $r = [-1, -\frac{3}{2}]^T$.



3.2.2 Membership is decidable in the Abelian case

Definition 7. *For a residual matrix A and residual vector $r = A \cdot e_1$, the* principal automaton *is the automaton generated from closure of e_1 under residuation defined by the pair (A, r).*

Given an invertible automaton $\mathcal{A}$ and a principal Abelian automaton $\mathcal{B}$, we determine if $f = \mathcal{A}(p)$ is in the semigroup generated by $\mathcal{B}$.

Thus one needs to check if there is some product automaton

$$D = \mathcal{B}_{p_1} \times \mathcal{B}_{p_2} \times \ldots \times \mathcal{B}_{p_n}$$

that implements f. We have no computable bound on n, so a priori this merely semidecidable.

Now consider the complete automaton C for $\mathcal{B}$, and let g be the automorphism defined by D. After minimization, D produces a subautomaton of C tat consists of a "transient part" and a copy of $\mathcal{B}$ (there may be SCCs in the transient part, but they are not subautomata). Hence, there is some word w such that $\partial_w g$ is just a single state in the copy of $\mathcal{B}$; also, w can be found effectively[1]. In fact, for all u, there is a w such that $\partial_{ww} g$ is atomic. Essentially this just means that g is strongly tame.

We may safely assume that $\mathcal{A}$ is minimal. Then is looks like $\mathcal{A}$ has to have $\mathcal{B}$ as a subautomaton to satisfy this ultimate atomicity condition, plus a transient part sitting on of $\mathcal{B}$. It should be decidable if things match up.

If $\mathcal{B}$ is not principal, there are multiple subautomata of C to content with, but that should not make a major difference. Ditto if $\mathcal{B}$ is just a random subautomaton of C.

3.2.3 Membership is open in the general case

The decidablity of MEMBERSHIP is arguably the most important open problem relevant to this thesis. Its decidablity would imply the decidability of IsGroup, as one could simply check that the inverse of each generator is contained in $\mathcal{S}(\mathcal{A})$.

3.3 IsGroup

3.3.1 IsGroup is decidable in the Abelian case

3.3.2 IsGroup is open in the general case

3.4 Knapsack is undecidable for automaton semigroups

We follow a proof strategy similar to [11].

Definition 8. *We define the* KNAPSACK PROBLEM *as follows: given as input generators $g_1 \dots g_k$ and a target semigroup element g, do there exist natural numbers $a_1 \dots a_k$ such that*

$$g_1^{a_1} \cdots g_k^{a_K} = g$$

[1]TODO

Definition 9. *The* Generalized Knapsack Problem *has as input generators* $g_1 \dots g_k, h_1, \dots h_l$*, and has as output whether there exist natural numbers* $a_1 \dots a_k, b_1, \dots, b_l$ *such that*

$$g_1^{a_1} \cdots g_k^{a_k} = h_1^{b_1} \cdots h_l^{b_l}$$

We demonstrate that the Generalized Knapsack Problem is undecidable in the class of automaton semigroups by reducing from Hilbert's tenth problem. The undecidablity of the Knapsack Problem easily follows.

Definition 10. *We define the decision problem* Hilbert *as following: "given a polynomial over the integers and an integer* a*, do there exist values of the arguments to the polynomial such that the polynomial evaluated at this point is equal to* a*?"*

It is well-known that there exist polynomials for which Hilbert is undecidable, see [12] for details.

Theorem 5. Knapsack *is undecidable in the class of automaton semigroups.*

Proof. Recall that the Heisenberg semigroup

$$H_3(\mathbb{N}) = \left\{ \begin{bmatrix} 1 & a & c \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} ; a, b, c \in \mathbb{N} \right\}$$

is an automaton semigroup [9]. Moreover, the class of automaton semigroups is closed under direct products, proven by Cain in [2]. We denote elements of $H_3(\mathbb{N})$ as

$$H_{x,y,z} = \begin{bmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

Proposition 1. *There exist fixed constants* $d, e \in \mathbb{N}$ *and a exponential expression* E *of the form*

$$s_1^{x_1} s_2^{x_2} \cdots s_n^{x_n}$$

with $s_i \in G = H_3(\mathbb{N}) \times \mathbb{N}^e$ for which the Knapsack *problem is undecidable.*

Proof. By introducing extra variables, we can construct from the polynomial $P(x_1, \dots, x+n)$ a system S of equations of the form $x \cdot y = z$, $x + y = z$, $x = c$ (for $c \in \mathbb{Z}$) such that the equation $P(x_1, \dots, x_n) = a$ has a solution in $\mathbb{N}$ if and only if the system of equations $S_a = S \cup \{x_0 = a\}$ has a solution in $\mathbb{N}$. Let X be the set of variables that occur in S_a.

Take a natural number a (the input of the reduction). Assume that S_a contains d equations of the form $x \cdot y$, and e many equations of the form $x + y = z$ or $x = c$. We enumerate these equations as $E_1, \dots E_{d+e}$, where the first d equations are all the muliplicative equations. Then set $G_i = H_3(\mathbb{N})$ for each $i \leq d$ and set $G_i = \mathbb{N}$ for each $i > d$. For every i, we define an element g_i and an exponential expression E_i over G_i as follows:

Case 1: $E_i = (x \cdot y = z)$. Thus we have $G_i = H_3(\mathbb{N})$. Set g_i to be the identity matrix in $H_3(\mathbb{N})$ and consider the following equation:

$$H_{1,0,0}^{x} = \begin{bmatrix} 1 & x & xy \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} = H_{1,0,0}^{z} H_{0,0,1}^{y} H_{1,0,0}^{x}$$

This equation has a solution if and only if $xy = z$.

Case 2: $E_i = (x + y = z)$ and so $G_i = \mathbb{N}$. Set $g_i = 0$ TODO

Case 3: $E_i = (x = c)$. (This includes our distinguished equation $x_0 = a$). We have $G_i = \mathbb{N}$. Then set $g_i = c$ and $E_i = x$. Then TODO

Finally, define E to be the direct product of the E_i's, $\prod_{i=1}^{d} E_i$. Then TODO. □

□

3.5 A monoid with decidable Word Problem and undecidable IsGroup

We establish the existence of a monoid with decidable Word Problem, but undecidable IsGroup. We may take this result as an intermediate step toward the decidability of the IsGroup problem for automaton semigroups.

Preliminaries

Here we take a *Turing machine* to be a 6-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where Q, Σ, Γ are all finite sets. $\delta : Q \times \Sigma \to Q \times \Gamma \times \{L, S, R\}$ is the transition function, Q is the state set, Σ is the input alphabet, $\Gamma \supseteq \Sigma$ is the tape alphabet, b is some blank symbol, with $b \in \Gamma - \Sigma$, q_{accept} is the unique accepting final state, and q_{reject} the single rejecting final state.

We further define a Turing machine *configuration* to be a triple $(u, q, v) \in \Gamma^* \times Q \times \Gamma^*$. Here, u denotes the tape contents to the left of the tapehead, q is the current state, and v begins at the tapehead and extends to the right.

A configuation C for a TM M is said to *yield* configuration C' if M can step directly from C to C'.

For a Turing machine M, take C_M to be the set of all valid configurations of M. Then define CG_M to be the graph (C_M, E), where $(u, v) \in E$ if and only if u yields v in M.

Definition 11. *Define the* canonical computation of M on w.

$$\mathsf{canon}(M, w) : \mathcal{T} \times \Sigma^* \to C_M^* \cup C_M^\omega$$

to be the function that maps input w to the sequence of configurations M takes on while computing over w. Note that $\mathsf{canon}(M, w)$ will be a finite sequence if and only if M halts on w.

Certainly, not every configuration in C_M will be along the sequence $\mathsf{canon}(M, w)$. Which is to say, there are unreachable configurations.

Informally, a *self-verifying Turing machine S* is one that, at every step, verifies that the current configuration lies upon the canonical computation. If S finds that this is not the case, S immediately rejects. Otherwise, the computation steps forward a single step.

In the configuration graph CG_S, there is a path extending from each valid starting configuration (ϵ, q_0, w) for $w \in \Sigma^*$. The remaining states form an infinite star graph with q_{reject} as the center.

Proposition 2. *There is a computable*[2] *function* sv *that maps Turing machines to equivalent self-verifying Turing machines.*

[2]The reader may interested to find that sv is in fact primitive recursive - see TODO for details.

Speaking informally, as the canonical computation proceeds, a program counter is kept - perhaps to the left of the tapehead. After every step, the Turing machine will examine what "time step" the computation is currently sitting in. It will perform the canonical computation for the first n steps. If it does not wind up where it's configuration says it is, it transitions to the death state. Otherwise, it continues.

In the interest of reader intuition, we expound upon a couple of implementation details here. For further reading, see [4], [5], and [19].

TODO rough sketch of implementation details.

The submonoid in question

Define the Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ to operate only on the blank tape; for all $s \in \Sigma$, $\delta(q, s) = (q_{reject}, b, S)$

Then take the ambient Abelian group $G_M = (C_M, \cdot)$ whose carrier set is all configurations of M. For c, c' in G_M, we have $c = c'$ if and only if c yields c'.

Proposition 3. *$G_{\mathsf{sv}(M)}$ has a decidable word problem.*

Proof. For every word w in $G_{\mathsf{sv}(M)}$, there exist nonnegative integers a, r such that $w = q_{accept}^a q_{reject}^r$. Further, we may compute a and b. Recall that $\mathsf{sv}(M)$ maintains a program counter p to the left of the input. So we may simply run M for the first p steps and then check for configuration equality.

So then given two words w_1, w_2 in $G_{\mathsf{sv}(M)}$, simply compute a_1, r_1, a_2, r_2. Then w is in $G_{\mathsf{sv}(M)}$ if and only if $a_1 = a_2$ and $r_1 = r_2$. $\square$

Proposition 4. *If s is the start configuration of the Turing machine, it is undecidable whether $\langle s \rangle$ is a group.*

Proof. It is well known that the following language is undecidable

$$\textsc{halts} = \{\langle M \rangle \mid \text{TM } M \text{ halts on } \epsilon\}$$

and so we reduce from $\textsc{halts}$. Given as input a TM M, we use an oracle for $\textsc{IsGroup}$ as follows: first, compute $\mathsf{sv}(M)$, and then consider $G_{\mathsf{sv}(M)}$. Let s be the starting configuration for M on ϵ.

If M halts, then the submonoid generated by s is the trivial group. If M hangs, then $\langle s \rangle$ is the free monoid of rank one. So then $\langle s \rangle$ is a group if and only if M halts. Since $\mathsf{sv}(M)$ and M are equivalent, we are done. $\square$

4 Open Questions

- All automaton semigroups are recursively presented. If these presentations are regular, or context-free, does that affect the soluability of these questions?
- Having a zero
- Isomorphism problem
- Bounded automata, etc

References

[1] L. Bartholdi and P. V. Silva. Groups defined by automata. *CoRR*, abs/1012.1531, 2010.

[2] A. J. Cain. Automaton semigroups. *TCS*, 410(47–49):5022–5038, 2009.

[3] A. J. Cain and V. Maltcev. Decision problems for finitely presented and one-relation semigroups and monoids. *International Journal of Algebra and Computation*, 19(6):747–770, 2009.

[4] M. Davis. A note on universal turing machines. *Annals of Mathematics studies*, 34:167–175, 1956.

[5] M. Davis. The definition of universal turing machine. *Proceedings of the American Mathematical Society*, 8:1125–1126, 1957.

[6] Pierre Gillibert. The finiteness problem for automaton semigroups is undecidable. *CoRR*, abs/1304.2295, 2013.

[7] T. Godin. Knapsack problem for automaton groups. *HAL*, 2016.

[8] John E. Hopcroft, Rajeev Motwani, Rotwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computability*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2000.

[9] R. Kravchenko I. Bondarenko. Finite-state self-similar actions of nilpotent groups. *Geometriae Dedicata*, 163(1):339–348, 2013.

[10] O. Kharlampovich, B. Khoussainov, and A. Miasnikov. From automatic structures to automatic groups. *ArXiv e-prints*, July 2011.

[11] D. Konig, M. Lohrey, and G. Zetzsche. Knapsack and subset sum problems in nilpotent, polycyclic, and co-context-free groups. *Contemporary Mathematics*, 2015.

[12] Yuri V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, Cambridge, MA, USA, 1993.

[13] Y. Muntyan. *Automata Groups*. PhD thesis, Texas A&M University, May 2009.

[14] V. Nekrashevych. *Self-Similar Groups*, volume 117 of *Math. Surveys and Monographs*. AMS, 2005.

[15] V. Nekrashevych and S. Sidki. *Automorphisms of the binary tree: state-closed subgroups and dynamics of 1/2-endomorphisms*. Cambridge University Press, 2004.

[16] T. Okano. *Invertible Binary Transducers and Automorphisms of the Binary Tree*. PhD thesis, Carnegie Mellon University, May 2015.

[17] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, New York, NY, USA, 2009.

[18] J.-P. Serre. *Arbres, Amalgames,* SL_2. Société Mathématique de France, Paris, France, 1977.

[19] J. C. Shepherdson. Machine configuration and word problems of given degree of unsolvability. *Z. f. Math. Logik u. Grundlagen d. Mathematik*, 11:149–175, 1965.

[20] Z. Sunic and E. Ventura. The conjugacy problem in automaton groups is not solvable. *Journal of Algebra*, 364(148–154), 2012.

[21] K. Sutner. Invertible transducers, iterations and coordinates. *TODO*, 2013.

[22] K. Sutner and K. Lewi. Iterating inverse binary transducers. *Journal of Automata, Languages, and Combinators*, 17(2–4):293–313, 2012.