Invertible Transductions and Iteration

Klaus Sutner

Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract. We study iterated transductions, where the basic transductions are given by a class of length-preserving invertible transducers over the binary alphabet. It is shown that in some cases the resulting orbit relation is rational and we determine the complexity of several natural computational problems associated with the iterated transductions.

1 Iterating Transductions

We are interested in the analysis of discrete structures of the form $\mathfrak{C} = \langle C, T \rangle$ where C is a rational set of words, in this context often referred to as the space of *configurations*, and T is a functional binary relation on C determined by a rational transducer. While first-order properties of $\mathfrak{C}$ such as reversibility or the existence of k-cycles are of considerable interest, higher order properties involving $T^\star$, the *iterate* of T, are critical for the understanding of the structures. From a computational perspective, iteration of even rather simple transductions T produces structures $\mathfrak{C}$ that are too complicated to admit a detailed analysis. To wit, the next-step function of a Turing machine is easily modeled as a rational transduction, so together with iteration we are dealing with a system that is potentially computationally universal and thus difficult to classify and understand. Perhaps the most surprising result along these lines is Cook's proof [3] of the universality of the elementary cellular automaton number 110, a transducer that is defined essentially by a ternary Boolean function. To the best of our knowledge, this is the first example of a universal system that was "discovered" rather than constructed, with the specific intent of producing universality.

The argument relies heavily on a version of Post tag systems, so-called cyclic tag systems, and demonstrates that these systems, which are known to be computationally universal, can indeed be simulated by the transducer, given initial conditions of sufficient complexity. In fact, the argument employs bi-infinite binary words to produce universality; more precisely, one needs to consider words of the form ${}^\omega uwv^\omega$ where u, w and v are finite, binary words. This type of configuration is quite natural: letting C be the collection of all such words, the first-order structure $\langle C, T \rangle$ is an elementary substructure of the full shift space $\langle \mathbf{2}^{\mathbb{Z}}, T \rangle$ of all bi-infinite words, where T is the transduction associated with the cellular automaton number 110, see [23]. Remnants of this hardness result persist at the level of finite words, evaluation of the transduction on finite words is shown to be $\mathbb{P}$-complete in [14].

H. Jürgensen and R. Reis (Eds.): DCFS 2013, LNCS 8031, pp. 18–29, 2013.

To avoid hardness results based on configurations of unbounded size it is natural to consider *length-preserving* transductions on ordinary finite words. For a length-preserving transduction, the question whether a configuration occurs in the orbit of another is naturally in PSPACE and indeed easily seen to be PSPACE-complete in general. It is NL-complete to determine whether a configuration has a predecessor, see [21]. Note, though, that questions about the behavior of length-preserving transductions may well become undecidable when configurations of arbitrary size are considered. For example, one can show that it is undecidable whether all orbits end in a fixed point, see [22]; more precisely, this problem is Π_1-complete. Similarly questions about the length of the limit cycle of a configuration are undecidable. For example, one cannot determine whether all configurations of size n will evolve to a limit cycle of size linear in n, see [21]. Incidentally, there is a close connection between length-preserving transductions and context-sensitive languages as shown in [13]. The reference shows that every context-sensitive language L can be written in the form $L = \Gamma^\star \tau^\star \rho_\Sigma$ where τ is a length-preserving transduction and $\rho_{\Sigma^\star}$ is the operation "intersection by $\Sigma^\star$" for suitable alphabet Γ and Σ.

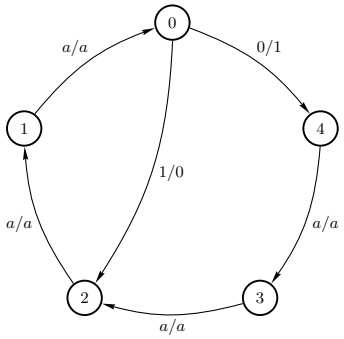
In this paper we further constrain transductions by insisting that they be length-preserving and invertible. Thus, $\mathfrak{C}$ consists of a collection of disjoint cycles, and each cycle contains only configurations of the same length: $T^\star \subseteq (\mathbf{2} \times \mathbf{2})^\star$. Correspondingly, the iterate $T^\star$ is an equivalence relation that we will refer to as the *orbit relation* of T. It follows that the first-order structure $\mathfrak{C}$ is an *automatic* permutation structure in the sense of [9,11]. Automatic structures have decidable first-order theories and natural decision algorithms can be given based on classical automata-theoretic methods. Note, though, that in practice only a small fragment of the first-order theory can be decided due to catastrophic growth in the state complexity of the associated machines.

To ensure invertibility, our transducers are special types of Mealy automata $\mathcal{A}$ where all transitions are of the form $p \xrightarrow{a/\pi(a)} q$; here $\pi = \pi_p$ is a permutation of the alphabet $\mathbf{2} = \{0, 1\}$ that depends on the source p of the transition. Thus there are two types of states: *toggle states* where π_p is the transposition and *copy states*, otherwise. By selecting any state p in $\mathcal{A}$ as the initial state we obtain a transduction $\mathcal{A}(p) : \mathbf{2}^\star \to \mathbf{2}^\star$. Write $\mathcal{S}(\mathcal{A})$ for the semigroup generated by the basic transductions $\mathcal{A}(p)$. These automata are called *binary invertible transducers* and have attracted a lot of attention in the group theory and dynamical systems community: the groups generated by all the transitions $\mathcal{A}(p)$ often have surprisingly interesting properties, see [1,6,7,15]. For example, the well-known lamplighter group can be described by a two-state invertible transducer and Grigorchuk has constructed a group of intermediate growth that can be interpreted as the transduction group of a binary invertible transducer on only 5 states and with a single toggle state.

In light of the complexity of the transduction groups associated with even rather small invertible transducers we further constrain our study to a special type of invertible transducers, so-called *cycle-cum-chord* transducers (or CCC transducers for short). These transducers have state set $\{0, 1, \ldots, n-1\}$ and transitions

$$p \xrightarrow{a/a} p-1, \quad p>0 \qquad \text{and} \qquad 0 \xrightarrow{0/1} n-1, \quad 0 \xrightarrow{1/0} m-1$$

where $1 \leq m \leq n$. We write $\mathfrak{A}_m^n$ for this transducer; the example $\mathfrak{A}_3^5$ is shown in figure 1. Note that $\mathfrak{A}_m^n$ contains but a single toggle state. It is shown in [25] that these transducers are fairly simple from the perspective of the associated semigroups: they are all free Abelian groups (except in the degenerate case $n = m$ when we obtain a finite Boolean group).



$$\underline{0} = (\underline{4}, \underline{2})\,\sigma$$
$$\underline{k} = (\underline{k^-}, \underline{k^-}) \quad 0 < k < 5$$

Fig. 1. The cycle-cum-chord transducer $\mathfrak{A}_3^5$ and its representation in the wreath form from section 2

There are several natural questions that are related to the analysis of the structures $\mathfrak{C}$ determined by our CCC transducers. First and foremost, there is the question of the complexity of the orbit relation associated with a transduction in $\mathcal{S}(\mathcal{A})$. In particular, there is the decision problem of determining when the orbit relation itself is rational, in which case the extended structure $\mathfrak{C}^\star = \langle C, T, T^\star \rangle$ is still automatic. More generally, we would like to understand the complexity of the *Orbit Problem*, the recognition problem of deciding $x f^\star y$ given two words x and y. A closely related question is the first canonical form problem: how hard is it to compute the length-lexicographical least element of an orbit, see [4,8]. Another elementary question is the *Iteration Problem*: what is the complexity of computing $x\, f^t$ for some transduction f, a word x and $t \geq 0$. We can strengthen the recognition problem and ask for a witness to membership in an orbit: given words x and y, find the least number $t \geq 0$ such that $x\, f^t = y$, or determine that no such t exists. We refer to this as the *Timestamp Problem*.

In this overview we will discuss results concerning these questions for cycle-cum-chord transducers. Though this class of machines is rather narrow, it turns out that a complete classification is currently out of reach. In section 2 we recap the basic definitions and describe the so-called *Knuth normal form* of a transduction, a central tool in the study of our transducers. We also comment on the rationality of orbits. In the next section, we discuss a natural coordinate system based on iterated transductions and describe the complexity of the timestamp and coordinate problems. Section 4 contains a rather lengthy list of open problems.

2 Transduction Groups and Knuth Normal Form

Our transductions are given by Mealy automata of the form $\mathcal{A} = \langle Q, \mathbf{2}, \delta, \lambda \rangle$ where Q is a finite set, $\mathbf{2} = \{0, 1\}$ is the input and output alphabet, $\delta : Q \times \mathbf{2} \to Q$ the transition function and $\lambda : Q \times \mathbf{2} \to \mathbf{2}$ the output function. As usual, we can think of $\mathbf{2}^\star$ as acting on Q via δ, see [2,18,10]. We are here only interested in *invertible transducers*: $\lambda(p, .) : \mathbf{2} \to \mathbf{2}$ is required to be a permutation for each state p. Write $\mathfrak{S}_2$ for the symmetric group on two letters and let σ be the transposition in this group. We refer to p as a *toggle state* when $\lambda(p, .) : \mathbf{2} \to \mathbf{2}$ is σ, and as a *copy state*, otherwise. By selecting an arbitrary state p as initial state we obtain a transduction $\mathcal{A}(p) : \mathbf{2}^\star \to \mathbf{2}^\star$. It is clear from the definitions that these maps are length-preserving bijections. To lighten notation we write $\underline{p}$ for this transduction whenever the automaton is clear from context. Lastly, $\mathcal{S}(\mathcal{A})$ denotes the semigroup generated by all the functions $\mathcal{A}(p)$ as p ranges over Q. For the CCC transducers from above, $\mathcal{S}(\mathcal{A})$ is already a group, as we shall see shortly. Given a transduction $f \in \mathcal{S}(\mathcal{A})$ we obtain a relational structure as in the introduction by letting T be its graph: $x\ T\ y$ iff $x\, f = y$. For technical reasons, this is preferable to dealing with functions directly. Note that if we interpret $\mathcal{A}$ as an acceptor over the alphabet $\mathbf{2} \times \mathbf{2}$ it recognizes T.

Following ideas from [19], we can think of $\mathbf{2}^\star$ as an infinite, complete binary tree. Our transductions naturally act as automorphisms of this tree, see [15,20]. Thus our groups $\mathcal{S}(\mathcal{A})$ can be viewed as subgroups of the ambient group $\mathsf{Aut}(\mathbf{2}^\star)$ of all automorphisms of $\mathbf{2}^\star$. In this setting, it is natural to write any automorphism f of $\mathbf{2}^\star$ as $f = (f_0, f_1)s$ where $s \in \mathfrak{S}_2$: s describes the action of f on the first level $\mathbf{2}$ of the tree, and f_0 and f_1 are the automorphisms induced by f on the two subtrees of the root, which are naturally isomorphic to the whole tree. The automorphisms f such that $f = (f_0, f_1)\sigma$ are *odd*, the others *even*. One can then write the ambient group in terms of wreath products as

$$\mathsf{Aut}(\mathbf{2}^\star) \simeq \mathsf{Aut}(\mathbf{2}^\star) \wr \mathfrak{S}_2 = (\mathsf{Aut}(\mathbf{2}^\star) \times \mathsf{Aut}(\mathbf{2}^\star)) \rtimes \mathfrak{S}_2.$$

The group operation is given by $(f_0, f_1)s\ (g_0, g_1)t = (f_0 g_{s(0)}, f_1 g_{s(1)})\, st$, see [15,20]. For example, the cycle-cum-chord transducers from section 1 can be written as $\underline{0} = (\underline{n^-}, \underline{m^-})\,\sigma$ and $\underline{k} = (\underline{k^-}, \underline{k^-})$ for $0 < k < n$. Here $1 \leq m \leq n$ and we have written p^- rather than $p - 1$ to improve legibility.

The collection $\mathfrak{I}$ of all maps defined by invertible transducers is easily seen to be closed under inverse and composition, and thus forms a subgroup $\mathfrak{I}$ of the ambient group. For automata groups $G \subseteq \mathfrak{I}$ the wreath form naturally induces three maps ∂_0, ∂_1 and par such that $f = (\partial_0 f, \partial_1 f)\,\mathsf{par} f$. The parity is simply determined by the corresponding state being toggle or copy. The operations ∂_s are the *left residuals*, see [17,5,15]: for any word x, define the function $\partial_x f$ by $(x\, f)\,(z\, \partial_x f) = (xz)\, f$ for all words z (for transductions, we write function application on the right and use diagrammatic composition for consistency with relational composition). It follows that

$$\partial_{xy} f = \partial_y \partial_x f \qquad \partial_x (fg) = \partial_x f\, \partial_{xf} g$$

The transduction semigroup $\mathcal{S}(\mathcal{A})$ is naturally closed under residuals. In fact, we can describe the behavior of all the transductions by a transition system $\mathcal{C}$, much the way $\mathcal{A}$ describes the basic transductions: the states of $\mathcal{C}$ are all transductions in $\mathcal{S}(\mathcal{A})$ and the transitions are $f \xrightarrow{s/sf} \partial_s f$ where $s \in \mathbf{2}$. Thus $\mathcal{C}$ contains $\mathcal{A}$ as a subautomaton. Of course, this system is infinite in general; it is referred to as the *complete automaton* in [15]. The computation of $x\, f$ follows a path in $\mathcal{C}$.

The following characterization of the transduction semigroups of cycle-cum-chord transducers was established in [25].

Theorem 1. *The semigroup generated by a cycle-cum-chord transducer $\mathfrak{A}^n_m$ is a free Abelian group for $m < n$, and the Boolean group $\mathbf{2}^n$ for $n = m$.*

From now on we will ignore the degenerate case $n = m$. It is easy to see that the semigroup is Abelian. Letting $s = \gcd(n, m)$, $\mathfrak{A}^n_m$ generates the free Abelian group $\mathbb{Z}^{n-s}$. To simplify the discussion, let us assume that n and m are coprime; the general situation can be recovered by considering shuffle products of transductions in the coprime case. The reason the semigroup turns out to be a group is that the following *cancellation identity* holds:

$$\underline{0}^2\, \underline{1}^2 \ldots \underline{(m^-)}^2\, \underline{m}\, \underline{m+1} \ldots \underline{n^-} = I.$$

To show that there are essentially no other identities one can use a device suggested by Knuth in [12]. One enlarges the transducer by adding infinitely many copy states k where $k \geq n$ together with transitions $\underline{k} = (\underline{k^-}, \underline{k^-})$. This extension does not change the (semi)group generated by the machine. In fact we have the *shift identities*

$$\underline{k}^2 = \underline{k+m}\, \underline{k+n}.$$

Using these identities one can then show that for every transduction f there is a unique flat representation

$$f = \underline{k_1}\, \underline{k_2}\, \ldots\, \underline{k_r},$$

where $k_1 < k_2 < \ldots < k_r$. For $f = I$ we assume $r = 0$. We refer to this representation as the *Knuth normal form (KNF)* of f, in symbols $\mathsf{KNF}(f)$. Indeed, by interpreting the cancellation and shift identities as rewrite rules we obtain a weakly convergent rewrite system that produces $\mathsf{KNF}(f)$, given f in semigroup representation.

In particular for $\mathfrak{A}^3_2$, Knuth normal form has a number of interesting properties that will be important in section 3. For any transduction f, write $\mathsf{sh}^s(f)$ for the transduction obtained by replacing any term $\underline{k}$ in the KNF of f by $\underline{k+s}$. In group representation, we have $\mathsf{sh}^1(a, b) = (-2b, a - 2b)$. Lastly, let $\gamma_0 = \underline{0}$, $\gamma_1 = \underline{0}\,\underline{1}$, $\gamma_2 = \underline{0}^{-1}$ and $\gamma_3 = \underline{0}^{-1}\underline{1}^{-1}$ and set $\gamma'_i = \mathsf{sh}^1(\gamma_i)$. A straightforward induction shows the following lemma.

Lemma 1. *Let $0 \leq k$ and $0 \leq i < 4$. Then $\mathsf{KNF}(\underline{0}^{2^{4k+i}}) = \mathsf{sh}^{8k+2i}(\gamma_i)$. More generally, for $f = \underline{0}^a \underline{1}^b$, we have $\mathsf{KNF}(f^{2^{4k+i}}) = \mathsf{sh}^{8k+2i}(\mathsf{KNF}(\gamma_i^a\, \gamma_i'^b))$.*

Because of the lemma, for $\mathfrak{A}_2^3$, rewriting is not required at all to determine Knuth normal form, rather a finite state transducer suffices to determine KNF in the following sense. For simplicity let us only consider the KNF for $\underline{0}^t$, $t \geq 0$, rather than the general group elements. We can think of the KNF of f as an ω-sequence $\kappa \in \mathbf{2}^\omega$ where $\kappa_i = 1 \iff \underline{i}$ appears in the normal form of f. Likewise we can think of KNF as a finite bit-vector u such that $\kappa = u\,0^\omega$. We can pre-compute these finite bit-vectors of $\underline{0}^a$ for $0 \leq a < 16$ and pad to length 8 whenever necessary:

00000000 10000000 00110000 1011000 000010111 100010111
001110111 101110111 000000111 100000111 001100111 101100111
000010001 100010001 001110001 101110001

All but the first 4 entries have length 9 and require a "carry" to the next block. According to lemma 1 we can now determine KNF of $\underline{0}^t$ as follows. Let T be a 0-indexed table whose entries are the 16 KNFs, right-padded or truncated to form blocks of length 8. If there is no carry, on input hex-digit d the correct output is T_d, but with a carry it is $T_{d+1 \bmod 16}$. Figure 2 shows a sketch of the appropriate transducer; input is hexadecimal, output is binary.

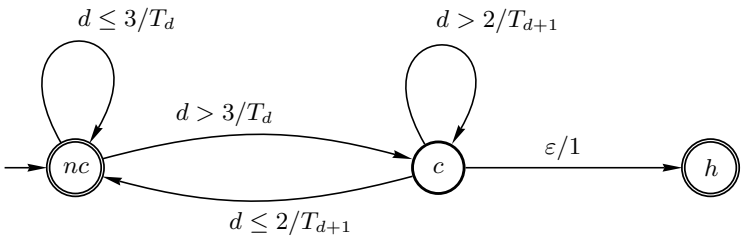


Fig. 2. A transducer that determines the Knuth normal form of a transduction $\underline{0}^a$ for CCC transducer $\mathfrak{A}_2^3$

The state nc is the no-carry state, c is carry, and h takes care of pending carries after the last input digit. For example, for $a = 3921 = (15F)_{16r}$ we get three blocks plus one 1 because of the carry:

$$T_1T_5T_0T_1 = 10000000\ 10001011\ 00000000\ 1,$$

corresponding to KNF $\underline{0}\ \underline{8}\ \underline{12}\ \underline{14}\ \underline{15}\ \underline{24}$. Note that the KNF transducer can be converted into a recurrence equation for the length of $\mathsf{KNF}(f)$, but it seems difficult to obtain a closed form solution. Also, a similar construction works for general group elements, but the machinery becomes considerably more complicated since we now have to deal with both generators $\underline{0}$ and $\underline{1}$ of the transduction group.

2.1 Orbit Rationality

Given a transduction f we can think of the associated orbit relation $f^\star$ as a language over $(\mathbf{2} \times \mathbf{2})^\star$. One can then exploit the group representation to calculate Brzozowski quotients of this language. We obtain a generally infinite transition system that recognizes the orbits of f and whose states naturally are given by pairs of transductions, see [25] for details. Somewhat surprisingly, for some CCC transducers this transition system turns out to be finite for all the associated transductions. Thus, $f^\star$ is rational and hence automatic. For space reasons we focus here on the CCC Transducer $\mathfrak{A}_2^3$, see the reference for the following result and some generalizations.

Theorem 2. *For any transduction f in $\mathcal{S}(\mathfrak{A}_2^3)$, the orbit relation of f is rational. Accordingly, the root function can be computed by a length-preserving finite state transducer.*

The proof is based on the explicit construction of an acceptor that recognizes the orbit relation of f, considered as a language over $(\mathbf{2} \times \mathbf{2})^\star$. As already mentioned, the construction uses Brzozowski quotients and is a priori only guaranteed to produce a potentially infinite transition system. However, for $\mathcal{S}(\mathfrak{A}_2^3)$ only finite systems are generated. For example, for $f = \underline{0}$ there are 34 states in the acceptor. Critical for finiteness is the fact that the following operation π on transductions has finite orbits: $\pi(f) = \partial_0 f$ for f even, and $\pi(f) = \partial_0 f\, \partial_1 f = \partial_0 f^2$ for f odd. As it turns out, except for the fixed point I, all orbits of π end in an 8-cycle.

Unsurprisingly, this property is not shared by all other transducers; for example, the orbit relation of $\underline{0}$ in $\mathfrak{A}_3^4$ fails to be rational. The proof comes down to showing that all powers of a certain rational matrix fail to have rational eigenvalues. In a first step one can exploit field theory to show that it suffices to check finitely cases, which cases can then be dispatched by computation in a computer algebra system. Needless to say, this argument is difficult to generalize and it is not clear how to characterize CCC transducers with rational orbits.

2.2 Computing Iterates

Knuth normal form also suggests that computing $x\, f^t$ can be computed easily: we have $az\, f = a\, f(z\, \partial_a f)$ and residuation for a transduction written in KNF comes down to a left shift, except possibly for a first term $\underline{0}$. Hence, after processing an initial segment of x, the residuals of f^t will have low weight and from then on, every single bit of x can be processed in constant time. In terms of the complete automaton $\mathcal{C}$ from section 2 this means that there are only a few non-trivial strongly connected components and every sufficiently long path winds up in one of them. For example, in the case of $\mathfrak{A}_2^3$ the complete automaton has 8 non-trivial strongly connected components the largest of which has 6 states.

Thus we have two natural representations for transductions: the semigroup representation $f = \underline{0}^{e_0} \underline{1}^{e_1} \ldots \underline{n-1}^{e_{n-1}}$ where $e_i \geq 0$, and the unique group representation $f = \underline{0}^{e'_0} \underline{1}^{e'_1} \ldots \underline{n-2}^{e'_{n-s-1}}$ where $e'_i \in \mathbb{Z}$. Correspondingly, the group representation of f is the integer-valued vector $(e'_0, \ldots, e'_{n-s-1})$. We will

refer to $\sum |e_i|$ as the *weight* of f. The weight can be used to bound the the complexity of the iteration problem: it is clear that we can compute residuals in time $O(n \log w)$ where w is the weight of the transduction in question. It follows that $x\,f$ can be computed in $O(|x|\, n \log w)$ time. However, we can do better than that.

Proposition 1. *Given a transduction $f \in \mathcal{S}(\mathfrak{A}_m^n)$ we can compute $x\,f$ in time linear in $|x|$, with coefficients depending on f.*

The idea is to express residuation as an affine operation of the form

$$\partial_s \boldsymbol{u} = \begin{cases} A \cdot \boldsymbol{u} & \text{if } \boldsymbol{u} \text{ is even,} \\ A \cdot \boldsymbol{u} - (-1)^s \boldsymbol{a} & \text{otherwise.} \end{cases}$$

where $\boldsymbol{u} \in \mathbb{Z}^{n-1}$ is the group representation of the transduction, see [16]. A is a rational matrix of suitable dimension and $\boldsymbol{a}$ a rational vector. The spectral radius of A is less than 1, hence residuation is a contraction and after a transient part all weights are bounded by a constant depending only on n and m.

We do not know how to obtain more precise bounds on the cost of computing $x\,f$. In particular there appears to be no easy way to determine the number and size of the non-trivial strongly connected components of the complete automaton, short of actual computation.

3 Timestamps and Coordinates

One can show that for any CCC transducer $\mathfrak{A}_m^n$ the group H of transductions generated by $\underline{p}$, $0 \le p < m$, acts transitively on $\mathbf{2}^\ell$ (which set of words is often referred to as a level set in connection with the infinite binary tree). For $\ell = km$ the quotient group H' obtained by factoring with respect to $\underline{i}^{2^k}$ acts simply transitively on the level set $\mathbf{2}^\ell$. As a consequence, there is a natural coordinate system for $\mathbf{2}^{km}$: for every $\ell = km$ there is a bijection

$$\mathbf{2}^\ell \to \mathbb{Z}/(2^k) \times \ldots \times \mathbb{Z}/(2^k)$$

where the product on the right has m terms. We will write $\langle\, w \,\rangle_\ell \in (\mathbb{Z}/(2^k))^m$ for the *coordinates* of a word w: $\langle\, w \,\rangle_\ell = (a_0, \ldots, a_{m^-})$ if, and only if, $w = 0^\ell\, \underline{0}^{a_0} \underline{1}^{a_1} \ldots \underline{m^-}^{a_{m^-}}$. We use $x \equiv y$ to express that two integer vectors of length m are componentwise congruent modulo 2^k. Also, for a transduction f, define the ℓ-coordinates of f by $\langle\, f \,\rangle_\ell = \langle\, 0^\ell\, f \,\rangle_\ell$. For example, in $\mathfrak{A}_2^3$, letting $f = \underline{0}^{-1}\underline{1}^3$ we get $\langle\, f \,\rangle_{2k} = (2^k - 1, 3)$ for $k \ge 2$. By commutativity it follows that $\langle\, 0^\ell\, f^i \,\rangle_\ell \equiv i \cdot \langle\, f \,\rangle_\ell$ and $\langle\, 0^\ell\, f^\star \,\rangle_\ell \equiv \mathbb{N} \cdot \langle\, f \,\rangle_\ell$, so that the orbit of 0^ℓ is a linear subspace of $(\mathbb{Z}/(2^k))^m$. Again by commutativity general orbits can be described as affine subspaces of $(\mathbb{Z}/(2^k))^m$:

$$\langle\, w\, f^\star \,\rangle_\ell \equiv \langle\, w \,\rangle_\ell + \mathbb{N} \cdot \langle\, f \,\rangle_\ell$$

Thus, it is of interest to be able to calculate coordinates. More formally, we wish to address the following problem, assuming a CCC transducer $\mathfrak{A}_m^n$ is fixed.

Problem:	**Coordinate Problem**
Instance:	A word $x \in \mathbf{2}^\ell$ where $\ell = km$.
Output:	The coordinates $\langle x \rangle_\ell \in (2^k)^m$ of x.

Closely related is the question how many times a given transduction f must be applied to obtain a particular point in the orbit of a given word x. We refer to this as the Timestamp Problem:

Problem:	**Timestamp Problem**
Instance:	A transduction f, two words $x, y \in \mathbf{2}^k$.
Output:	The least $t \geq 0$ such that $y = x\, f^t$, if it exists; `NO` otherwise.

Clearly the Orbit Problem reduces to the Timestamp Problem, which, as we will see shortly, in turn reduces to the Coordinate Problem. We will show that all of them can be solved in quadratic time. Let us first deal with the Timestamp Problem, see [24].

Theorem 3. *The Timestamp Problem can be solved in quadratic time: given two words x and y of length $\ell = km$ and a transduction $f \in \mathcal{S}(\mathfrak{A}_m^n)$ we can find a timestamp $t \geq 0$ such that $x\, f^t = y$, or determine that no such t exists, in $O(\ell^2)$ steps.*

The technique of the last theorem can be pushed slightly to provide a fast algorithm to compute coordinates. Suppose $x \in \mathbf{2}^\ell$ where $\ell = km$. We need to compute integers $e_0, \ldots, e_{m^-}$ such that

$$x = 0^\ell\, \underline{0}^{e_0} \ldots \underline{m^-}^{e_{m^-}}.$$

Let us call the transduction on the right f. Then for any $r < \ell$

$$x = 0^r\, f \cdot 0^{\ell - r} \partial_{0^r} f.$$

Since the first bit of $0^{\ell-r}\partial_{0^r} f$ depends only on the parity of $\partial_{0^r} f$ we can determine the coefficients of the binary expansions of the exponents e_i.

Theorem 4. *The Coordinate Problem can be solved in quadratic time: given a word x of length $\ell = km$ we can determine its coordinates in $O(\ell^2)$ steps.*

Given the algorithm for the Coordinate Problem one can also tackle the Timestamp Problem via a reduction.

Proposition 2. *The Timestamp Problem reduces to the Coordinate Problem in time $O(\ell \log w + \log^2 k)$ where w is the weight of the transduction and km is the length of the words.*

For some CCC transducers the quadratic bounds from the last few results can be improved upon: finite state machines sometimes suffice to calculate coordinates and timestamps. As an example, consider again $\mathfrak{A}_2^3$. The following algorithm solves the Coordinate Problem in this case. Given a word x (here assumed to be 0-indexed) we calculate its coordinates in reverse binary as follows. The γ_i are as in section 2.1.

```
// coordinate algorithm 𝔄₂³
    h = (0, 0);
    for r = 0, ..., n − 1 do
        s_r = h_1 + x_{2r} mod 2;          // phase 1: bind s_r
        h = ∂_0(h + s_r · γ_r);
        t_r = h_1 + x_{2r+1} mod 2;        // phase 2: bind t_r
        h = ∂_0(h + t_r · γ_r);
    return (s, t);
```

As stated, the algorithm appears to require quadratic time. However, it can be implemented on a finite state machine because of the contraction property of residuals spelled out in section 2.

Theorem 5. *The Coordinate Problem for $\mathfrak{A}_2^3$ can be solved by a transducer that computes the coordinates in reverse binary.*

It is straightforward to modify this algorithm to deal with timestamps.

4 Open Problems

We have characterized the complexity of various computational problems associated with the iteration of transductions defined by a rather narrow class of invertible binary transducers. In particular it can be shown that for these transducers iterates, time-stamps and coordinates can be computed quickly. The argument uses Knuth normal form as an essential technical device. Incidentally, we do not know in general when Knuth normal form can be computed by a finite state transducer as in section 2.1 rather than a canonical rewrite system.

One obvious open question is to determine the cycle-cum-chord transducers for which the orbit relation of any transduction is rational. The property appears to be quite rare, but currently we do not even know whether it is decidable. The rationality problem naturally carries over to other more complicated classes of invertible transducers. A particularly plausible generalization of cycle-cum-chord transducers are so-called m-lattices, invertible transducers whose transduction groups are isomorphic to $\mathbb{Z}^m$, see [20,16]. One well-known example are the so-called "sausage automata" in [15], given in wreath notation by $\underline{0} = (I, \underline{n})\,\sigma$ and $\underline{k} = (\underline{k-1}, \underline{k-1})$ for $2 \leq k \leq n$. Here we ignore the identity I, as is customary. We do not know to which degree the timestamp and/or coordinate machinery arguments carry over to m-lattices, though the particular example of the sausage automaton is easy to deal with. In this context an interesting question is whether isomorphism of our structure $\mathfrak{C}$ is decidable. Similarly we do not know what the expression complexity of model checking for these structures is in general.

It is tempting to consider general invertible transducers with but a single toggle state; after all, the impact of the complexity of the underlying groups (which may be very complicated as in Grigorchuk's example) on our decision problems is not clear a priori. Another plausible generalization would be to place restrictions on the topology of the underlying transition diagram.

It is straightforward to check whether $\mathcal{S}(\mathcal{A})$ is commutative, using standard automata-theoretic methods. We do not know whether it is decidable whether $\mathcal{S}(\mathcal{A})$ is a group, though this property is obviously semidecidable. Unsurprisingly, many other decidability questions regarding transduction semigroups or groups of invertible transducers are also open, see [7, chap. 7] for an extensive list.

References

1. Bartholdi, L., Silva, P.V.: Groups defined by automata. In: CoRR, abs/1012.1531 (2010)
2. Berstel, J.: Transductions and context-free languages (2009), `http://www-igm.univ-mlv.fr/~berstel/LivreTransductions/LivreTransductions.html`
3. Cook, M.: Universality in elementary cellular automata. Complex Systems 15(1), 1–40 (2004)
4. Fortnow, L., Grochow, J.A.: Complexity classes of equivalence problems revisited. Inf. Comput. 209(4), 748–763 (2011)
5. Gluškov, V.M.: Abstract theory of automata. Uspehi Mat. Nauk. 16(5(101)), 3–62 (1961)
6. Grigorchuk, R., Šunić, Z.: Self-Similarity and Branching in Group Theory. In: Groups St. Andrews 2005. London Math. Soc. Lec. Notes, vol. 339. Cambridge University Press (2007)
7. Grigorchuk, R.R., Nekrashevich, V.V., Sushchanski, V.I.: Automata, dynamical systems and groups. Proc. Steklov Institute of Math. 231, 128–203 (2000)
8. Howard Johnson, J.: Rational equivalence relations. Theoretical Computer Science 47, 167–176 (1986)
9. Khoussainov, B., Nerode, A.: Automatic presentations of structures. In: Leivant, D. (ed.) LCC 1994. LNCS, vol. 960, pp. 367–392. Springer, Heidelberg (1995)
10. Khoussainov, B., Nerode, A.: Automata Theory and its Applications. Birkhäuser (2001)
11. Khoussainov, B., Rubin, S.: Automatic structures: overview and future directions. J. Autom. Lang. Comb. 8(2), 287–301 (2003)
12. Knuth, D.: Private communication (2010)
13. Latteux, M., Simplot, D., Terlutte, A.: Iterated length-preserving rational transductions. In: Brim, L., Gruska, J., Zlatuška, J. (eds.) MFCS 1998. LNCS, vol. 1450, pp. 286–295. Springer, Heidelberg (1998)
14. Neary, T., Woods, D.: P-completeness of cellular automaton rule 110. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 132–143. Springer, Heidelberg (2006)
15. Nekrashevych, V.: Self-Similar Groups. AMS. Math. Surveys and Monographs, vol. 117 (2005)
16. Nekrashevych, V., Sidki, S.: Automorphisms of the binary tree: state-closed subgroups and dynamics of 1/2-endomorphisms. Cambridge University Press (2004)
17. Raney, G.N.: Sequential functions. J. Assoc. Comp. Mach. 5(2), 177–180 (1958)
18. Sakarovitch, J.: Elements of Automata Theory. Cambridge University Press (2009)
19. Serre, J.-P.: Arbres, Amalgames, SL_2. Astérisque Société Mathématique de France, Paris (1977)
20. Sidki, S.: Automorphisms of one-rooted trees: Growth, circuit structure, and acyclicity. J. Math. Sciences 100(1), 1925–1943 (2000)

21. Sutner, K.: On the computational complexity of finite cellular automata. J. Comput. System Sci. 50(1), 87–97 (1995)
22. Sutner, K.: Universality and cellular automata. In: Margenstern, M. (ed.) MCU 2004. LNCS, vol. 3354, pp. 50–59. Springer, Heidelberg (2005)
23. Sutner, K.: Computational classification of cellular automata. Int. J. General Systems 41(6), 1–13 (2012)
24. Sutner, K.: Invertible transducers, iteration and coordinates. In: Konstantinidis, S. (ed.) CIAA 2013. LNCS, vol. 7982, pp. 306–318. Springer, Heidelberg (2013)
25. Sutner, K., Lewi, K.: Iterating invertible binary transducers. In: Kutrib, M., Moreira, N., Reis, R. (eds.) DCFS 2012. LNCS, vol. 7386, pp. 294–306. Springer, Heidelberg (2012)