12-14-2007

Transducer dynamics

Egor Dolzhenko
University of South Florida

Transducer dynamics

by

Egor Dolzhenko

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Arts
Department of Mathematics
College of Arts and Sciences
University of South Florida

Major Professor: Nataša Jonoska, Ph.D.
Brian Curtin, Ph.D.
Gregory McColm, Ph.D.

Date of Approval:
December 14, 2007

Keywords: Sequences of Words, Finite State Automata with Output, Entropy,
Picture Languages, Local Languages

Acknowledgment

I would like to thank my adviser Dr. Nataša Jonoska for her support and motivation ever since I took my first proof based mathematics course with her. Also, I would like to thank my committee members, Dr. Brian Curtin and Dr. Gregory McColm for their help and useful comments to direct me in my thesis work.

Lastly, I would like to acknowledge the support of the National Science Foundation, as this work has been supported, in part, by the NSF grant CCF #0726396.

Contents

List of figures

Transducer Dynamics

Egor Dolzhenko

ABSTRACT

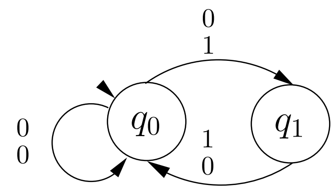
Transducers are finite state automata with an output. In this thesis, we attempt to classify sequences that can be constructed by iteratively applying a transducer to a given word. We begin exploring this problem by considering sequences of words that can be produced by iterative application of a transducer to a given input word, i.e., identifying sequences of words of the form $w, \tau(w), \tau^2(w), \ldots$. We call such sequences transducer recognizable. Also we introduce the notion of "recognition of a sequence in context", which captures the possibility of concatenating prefix and suffix words to each word in the sequence, so a given sequence of words becomes transducer recognizable. It turns out that all finite and periodic sequences of words of equal length are transducer recognizable. We also show how to construct a deterministic transducer with the least number of states recognizing a given sequence. To each transducer τ we associate a two-dimensional language $L^2(\tau)$, consisting of blocks of symbols in the following way. The first row, w, of each block is in the input language of τ, the second row is a word that τ outputs on input w. Inductively, every subsequent row is a word outputted by the transducer when its preceding row is read as an input. We show a relationship of the entropy values of these two-dimensional languages to the entropy values of the one-dimensional languages that appear as input languages for finite state transducers.

1 Introduction

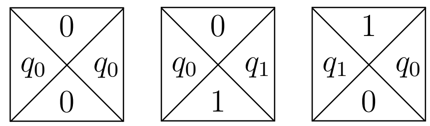
It is commonly acknowledged that DNA computing, as a branch of science, started by Adleman's paper [1]. Since then, many related models of computing have been developed and explored. The success of each such model depends on its computational power, robustness, and complexity. Some of these models are closely related to the concept of Wang tiles. We call a finite set of distinct unit squares with colored edges a set of *Wang prototiles*. We assume that each prototile appears in an arbitrarily large number of copies called *tiles*. A tile ξ with left edge colored l, bottom edge colored b, top edge colored t and right edge colored r is denoted with $\xi = [l, b, t, r]$. No rotation or reflexion of the tiles is allowed. Two tiles $\xi = [l, b, t, r]$ and $\xi' = [l', b', t', r']$ can be placed next to each other, ξ to the left of ξ' iff $r = l'$, and ξ' on top of ξ iff $t = b'$. More information about Wang tiles can be found in [6]. Recently, a physical representation of Wang tiles with DNA molecules has been demonstrated [13,14].

It is well known that by iteration of generalized sequential machines (finite state machines mapping symbols into strings) all computable functions can be simulated (see for ex. [9,10]). The full computational power depends on the possibility for iterations of a finite state machine. As there is a natural simulation of the process of iteration of transducers and recursive (computable) functions with Wang tiles [7], this idea has been developed further in [3] where a successful experimental simulation of a programmable transducer (finite state machine mapping symbols into symbols) with DNA Wang tiles having iteration capabilities is reported. This experimental development provides means for generating patterns and variety of two-dimensional arrays at the nano level.

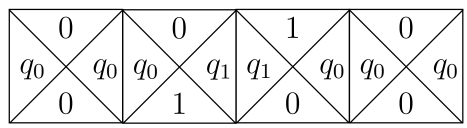
We give a brief example by illustrating connection of the transducers to Wang tiles (complete description of this model can be found in [7]). Consider transducer τ pictured below.



To each transition of this transducer we can associate Wang prototiles as follows.



Where each state and each symbol of the alphabet represents a distinct color. These tiles can be assembled into the row depicted below.



Notice that the bottom edge of this row represents input word 0100 and top edge represents 0010, that is, 0010 is the output of the transducer on the input 0100. Similarly we can construct another row of four tiles, with bottom edge representing word 0010 and the top edge word 0001 by stacking this row on top of the first. Continuing in this way we can construct a block of arbitrary height.

The above example illustrates the main goal of this work, the classification of patterns that can be generated by the described process. We begin exploring this problem by considering sequences of words that can be produced by iterative application of a transducer to a given input word, i.e., identifying sequences of words of the form $w, \tau(w), \tau^2(w), \dots$. We call such sequences transducer recognizable. Also we introduce notion of “recognition of a sequence in context”, which gives rise to a possibility of concatenating prefix and suffix to each word in the sequence, so a given sequence of words becomes transducer recognizable. It turns out that all finite and periodic sequences of words of equal length are transducer recognizable. Additionally we briefly explore other ways to iteratively apply a transducer to a word in order to get a sequence.

The next question which we consider is the following: Given a sequence of words $\mathbf{s}$, how can one construct a deterministic transducer with the least number of states recognizing **s**.

First, we confirm that after minor modifications we can apply an algorithm for minimization of Mealy machines to deterministic transducers. Hence if we apply this algorithm to any transducer we obtain an equivalent transducer with the smallest possible number of states. By equivalent transducers we mean transducers that accept the same set of words, and for each accepted word, both transducers produce the same output. In general, however, application of this algorithm to a transducer that recognizes sequence **s** does not result in a transducer with the smallest number of states that recognizes sequence **s**. As already noted, the minimization algorithm always results in a transducer equivalent to the original one, and two transducers recognizing sequence **s** do not have to be equivalent since they can differ on the words that are not part of the sequence **s**. To overcome this problem we define a relation on the set of states that indicates the states that can be in some sense joined together into one state, such that the resulting transducer remains deterministic and recognizes sequence **s**. We show that this relation provides a way for a transducer with minimal number of states to be constructed.

Next we note that every finite transducer recognizable sequence $w, \tau(w), \tau^2(w), \ldots$ can be associated with a two-dimensional block whose first row is w, second row is $\tau(w)$, and i^{th} row is $\tau^i(w)$. For a given transducer τ, we denote all possible two-dimensional blocks that can be constructed in this way by $L^2(\tau)$. We analyze the connection between $L^2(\tau)$ and the input language of τ, $L(\tau)$, from the point of view of local languages and entropy. We observe that the entropy $L^2(\tau)$ for deterministic transducer τ is always zero. However if we consider nondeterministic transducers, i.e. transducers that can have more than one output for an input word, this is no longer true. In fact it turns out that for any given nondeterministic transducer τ, the entropy of $L(\tau)$ is always an upper bound for the entropy of $L^2(\tau)$.

Note that parts of this thesis have been submitted for publication [5].

1.1 Notation

A nonempty finite set A is called an alphabet. Members of the alphabet A are called symbols. A word over alphabet A is a finite sequence of symbols from A, whose elements are written next to one another and not separated by commas. The length of a word w is a number of symbols in w and is denoted by $|w|$. A word of length zero is denoted by λ. Let A^* denote the set of all possible words over alphabet A and let $A^+ = A^* \setminus \{\lambda\}$. For a word w let $w(i)$ define i^{th} symbol of this word. Also if $w = w(1)w(2)\dots w(n-1)w(n)$ then $w^R = w(n)w(n-1)\dots w(2)w(1)$. For example, if $w = 011$ then $w(2) = 1$ and $w^R = 110$.

Definition 1.1. A *Deterministic transducer* is a six-tuple

$$\tau = (A, Q, \phi, \gamma, q_0, F),$$

where A is a finite alphabet, Q is a finite set of states, q_0 is an initial state ($q_0 \in Q$), F is a set of final states ($F \subseteq Q$), ϕ is a transition function ($\phi : Q \times A \to Q$), and γ is an output function ($\gamma : Q \times A \to A$).

Transducers are often represented, and even defined, by diagrams. Given a deterministic transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ we start constructing its diagram by depicting the names of the states in Q. Next, for each pair (q_1, a, q_2) in ϕ and (q_1, b, q_2) in γ we draw an arrow from q_1 to q_2. Above this arrow we put symbol $\frac{b}{a}$, and refer to a as an input label and to b as an output label of this arrow. We indicate that q_0 is an initial state by the small arrow pointing at it. Finally we circle final states, i.e. all the members of the set F.

Since most of this work deals with the deterministic transducers, we will refer to deterministic transducers simply as *transducers*.

Definition 1.2. For $\tau = (A, Q, \phi, \gamma, q_0, F)$, $q \in Q$, and $a \in A$, let $\phi^*(q, a) = \phi(q, a)$. For $w = av, a \in A$, and $v \in A^+$, let $\phi^*(q, w) = \phi^*(\phi(q, a), v)$.

Definition 1.3. For $\tau = (A, Q, \phi, \gamma, q_0, F)$, $q \in Q$, and $a \in A$, let $\gamma^*(q, a) = \gamma(q, a)$. For $w = av, a \in A$, and $v \in A^+$, let $\gamma^*(q, w) = \gamma(q, a)\gamma^*(\phi(q, a), v)$.

From now on ϕ refers to ϕ^* and γ refers to γ^*. Since ϕ coincides with ϕ^* and γ coincides with γ^* on A, this should not produce any ambiguities.

Definition 1.4. A word $w \in A^*$ is *accepted* by a transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ if $\phi(q_0, w) \in F$. The set of all words that τ accepts is denoted $L(\tau)$. If the state q in Q has the property that for all words w, $\phi(q, w) \notin F$ then such q is denoted as q_{junk}.

We assume that any transducer τ contains at most one state q_{junk}.

Definition 1.5. For a transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ and $w \in A^*$ if τ accepts w and $\gamma(q_0, w) = u$, then we write $\tau(w) = u$. Let $Rng(\tau)$ denote the set of all words u such that there is w, such that $\tau(w) = u$.

Definition 1.6. A *nondeterministic transducer* is a five-tuple

$$\tau = (A, Q, \phi, q_0, F)$$

where A is a finite alphabet, Q is a finite set of states, q_0 is an initial state ($q_0 \in Q$), F is a set of final states ($F \subseteq Q$) and ϕ is a transition relation ($\phi \subseteq Q \times A \times A \times Q$).

Note that all of the above notions are defined similarly for nondeterministic transducers, however, in the definition of the nondeterministic transducer, ϕ is not a function. For instance, given w in $L(\tau)$, $\tau(w)$ now defines a set, since τ may have more than one output on w.

2 Recognition in context

In [2] the authors describe a way to use Wang tiles to simulate a transducer τ on an input word w. Briefly, each tile represents a specific transition, with colors of the left and right edges encoding source and target states, and top and bottom colors encoding input and output symbols. There is a row of $|w|$ tiles with the bottom edge encoding word w, such that leftmost vertical edge encodes q_0, initial state of τ, the rightmost vertical edge of this row is one of the terminal states of τ, and all of the adjacent vertical edges encode the same state. This implies that top edge of this row encodes $\tau(w)$, the output of τ on input w. Note that when τ is deterministic, the row with the properties above is unique.

In case that word $\tau(w)$ is accepted by τ, there is another row with the properties mentioned above, such that its top edge is $\tau(\tau(w))$ and the bottom edge is $\tau(w)$. Continuing in this way and placing these rows on top of one another two dimensional block is obtained, corresponding to the sequence w, $\tau(w)$,$\tau^2(w), \ldots$.

The main goal of this section is to formalize the above discussion through the notions of recognition and recognition in context and to discuss what sequences of what periods can be obtained through such process. Furthermore, last section discusses sets of sequences such that each sequence in the set is of the form s_1, $\tau(s_1)$,$\tau^2(s_1), \ldots$, where s_1 denotes its first element. For example this may be the case when all sequences have similar structure and differ only in the length of their words.

2.1 Recognition

This section deals only with deterministic transducers.

Definition 2.1. If $\mathbf{s} = s_1, s_2, \ldots, s_k$, is a finite sequence of words, then we write $\#\mathbf{s} = k$ and if $\mathbf{s}$ is infinite, then $\#\mathbf{s} = \infty$.

Definition 2.2. A sequence $\mathbf{s} = s_1, s_2, \ldots$ is called *periodic* if there exists $p \in \mathbb{N}$ such that for all $s_i = s_{p+i}$, $i \in \mathbb{N}$. The least such p is called the *period* of $\mathbf{s}$.

Definition 2.3. Let $\mathbf{s} = s_1, s_2, s_3, \ldots$ be a sequence of words over alphabet A such that $|s_i| = |s_j|$ for all i, j. If there exists a deterministic transducer τ such that

$$s_{i+1} = \tau(s_i)$$

for $1 \leq i < \#\mathbf{s}$, then $\mathbf{s}$ is said to be *transducer recognizable* and τ is said to *recognize* $\mathbf{s}$.

Definition 2.4. A transducer τ *recognizes precisely* a sequence $\mathbf{s}$ if τ recognizes $\mathbf{s}$ and for any other sequence $\mathbf{t}$, such that τ recognizes $\mathbf{t}$, there is a natural number n, such that $\tau^n(s_1) = t_1$. Here s_1 and t_1 denote first elements of sequences $\mathbf{s}$ and $\mathbf{t}$ respectively.

Proposition 2.1. *Let $\boldsymbol{s} = s_1, s_2, \ldots$ be a sequence of words over A with $|s_i| = |s_j| = k$ for all $1 \leq i, j < \#\boldsymbol{s}$. Then this sequence is transducer recognizable if and only if following holds:*

$$\textit{For all } r = 1, \ldots, k, \ \textit{if } \forall t = 1, \ldots, r, \ s_i(t) = s_j(t) \ \textit{then } s_{i+1}(r) = s_{j+1}(r) \ \textit{for all } i, j. \quad (1)$$

Proof. In case (1) holds, consider the transducer (note that λ denotes an empty word)

$$\tau = (A, Q, \phi, \gamma, \lambda, F)$$

that would recognize $\mathbf{s}$, where Q, the set of states, is defined by

$$Q = \{\lambda\} \cup \{s_i(1) \ldots s_i(t) | s_i \in \mathbf{s} \text{ and } 1 \leq t \leq k\}.$$

since k is fixed, this set is finite.

$$F = \{s_i | s_i \text{ is a member of a sequence } \mathbf{s}\}$$

For every $s_i \in \mathbf{s}$ such that $1 \leq i < \#\mathbf{s}$ define

$$\phi(s_i(1) \ldots s_i(t-1), s_i(t)) = s_i(1) \ldots s_i(t)$$

and

$$\gamma(s_i(1) \ldots s_i(t-1), s_i(t)) = s_{i+1}(t).$$

Also, add $\phi(\lambda, s_i(1)) = s_i(1)$ and $\gamma(\lambda, s_i(1)) = s_{i+1}(1)$. Then the output symbol is uniquely determined by $s_i(1)\ldots s_i(t)$ due to (1). Hence τ is deterministic. Finally add q_{junk} to Q and let all the missing transitions lead to it. Note that $\phi(q_0, w) \in F$, i.e., this transducer accepts word w if and only if $w \in \mathbf{s}$ and that $\gamma(\lambda, s_i) = s_{i+1}$ by construction. Thus τ recognizes $\mathbf{s}$.

Conversely, if for some $s_i, s_j \in \mathbf{s}$ with $1 \leq i, j < \#\mathbf{s}$ and for some l have $s_i(1)\ldots s_i(l) = s_j(1)\ldots s_j(l)$ but $s_{i+1}(l) \neq s_{j+1}(l)$ and there is deterministic transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ that recognizes $\mathbf{s}$ then let $q = \phi(q_0, s_i(1)\ldots s_i(l-1))$ $= \phi(q_0, s_j(1)\ldots s_j(l-1))$. This is a contradiction, since $s_{i+1}(l) = \gamma(q, s_i(l)) \neq \gamma(q, s_j(l)) = s_{j+1}(l)$. □

If $\mathbf{s}$ is a transducer recognizable sequence, let $\tau_{\mathbf{s}}$ denote the transducer constructed by the algorithm in the proposition 2.1 that recognizes $\mathbf{s}$. Note some of the properties of τ_s:

i. If $\phi(q_0, w) \in F$, then by construction of $\tau_{\mathbf{s}}$, $\phi(q_0, w) = w$ and $w \in \mathbf{s}$

ii. If $\phi(q_0, w') = \phi(q_0, w'')$, then $w' = w''$, since by i, $w' = \phi(q_0, w') = \phi(q_0, w'') = w''$.

iii. The transducer $\tau_{\mathbf{s}}$ recognizes $\mathbf{s}$ precisely, since τ recognizes $\mathbf{s}$ and for any other sequence, $w, \tau(w), \tau^2(w)\ldots$, that τ recognizes, w must be accepted by τ and then, by i, $w \in \mathbf{s}$.

Suppose $|A| = 2$ and let $\mathbf{s}$ be a transducer recognizable periodic sequence over alphabet A of period greater than one. Then the period of $\mathbf{s}$ must be even. To see this suppose $A = \{a, b\}$. Let t be the least natural number such that there is i and j, $s_i(t) \neq s_j(t)$. Then it follows that $s_1(t)s_2(t)s_3(t)\ldots = ababab\ldots$ or $s_1(t)s_2(t)s_3(t)\ldots = bababa\ldots$. This is so, since due to determinism of τ, it must be true that $\phi(q_0, s_i(1)\ldots s_i(t-1)) = \phi(q_0, s_j(1)\ldots s_j(t-1)) =: q'$ for all i, j. First, $\gamma(q', a) = a$ and $\gamma(q', b) = b$ can not happen by the choice of t, since above implies that $s_i(t) = s_1(t)$ for all i. Second, $\gamma(q', a) = a$ and $\gamma(q', b) = a$ can't happen either, since in this case $s_i(t) = a$ for all i or $s_1(t) = b$ and $s_i(t) = a$, $i > 1$ in which case the sequence is not even periodic. Thus one of the two cases mentioned above must be true, which implies that period of the sequence $\mathbf{s}$ must be even.

Proposition 2.2. *For each natural number n, there exists a transducer recognizable sequence $\{s_i\}_0^\infty$ over alphabet $A = \{a_0, a_1, \dots, a_k\}$ of period $|A|^n$.*

Proof. (by induction on n) For $n = 1$, let $s_i = a_{i \bmod |A|}$. Since $s_i = s_j$ implies that $a_{i \bmod |A|} = a_{j \bmod |A|}$ and $i \equiv j \pmod{|A|}$ we have that $i + 1 \equiv j + 1 \pmod{|A|}$ and $s_{i+1} = s_{j+1}$. Thus this sequence satisfies the conditions of theorem 2.1, and thus it is transducer recognizable and clearly of period $|A|$. Assume that the premise holds for $n = t - 1$, i.e. that there exists a sequence $\{s_i^{'}\}_0^\infty$ of period $|A|^{t-1}$.

For $n = |A|^t$ let $s_i = s_i^{'} a_{m \bmod |A|}$ such that $i = m|A|^{t-1} + r$ where $0 \le r < |A|^{t-1}$. Let $s_i = s_j$. By the division algorithm we get that $i = m_1|A|^{t-1} + r_1$ and $j = m_2|A|^{t-1} + r_2$. Since $s_i = s_j$ implies that $s_i^{'} = s_j^{'}$, we have that $i \equiv j \pmod{|A|^{t-1}}$ and hence $r_1 = r_2$. Thus $i - j = (m_1 - m_2)|A|^{t-1}$. Since it is also true that $a_{m_1 \bmod |A|} = a_{m_2 \bmod |A|}$, we have $m_1 = m_2 \pmod{|A|}$ and hence $i - j = h|A|^k$ for some natural number h, i.e., $i \equiv j \bmod |A|^t$.

One the other hand, if $i \equiv j \pmod{|A|^t}$ then $s_i^{'} = s_j^{'}$ due to inductive assumption and thus by construction $s_i = s_j$. Thus period of this sequence is $|A|^t$.

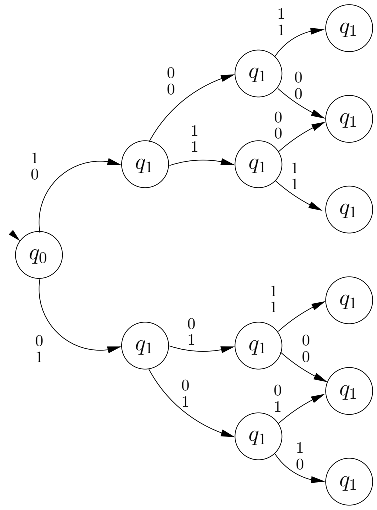
To verify that this sequence is transducer recognizable, we need to check that for every r $(1 \le r \le t)$, $s_i(1) \dots s_i(r) = s_j(1) \dots s_j(r)$ implies that $s_{i+1}(r) = s_{j+1}(r)$. If $r = t$ then equality follows, since $s_i = s_j$ implies $i + 1 \equiv j + 1 \pmod{|A|^n}$ thus $s_{i+1} = s_{j+1}$. If $r < t$ then the conclusion follows from the inductive assumption. ☐

As an example, consider a sequence of period 8 and the transducer recognizing it on the Figure 2.1

Definition 2.5. Let $\mathbf{s} = s_1, s_2, s_3, \dots$ be a sequence of words over alphabet A and suppose that there exists a deterministic transducer τ and sequence $\mathbf{p} = p_1, p_2, p_3, \dots$ such that for all i, j $|p_i| = |p_j|$ and sequence $\mathbf{d} = d_1, d_2, d_3, \dots$ with all $\forall i, j$ $|d_i| = |d_j|$, with $\#\mathbf{d} = \#\mathbf{p} = \#\mathbf{s}$ such that sequence τ recognizes $p_1s_1d_1, p_2s_2d_2, p_3s_3d_3 \dots$. Then we say that $\mathbf{s}$ is *recognizable* in context, and that τ *recognizes* $\mathbf{s}$ *in context.* Sequences $\mathbf{p}$ and $\mathbf{d}$ are called *prefix* and *suffix* respectively.

Corollary 2.1. *Every finite sequence consisting of words of equal length is recognizable in context.*

Proof. Suppose that the length of the sequence $\{w_i\}$ is k, then pick n so that $k \le |A|^n$, and let $\{p_i\}$ be the sequence of period $|A|^n$ as it is constructed in Proposition 2.2. Then



1 1 1
0 1 1
1 0 1
0 0 1
1 1 0
0 1 0
1 0 0
0 0 0

Figure 1: Sequence of period 8 and transducer recognizing it

define new sequence $\{s_i\}_0^{k-1}$ by $p_i = b_i$, $i = 0 \dots k-1$. Then $\{w_i\}$ is recognizable in context with prefix $\{p_i\}$. To see this note that if $p_i w_i = p_j w_j$ then $p_i = p_j$, hence $i = j$. In other words sequence $\{p_i w_i\}_1^k$ suffices the premise of the Proposition 2.1.

□

Corollary 2.2. *Every sequence consisting of words of equal length of period $|A|^n$ is recognizable in context.*

Proof. Let $\{w_i\}$ be the sequence of period $|A|^n$. Theorem 2.2 yields transducer recognizable sequence sequence $\{p_i\}$ of period $|A|^n$. Hence, similarly to the previous corollary, sequence $\{w\}$ is recognizable in context with prefix $\{p_i\}$. □

Definition 2.6. Let $D = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots\}$ be a set of sequences of words of equal length over alphabet A. Then if there exists a deterministic transducer τ, such that τ recognizes $\mathbf{s}_i$ in context, for all $i = 1, \dots, |D|$, with the same prefix and suffix for every set $\mathbf{s}_i$, then we say that τ *recognizes set* D and that set D is *recognizable in context.*

This way we require of one transducer to recognize a set of sequences with the same prefix and suffix.

Example 2.1. Set $M = \{\{0^{n-i}10^{i-1}\}_{i=1}^n | n = 2, 3, \dots\}$ is not recognizable in context.

Proof. Suppose that there exists τ such that τ recognizes M in context with prefix $\mathbf{p}$ and suffix $\mathbf{d}$. Consider the first element of the sequence $\{0^{n-i}10^{i-1}\}_{i=1}^{n} \in M$, such that $n-1$ is greater than twice the number of states of τ. Then it follows that on the sub word $0^{n-1}1$ of $p_1 0^{n-1} 1 d_1$ transducer goes through states $q_{t_1}, q_{t_2}, q_{t_3}, \ldots, q_{t_{n+1}}$ and yields $0^{n-2}10$, i.e.,

$$\phi(q_{t_1}, 0) = (0, q_{t_2}),$$
$$\phi(q_{t_2}, 0) = (0, q_{t_3}),$$
$$\ldots$$
$$\phi(q_{t_{n-1}}, 0) = (1, q_{t_n}),$$
$$\phi(q_{t_n}, 1) = (0, q_{t_{n+1}}).$$

By the pigeon hole principle, there are t_m, t_l such that $q_{t_m} = q_{t_l}$ and $t_m < t_l \leq n/2$. Since $q_{t_m} = q_{t_l}$ $\forall j = l \ldots n-1$, $\exists \psi \in \{t_m, \ldots, t_l\}$ such that $q_{t_j} = q_\psi$. Hence it must be true, that $\phi(q_{t_{n-1}}, 0) = (0, q_{t_n})$, but, by definition, $\phi(q_{t_{n-1}}, 0) = (1, q_{t_n})$. Contradiction. □

2.2 Modifications and Future Directions

From Example 2.1, it can be seen that it is possible to construct a relatively simple set of sequences that can not be recognized in context. Consider sequences $\mathbf{x} = x_1, x_2, x_3, x_4$ and $\mathbf{y} = y_1, y_2, y_3, y_4$ where x_i and y_i are defined by

$$\begin{array}{lllllll} y_1 & = & 0001 & \quad & x_1 & = & 1000 \\ y_2 & = & 0010 & & x_2 & = & 0100 \\ y_3 & = & 0100 & & x_3 & = & 0010 \\ y_4 & = & 1000 & & x_4 & = & 0001 \end{array}$$

Both of the sequences have very similar structure in a sense that $x_i = y_i^R$. However, using Proposition 2.1, it is easy to check that $\mathbf{x}$ is transducer recognizable and $\mathbf{y}$ is not. If we modify the Definition 2.3 by inserting $s_i = (\tau(s_{i-1}^R))^R$ instead of $s_i = \tau(s_{i-1})$, then $\mathbf{y}$ becomes recognizable and $\mathbf{x}$ is not. In both cases we face the same problem - deterministic transducer must determine the output based on the part of the word it have already read. This situation can be improved by applying the transducer twice to each word, in the sense of the following definition.

Definition 2.7. Let $\mathbf{s} = s_1, s_2, s_3, \ldots$ be a sequence of words over alphabet A such that for all i, j $|s_i| = |s_j|$. If there exists a deterministic transducer τ and sequence $\mathbf{p} = p_1, p_2, p_3, \ldots$ such that $\forall i, j$ $|p_i| = |p_j|$, and sequence $\mathbf{d} = d_1, d_2, d_3, \ldots$ with $\forall i, j$ $|d_i| = |d_j|$ and $\#\mathbf{d} = \#\mathbf{p} = \#\mathbf{s}$ such that

$$p_{i+1}s_{i+1}d_{i+1} = \tau((\tau((p_i s_i d_i)^R))^R)$$

for $i = 1, 2, 3, ..., \#\mathbf{s} - 1$. Then τ is said to *recognize* $\mathbf{s}$ *with a flip in context* and $\mathbf{s}$ is *recognizable with a flip in context.*

And hence we can correspondingly adjust definition for recognition of a set.

Definition 2.8. Let $D = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots\}$ be a set of sequences of words over alphabet A. Then if there exists a deterministic transducer τ such that τ recognizes $\mathbf{s}_i$ with a flip in context for all $i = 1 \ldots |D|$, with the same prefix and suffix for every set $\mathbf{s}_i$, then we say that τ *recognizes set* D *with a flip in context* and that set D is *recognizable with a flip in context.*

Proposition 2.3. *All transducer recognizable sequences are recognizable with a flip in context.*

Proof. Let $\mathbf{s} = s_1, s_2, s_3 \ldots$ be a transducer recognizable sequence of words over A. Let $\tau = (A, Q, \phi, \gamma, q_0, F)$ be the transducer that recognizes $\mathbf{s}$. Using τ lets define $\tau' = (A, Q', \phi', \gamma', q_0', f')$ as it is done in Figure 2.2. Let prefix $\mathbf{p}$ and suffix $\mathbf{d}$ be defined by $p_i = 0$ and $d_i = 1$ for all i. This way τ' will be able to distinguish between w and w^R for each $w \in \mathbf{s}$. Since $\tau'((\tau'((p_i s_i d_i)^R))^R) = \tau'((\tau'((0 s_i 1)^R))^R) = \tau'((\tau'(1 s_i^R 0))^R) = \tau'((1 s_i^R 0)^R) = \tau'(0 s_i 1) = 0 s_{i+1} 1$, it follows that $\mathbf{s}$ is recognizable with a flip in context.

□

Example 2.2. $M = \{\{0^{n-i}10^{i-1}\}_{i=1}^{n} | n = 2, 3, \ldots\}$ is recognizable with a flip in context.

Proof. Let $\mathbf{s} \in M$. Thus $\mathbf{s} = \{0^{t-i}10^{i-1}\}_{i=1}^{t}$ for some t. Then $s_i = 0^{t-i}10^{i-1}$ is the i^{th} word in the sequence $\mathbf{s}$. Define prefix $\mathbf{p}$ and suffix $\mathbf{d}$ by $p_i = 1$ and $d_i = 0$ for each i. Let transducer τ be as it is defined in the Figure 2.2. Then

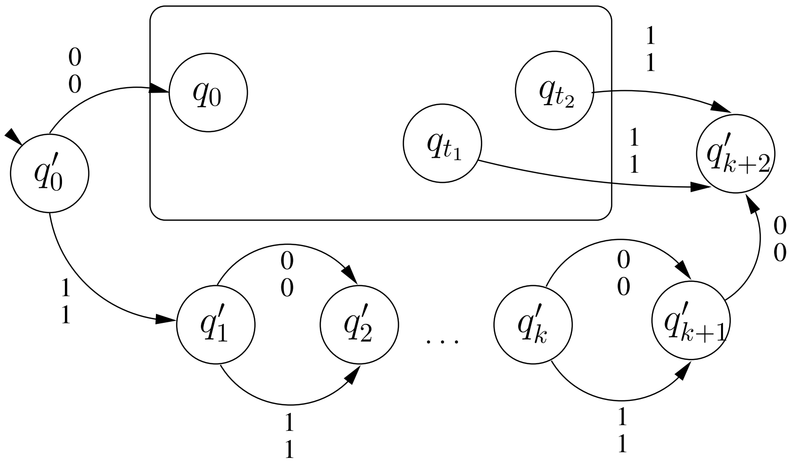


Figure 2: Transducer used in Proposition 3.3

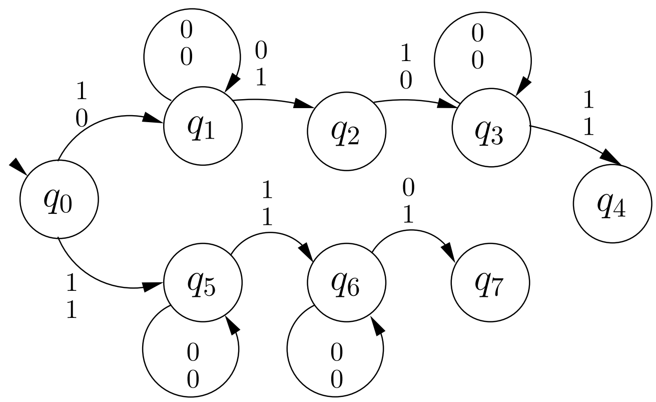


Figure 3: Transducer that recognizes set M with a flip

$\tau((\tau(w_i^R))^R) = \tau((\tau((10^{t-i}10^{i-1}0)^R))^R) = \tau((\tau(00^{i-1}10^{t-i}1))^R) = \tau((10^i10^{t-i-1}1)^R) = \tau(10^{t-i-1}10^i1) = 10^{t-i-1}10^i0 = 1s_{i+1}0$. Thus set M is recognizable with a flip in context. □

3 Minimization

A Mealy machine is a five tuple $\tau = (A, Q, \phi, \gamma, q_0)$ whose elements have the same definition as the first five elements of the deterministic transducer. Hence deterministic transducer are somewhat more general than Mealy machines, which have been studied well. However aim of the following section is to show that when it comes to minimization the same algorithm can be applied. The second section deals with finding in some sense smallest deterministic transducer, if one exists, that can recognize given sequence.

3.1 Minimization of a Deterministic Transducer

The definition of the deterministic transducer described above is more general than that of the Mealy machine, however, only difference is the presence of the final states in deterministic transducer, which can be confirmed in [4]. Thus it seems plausible to use algorithm, similar to the algorithm for the minimization of a Mealy machine, that would take into account the set of the final states. Hence the content of this section is, although modified for more general case, taken from [4].

Definition 3.1. A state a of transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ is accessible if there is some input word $w \in A^*$ such that $\phi^*(q_0, w) = q$. Transducer τ is connected if every state is accessible.

For simplicity assume that all transitions going to the state q_{junk} have input symbol equal to the output symbol, i.e. $\phi(q, a) = (a, q_{junk})$, where q_{junk} is the state for which $\forall w \in A^* \phi(q_{junk}, w) \notin F$. Also, without loss of generality, it is assumed that q_{junk} is the only state with this property and that every transducer is connected.

Definition 3.2. For a transducer τ and $w \in A^*$, let $\tau(w) = u$ if τ accepts w and outputs u.

Definition 3.3. Two transducers

$$\tau_1 = (A_1, Q_1, \phi_1, \gamma_1, q_{0_1}, F_1)$$

and

$$\tau_2 = (A_2, Q_2, \phi_2, \gamma_2, q_{0_2}, F_2)$$

are equivalent if and only if

1. $A_1 = A_2$, and
2. $L(\tau_1) = L(\tau_2)$ and for all $w \in L(\tau)$ $\tau_1(w) = \tau_2(w)$.

As a relation, equivalence establishes an equivalence relation on a set of all deterministic transducers.

Definition 3.4. Two states q_a and q_b of transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ are equivalent if and only if

$$\tau_a = (A, Q, \phi, \gamma, q_a, F)$$

is equivalent to

$$\tau_b = (A, Q, \phi, \gamma, q_b, F)$$

.

Proposition 3.1. *Two states q_a and q_b of a transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ are equivalent if and only if*

1. *$\forall s \in A$ $\gamma(q_a, s) = \gamma(q_b, s)$, and*
2. *$\forall s \in A$ $\phi(q_a, s)$ is equivalent to $\phi(q_b, s)$.*

Proof. If both of the conditions hold, then let $w \in A^*$ and $w = sn$ where $s \in A$, then $t = \gamma(q_a, s) = \gamma(q_b, s)$ and $\tau_{\phi(q_a,s)}(n) = \tau_{\phi(q_b,s)}(n) = k$. Thus $\tau_{q_a}(w) = \tau_{q_b}(w) = tk$.

Conversely, let q_a be equivalent to q_b, then let $w = sn$, where $s \in A$ and $n \in A^*$. Since $\tau_a(sn) = \tau_b(sn)$ it follows that $\tau_{\phi(q_a,s)}(n) = \tau_{\phi(q_b,s)}(n)$ hence $\phi(q_a, s)$ is equivalent to $\phi(q_b, s)$. Also $\gamma(q_a, s) = \gamma(q_b, s)$ since if $\phi(q_a, s) \notin q_{junk}$ there exists $w^{'}$ such that $\tau_{\phi(q_a,s)}$ accepts $w^{'}$ and hence τ_{q_a} accepts $sw^{'}$, implying that $\tau_{q_a}(sw^{'}) = \tau_{q_b}(sw^{'})$, thus $\gamma(q_a, s) = \gamma(q_b, s)$. If $\phi(q_a, s) = q_{junk}$ and $\phi(q_b, s) = q_{junk}$ then $\gamma(q_a, s) = \gamma(q_b, s) = s$ by previous assumption that transitions leading to q_{junk} produce output equal to input. □

Definition 3.5. A transducer is *reduced* if it contains no pair of equivalent states.

Definition 3.6. States q_a and q_b of a transducer $\tau = (A, Q, \phi, \gamma, q_0, F)$ are *k-distinguishable* if there exists a word $w \in A^*$ $|w| \leq k$, such that

$$\tau_{q_a}(w) \neq \tau_{q_b}(w).$$

Then w is called a *distinguishing word*.

Definition 3.7. If two states q_a and q_b are not k distinguishable, then they are *k-equivalent*.

Proposition 3.2. *Two states q_a and q_b of a transducer τ are k-equivalent if and only if*

1. *They are 1-equivalent.*

2. *For each $s \in A^*$ $\phi(q_a, s)$ and $\phi(q_b, s)$ are $k-1$ equivalent.*

Proof. Essentially the same as in previous proposition. □

If two states of a transducer are k-equivalent for all k then they are equivalent. This relation defines a partition of the set of the states of a transducer, which is used for construction of a new, reduced transducer. The algorithm given in [4] could be used in spite the fact that it was written for Mealy machines, as long as appropriate definitions are used.

3.2 Minimal Transducer that Recognizes Sequence of Words

In the previous section we tried to minimize the number of states in the deterministic transducer. For instance given a transducer recognizable sequence of words $\mathbf{s}$ the algorithm in [4] can be applied to $\tau_{\mathbf{s}}$ to find an equivalent transducer, but with the minimal number of states. On the other hand if, for a given sequence $\mathbf{s}$, we need to find a transducer with the minimal number of states that would recognize $\mathbf{s}$, the above algorithm would not work, as there may be other transducers, not equivalent to $\tau_{\mathbf{s}}$ that also recognize $\mathbf{s}$. Probably the most basic algorithm for finding the minimal transducer that would recognize a given sequence $\mathbf{s}$ could proceed as follows:

1. For a given sequence $\mathbf{s}$ construct $\tau_{\mathbf{s}}$ using algorithm outlined in Proposition 2.1.

2. Since there are only finitely many transducers over finite alphabet with fixed number of states, enumerate all transducers that have fewer states than the transducer constructed in step 1.

3. Since $\mathbf{s}$ can not be aperiodic to be transducer recognizable and there are only $|A|^k$ different words of length k over alphabet A, a transducer with the minimal number of states recognizing $\mathbf{s}$ can be found in finitely many steps.

In the previous section, an equivalence relation on the set of states was used to determine the states that are equivalent and a new transducer - obtained through equivalence classes of that relation - was equivalent to the original one. Here, a slightly different approach must be taken. The idea is to examine a new relation on the set of states that would indicate the states that in some sense can be joined together without affecting the transducer's ability to recognize a given sequence.

Let $\tau^{'}$ denote the transducer constructed from τ by making all of the states of τ, except q_{junk}, final. Then let $\text{dom}(\tau)$ denote all of the words accepted by $\tau^{'}$.

Definition 3.8. Let $q_1, q_2 \in Q$, then states q_1 and q_2 are in relation ϵ ($q_1 \epsilon q_2$) if and only if for all $w \in dom(\tau_{q_1}) \cap dom(\tau_{q_2})$ have that $\gamma^*(q_1, w) = \gamma^*(q_2, w)$.

Proposition 3.3. *If $q_1 \epsilon q_2$ and $s \in dom(\tau_{q_1}) \cap dom(\tau_{q_2}) \cap A$ then $\phi(q_1, s) \epsilon \phi(q_2, s)$.*

Proof. Let q_1,q_2,s be as described in premise and $\neg \phi(q_1, s) \epsilon \phi(q_2, s)$. Then it follows that $\exists w \in dom(\tau_{\phi(q_1,s)}) \cap dom(\tau_{\phi(q_2,s)})$, such that $\gamma^*(\phi(q_1, s), w) \neq \gamma^*(\phi(q_2, s), w)$. Then $sw \in dom(\tau_{q_1}) \cap dom(\tau_{q_2})$ and $\gamma^*(q_1, sw) = \gamma(q_1, s)\gamma^*(\phi(q_1, s), w) \neq \gamma(q_2, s)\gamma^*(\phi(q_2, s), w) = \gamma^*(q_2, sw)$, which is a contradiction. □

Proposition 3.4. *Let τ be a deterministic transducer recognizing the sequence $\boldsymbol{s} = s_1, s_2, \ldots$, and let $E \subseteq \{(u, v) | u, v \in Q$ and $u \epsilon v\}$ be such that:*

- *E defines an equivalence relation on set of states Q*
- *$\forall (u, v) \in E$, if $a \in dom(\tau_u) \cap dom(\tau_v) \cap A$, then $(\phi(u, a), \phi(v, a)) \in E$.*

Then there exists $\tau^{'}$, with set of states equal to the set of the equivalence classes produced by E, such that $\tau^{'}$ recognizes $\boldsymbol{s}$.

Proof. Let $Q^{'}$ denote the set of equivalence classes induced by E. Denote the equivalence class to which q belongs by $[q]$. Then $\forall[q] \in Q^{'}$ and $\forall a \in A$, let $q^{'} \in [q]$ be a state such that $\phi(q^{'}, a) \neq q_{\text{junk}}$. Now define $\phi^{'}([q], a) = [\phi(q^{'}, a)]$ and $\gamma^{'}([q], a) = \gamma(q^{'}, a)$. If $\forall p \in [q]$, $\phi(p, a) = q_{\text{junk}}$ then $\phi^{'}([q], a) = [q_{\text{junk}}]$ and $\gamma^{'}([q], a) = a$. Due to the assumptions, if $q_1, q_2 \in [q]$, and $\phi(q_1, s) \neq q_{\text{junk}}$ and $\phi(q_2, s) \neq q_{\text{junk}}$, i.e. , $s \in \text{dom}(\tau_{q_1}) \cap \text{dom}(\tau_{q_2}) \cap A$, then $[\phi(q_1, a)] = [\phi(q_2, a)]$ and thus the above construction produces a deterministic transducer $\tau^{'} = (A, Q^{'}, \phi^{'}, \gamma^{'}, [q_0], Q^{'})$ without any ambiguities.

We show that $\tau^{'}$ recognizes sequence $\mathbf{s}$. Let s_k and s_{k+1} be two consecutive words of the sequence $\mathbf{s}$. Let τ go through the states $q_0, q_1, \ldots, q_n$ on input s_k, where $\phi(q_i, s_k(i+1)) \neq q_{\text{junk}}$. Correspondingly, the transducer $\tau^{'}$ will go through the sequence of states $[q_0], [q_1], \ldots, [q_n]$ and produce s_{k+1}, since for any $i = 0 \ldots n-1$ $\phi(q_i, s_k(i+1)) = q_{i+1}$ implies that $\phi^{'}([q_i], s_k(i+1)) = [q_{i+1}]$ and $\gamma^{'}([q_i], s_k(i+1)) = \gamma(q_i, s_k(i+1)) = s_{k+1}(i+1)$. Thus this transducer recognizes $\mathbf{s} = s_1, s_2, \ldots$ and the size of $|Q|$ is equal to the number of equivalence classes of $|E|$. □

Let E be a relation on the set of states of the deterministic transducer τ that satisfies the premise of the Proposition 3.4. We say that E *yields* $\tau^{'}$ if $\tau^{'}$ is constructed through the algorithm outlined in the Proposition 3.4 using relation E.

Proposition 3.5. *Let $\mathbf{s}$ be the sequence of words over A and let $\tau_{\mathbf{s}}$ be the transducer constructed by the algorithm in Proposition 2.1 that recognizes $\mathbf{s}$. Let E be a set that satisfies the conditions of Proposition 3.4, that would produce the smallest number of equivalence classes. Then E yields a transducer with minimal number of states that recognizes the sequence s.*

Proof. Let $\tau_s = (A, Q, \phi, \gamma, q_0, F)$, and let $\tau^{'} = (A, Q^{'}, \phi^{'}, \gamma^{'}, q_0^{'}, F^{'})$ denote a minimal transducer with the minimal number of states that would recognize $\mathbf{s}$.
Consider following relation:
$p \sim q$ if and only if $p, q \neq q_{junk}$ and there are words $n_1, n_2 \in A^*$ such that $\phi(q_0, n_1) = p$ and $\phi(q_0, n_2) = q$ and $\phi^{'}(q_0^{'}, n_1) = \phi^{'}(q_0^{'}, n_2)$. We say that words n_1 and n_2 *correspond* to states p and q.

1. Relation $\sim$ is an equivalence relation on $Q \setminus \{q_{junk}\}$. It is clear that it is reflexive

and symmetric. Also if $q_1 \sim q_2$ and $q_2 \sim q_3$ then $\exists n_1^{'}$ and $n_2^{'}$ corresponding to q_1 and q_2 and $n_1^{''}$ and $n_2^{''}$ corresponding to q_2 and q_3. Due to the properties of τ_s, it must be that $n_2^{'} = n_1^{''}$. Hence $\phi^{'}(q_0^{'}, n_1^{'}) = \phi^{'}(q_0^{'}, n_2^{'}) = \phi^{'}(q_0^{'}, n_1^{''}) = \phi^{'}(q_0^{'}, n_2^{''})$. Thus $q_1 \sim q_3$.

2. if $p \sim q$ then if for some $n_1, n_2 \in A^*$ $\phi(q_0, n_1) = p$ and $\phi(q_0, n_2) = q$ and $\phi^{'}(q_0^{'}, n_1) = \phi^{'}(q_0^{'}, n_2)$ hence $\phi^{'}(q_0^{'}, n_1 a) = \phi^{'}(q_0^{'}, n_2 a)$ and $\phi(p, a) \sim \phi(q, a)$ $\forall a \in dom(\tau_p) \cap dom(\tau_q) \cap A$.

3. If $p \sim q$ then there are n_1 and n_2 as described above. Let $w \in \text{dom}(\tau_p) \cap \text{dom}(\tau_q)$ hence $\phi(p, w) \neq q_{\text{junk}}$ and $\phi(q, w) \neq q_{\text{junk}}$. From the definition of τ_s it follows that $\exists n_1^{'}, n_2^{'} \in A^*$ such that $n_1 w n_1^{'} \in s$ and $n_2 w n_2^{'} \in s$. It follows that $\gamma^{*'}(\phi^{'}(q_0^{'}, n_1), w) = \gamma^{*'}(\phi^{'}(q_0^{'}, n_2), w)$. Since outputs of both transducers must agree on $n_1 w n_1^{'}$ and $n_2 w n_2^{'}$ it follows that $\gamma(p, w) = \gamma(q, w)$. Thus $p \epsilon q$.

Since for each $p \in Q \setminus \{q_{\text{junk}}\}$, $p \in A^*$ and $\phi(q_0, p) = p$, and the word p with this property is unique it follows that the equivalence classes of $\sim$ can be put in one to one correspondence with the subset of states of $\tau^{'}$ as follows: Let each $[q]$ correspond to $\phi^{'}(q_0^{'}, w)$, where w is such that $\phi(q_0, w) = q$.

Now construct an equivalence relation E by adding q_{junk} into any one of the equivalence classes of the relation $\sim$. It follows that E satisfies the premise of the Proposition 3.4. Thus E yields transducer with minimal number of states that recognizes **s**. □

Thus it was shown that, if for a given sequence **s**, τ_s is a transducer that recognizes precisely s with properties given in proposition 2.1 then it is possible to construct a minimal transducer through relation ϵ similarly as it was done in the first section. The main difference is that in this case things are a little more complicated since ϵ is not an equivalence relation.

4 Local picture languages.

4.1 Introduction

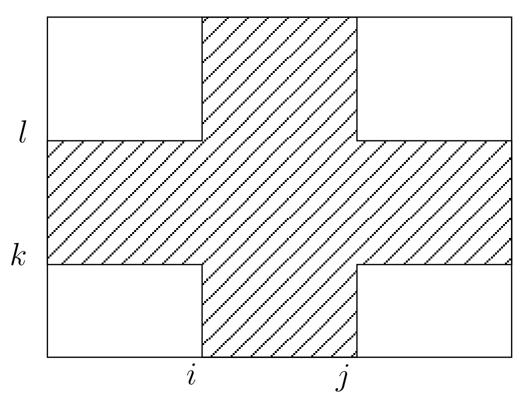
Let $L(\tau)$ denote the input language of the transducer τ and let $L^2(\tau)$ denote the two-dimensional language associated with iterative applications of a given transducer τ to words of $L(\tau)$. In this section we will attempt to analyze the relation between $L^2(\tau)$ and $L(\tau)$ from the point of view of local languages and entropy. We observe that for deterministic transducers, entropy is always equal to zero, however this is not the case if we will consider the two-dimensional language corresponding to a nondeterministic transducers.

4.2 Notation and definitions

Definition 4.1. A *picture language* over alphabet A is a subset of A^{**} (where A^{**} denotes the set of all possible rectangular blocks over alphabet A)

Definition 4.2. A *local picture language L of order k* is a picture language satisfying $B \in L$ if and only if $F_{k,k}(B) \subseteq Q_{k,k}$, where $Q_{k,k}$ is a finite set of $k \times k$ blocks and $F_{k,k}(B)$ denotes the set of all $k \times k$ sub blocks of B.

The notation used in this section is illustrated using the following figure:



The shaded vertical sub block of the depicted block can be described as $B[][i \dots j]$, while $B[k \dots l][]$ stands for a shaded horizontal block. The intersection of $B[k \dots l][]$ and $B[][i \dots j]$ is described by $B[k \dots l][i \dots j]$. The i^{th} column of B is denoted as $B[][i]$, and k^{th} row is $B[k][]$. Note that this notation is also applicable to 1-dimensional words, since they can be considered as blocks of unit height.

If B_1, $B_2 \in A^{**}$ such that B_1 is a block of size $m \times n$ and B_2 is the block of size $m \times k$, then we define the concatenation of B_1 and B_2 to be the block C of size $m \times (n + k)$ such that $C[][1 \dots n] = B_1$ and $C[][n + 1 \dots n + k] = B_2$.

Up to the last section of this chapter only deterministic transducers were considered, in which case we use the following convention: For a transducer τ and $w \in L(\tau)$, let $\Lambda^n_\tau(w)$ denote a $n \times |w|$ block $B \in A^{**}$ with $B[1][] = w$ and $B[k][] = \tau(B[k-1][])$ for $1 < k \leq n$. If for some k, $B[k][]$ is not accepted by τ then $\Lambda^n_\tau(w)$ is undefined. In the last section, involving entropy, non-deterministic transducers will be considered, requiring the notation to be extended. In that case $\Lambda^n_\tau(w)$ will denote a set of all $n \times |w|$ blocks $B \in A^{**}$ with $B[1][] = w$ and $B[k][] \in \tau(B[k-1][])$ for $1 < k \leq n$.

Definition 4.3. A Picture language L is *transducer recognizable* if there exists τ, such that $L = \{\Lambda^n_\tau(w) | w \in L(\tau)\}$. In this case L is denoted by $L^2(\tau)$.

Since we consider deterministic transducers we need to make small adjustments to the definition of local languages in both dimensions. To see the reason behind this consider $L \subseteq A^{**}$, a local picture language defined by some $Q_{3,3}$. For an arbitrary natural number k, let $B \in A^{**}$ be $2 \times k$ block. Since $F_{3,3}(B) = \{\}$, $B \in L$. Of course this situation can never happen in $L^2(\tau)$ with transducer τ being deterministic. This motivated the following definition.

Definition 4.4. We say that L is almost local of order k if and only if the set $L' = \{B \in A^{**} | B \in L \text{ or } F_{k,k}(B) = \{\}\}$ is a local picture language of order k. Similarly, a one dimensional language L is *almost local of order* k if and only if $L' = \{w \in L | w \in L(\tau) \text{ or } |w| < k\}$ is a local language, where k is the minimum of lengths of all allowed words.

From here on $L(\tau)$ or $L^2(\tau)$ are referred to as local if they are local or almost local.

4.3 Local picture languages

In general, if a transducer τ has a local language as its input language then $L^2(\tau)$ does not have to be local. For example, consider the transducer depicted in Figure 4. This transducer is deterministic both on input and output. All of its states are final, and output symbols are depicted inside the states, since they are the same for all of the transitions. The input language of the transducer, $L(\tau)$ is local (no bbb).

Suppose $L^2(\tau)$ is local and, as in the definition, there is a set $Q_{k,k}$ of allowed blocks of size $k \times k$ and $A, B, C \in A^{**}$ are defined as follows:

$$A = \begin{matrix} & & & \dots & & & \\ a & a & a & a & \dots & a & a & a \\ a & a & a & a & \dots & a & a & a \\ a & b & a & a & \dots & a & a & b \end{matrix}$$

$$B = \begin{matrix} & & & \dots & & & \\ a & a & a & a & \dots & a & a & a \\ a & a & a & a & \dots & a & a & b \\ b & a & a & a & \dots & a & b & a \end{matrix}$$

$$C = \begin{matrix} & & & \dots & & & \\ a & a & a & a & \dots & a & a & a \\ a & a & a & a & \dots & a & a & b \\ a & b & a & a & \dots & a & b & a \end{matrix}$$

Note that the blocks $A, B \in L^2(\tau)$ can be extended indefinitely in height and width. Thus if A and B are extended to the $k \times k$ blocks then $A, B \in Q_{k,k}$. Also C could be extended to have length of $k+1$, in which case we get that

$$F_{k,k}(C) = \{A, B\} \subseteq Q_{k,k}$$

Thus it must be that $C \in L^2(\tau)$. However, this is a contradiction, since on input $C[1][]$ (bottom row of block C) the output of τ is different from $C[2][]$. Hence the language is not local.

Suppose that $L^2(\tau)$ is a local picture language. Does this imply that $L(\tau)$ is

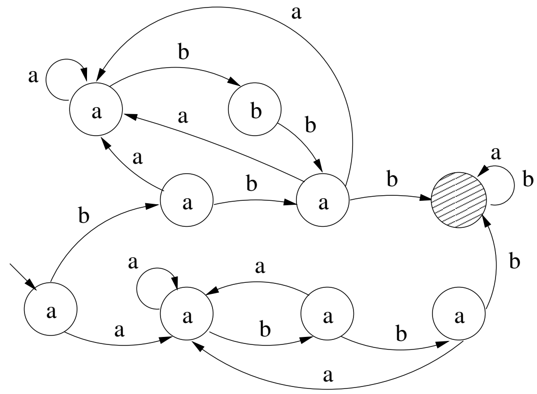


Figure 4: Transducer τ such that $L(\tau)$ is a local, but $L^2(\tau)$ is not

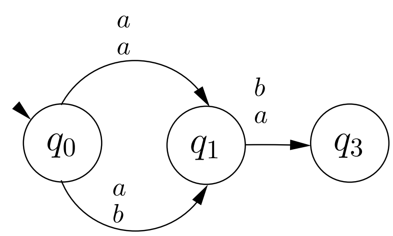


Figure 5: Transducer generating a local picture language

local? If $L^2(\tau)$ is local with $Q_{k,k}$ then the first thing that comes to mind is to consider a local language with set of allowed words equal to the set of all words in $L(\tau)$ of length k. However it may happen that there are blocks B_1 and B_2 in $Q_{k,k}$ with first rows w_1 and w_2 respectively, such that w_1 and w_2 overlap: $w_1[][2 \dots k] = w_2[][1 \dots k-1]$, but B_1 and B_2 do not, i.e. , $B_1[][2 \dots k] \neq B_2[][1 \dots k-1]$.

Example: Let

$$B_1 = \begin{matrix} a & b \\ a & a \end{matrix}$$

and

$$B_2 = \begin{matrix} a & b \\ b & a \end{matrix}$$

and τ be the transducer depicted on Figure 5. Note that $L^2(\tau)$ is local (almost local). In fact, $L^2(\tau) \setminus \{B \in A^{**} | F_{2,2}(B) = \{\}\} = \{B_1, B_2\}$. Also, B_1 and B_2 do not overlap, but $w_1 = ba$, $w_2 = aa$ do.

As the above example shows, we need to enlarge the set of allowed words by including words in $L(\tau)$ of length $k+1$. This way, if the two words of length k overlap as did w_1 and w_2 above, their resulting word, i.e $w = w_1 t$ where $t = w_2[][k]$ will be present in the local language only if $w \in L(\tau)$, which, since $L^2(\tau)$ is local picture language, makes sure that corresponding B_1 and B_2 also overlap. More formally we have the following proposition:

Proposition 4.1. *Let $M = \{w \in L(\tau) | \Lambda_\tau^k(w)$ is undefined$\}$. If $L^2(\tau)$ is a local picture language of order k, then $L(\tau) \setminus M$ must be a local language with the set of allowed words $P = \{w | w \in L(\tau) \setminus M$ and $|w| = k$ or $|w| = k+1\}$.*

Proof. Let $H \subseteq A^{**}$ be a local language with P as the set of allowed words. If $w \in H$ where $|w| = k$, then $w \in L(\tau) \setminus M$ by definition of H. If $|w| > k$ then consider the following construction:

$$B = \Lambda_\tau^k(w[1 \dots k])\Lambda_\tau^k(w[2 \dots k+1])[][1] \dots \Lambda_\tau^k(w[|w| - k \dots |w|])[][1]$$

Since $w[1 \dots k+1] \in L(\tau) \setminus M$, due to the assumptions $\Lambda_\tau^k(w[1 \dots k+1])$ is defined. Since $L^2(\tau)$ is local picture language,

$$F_{k,k}(\Lambda_\tau^k(w[1 \dots k+1]) = \{\Lambda_\tau^k(w[1 \dots k]), \Lambda_\tau^k(w[2 \dots k+1])\}.$$

Thus $B = \Lambda_\tau^k(w[1 \dots k+1]) \dots \Lambda_\tau^k(w[|w| - k \dots |w|])[][1]$. Continuing in this way get that $B = \Lambda^k(w)$.

If $w \in L(\tau) \setminus M$, then $\Lambda^k(w)$ is defined. Hence if u is any factor of w, i.e.

$$F_{k,k}(\Lambda^k(w)[|v| \dots |vu|]) \subseteq F_{k,k}(\Lambda^k(w)) \subseteq Q_{k,k}.$$

Thus if $|u| = k$ then $u \in L(\tau) \setminus M$ and if $|u| = k+1$, $u \in L(\tau) \setminus M$. Hence $w \in H$. □

Corollary 4.1. *If $L^2(\tau)$ is a local picture language and $\tau(L(\tau)) \subseteq L(\tau)$ then $L(\tau)$ is local.*

Proof. Since $\Lambda^k(w)$ is defined for every w, $M = \{w \in L(\tau) | \Lambda^k(w)$ is undefined $\} = \{\}$. □

4.4 Possible applications and future directions.

Intuitively, intrinsically non-local languages generated by transducers can be described as the non-local languages that can be obtained through a nontrivial transducer, i.e something different than, say, a transducer with output function equal to the shift to the right.

Example 1: One of the examples of intrinsically non-local languages is the set of all two-dimensional blocks containing at most one '1', i.e., blocks of the form

$$B = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

It is easy to check that there is no transducer that would produce precisely this language. However it is possible to construct a similar language by adding a padding to a latter one, for example

$$\begin{matrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Such language would be recognizable by transducer depicted on Figure 6.

For a given set of tiles, which can be thought of as a set of rectangles, lets consider the set of all 2 dimensional blocks that could be constructed from the tiles without rotation, and consider the problem of determining if there is a transducer, for which $L^2(\tau)$ consides with that set. Unfortunately, in general this is not possible, for example consider set of two tiles

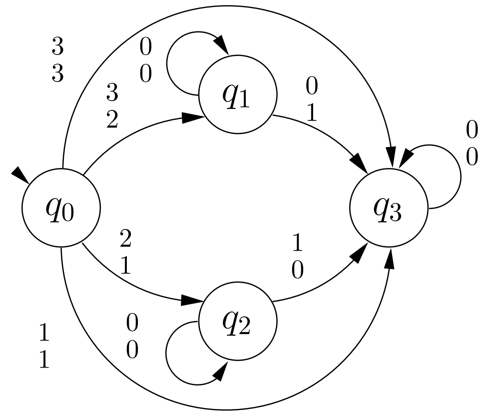
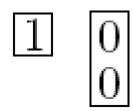


Figure 6: Transducer generating non-local language



and suppose that there is a transducer such that $L^2(\tau)$ equals the set of all possible two dimensional blocks obtained through translations of a given set of tiles. As blocks

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

and

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{array}$$

show (denote these blocks as $C2$ and $C3$ respectively), this is not possible since if such transducer τ would exist, then τ must accept $C3[][2]$ and output word 111111, as it produces one of the valid tilings. However this is impossible, since $C3[][2] = C2[][2]$, and thus τ outputs 111111 as a third row in $C2$, which is not one of the valid tilings.

4.5 Entropy of $L^2(\tau)$

Definition 4.5. For $L \subseteq A^{**}$ the *entropy* of L is

$$h(L) = \lim_{n \to \infty} \sup \frac{1}{n^2} \log(|B_{n,n}|),$$

where $B_{n,n} = \{C \in L | C \text{ is an } n \times n \text{ block }\}$.

It is clear that entropy of $L^2(\tau)$ is 0 for any deterministic transducer τ, since for each n it contains at most $|A|^n$ blocks (to each $w \in L^2(\tau)$ corresponds unique block of height n). However, if $L^2(\tau)$ is a two dimensional language corresponding to a nondeterministic transducer, entropy may no longer be 0. For example, let transducer $\tau = (A, \{q\}, \phi, \{q\}, \{q\})$ with $\phi = \{(q, a, b, q)\,|a, b \in A\}$, i.e. τ consists of one state with all possible transitions. Hence $L^2(\tau) = A^{**}$ and for each n, $|B_{n,n}| = |A|^{n^2}$. Thus $h(L^2(\tau)) = h(A^{**}) = \log |A|$. For a given transducer τ and $w \in L(\tau)$ define $\deg(w) = |\tau(w)|$, the number of distinct words that can be outputted by τ on input w, and extend this notation to sets by $deg(S) = \sup_{s \in S}\{deg(s)\}$. Then the number of distinct $n \times n$ blocks contained in $L^2(\tau)$ with w ($|w| = n$) as a first row is bounded above by the following expression:

$$\Lambda^n(w) \leq \prod_{i=0}^{n-2}(deg(\tau^i(w)).$$

If for all $w \in L(\tau)$ with $|w| = n$ have that $deg(w) \leq n^k$ then

$$\begin{aligned} h(L^2(\tau)) &= \lim_{n\to\infty} \sup \frac{1}{n^2} \log(B_{n,n}) \\ &\leq \lim_{n\to\infty} \sup \frac{1}{n^2} \log(\sum_{|w|=n} \prod_{i=0}^{n-2}(deg(\tau^i(w)))) \\ &\leq \lim_{n\to\infty} \sup \frac{1}{n} \log(|A|n^k) = 0. \end{aligned}$$

Thus if there exists k such that for any $w \in L(\tau)$, $\tau(w) \leq |w|^k$, i.e. $|\tau(w)|$ has a polynomial bound, the entropy is still equal to zero.

Proposition 4.2. *For any regular language L there is τ such that $L(\tau) = L$ and $h(L^2(\tau)) = h(L(\tau))$*

Proof. Let M denote a FSA, such that $L(M) = L$, and let ϕ be its transition relation. Construct transducer τ from M by redefining transition relation as $\phi' = \{(q_1, a, s, q_2)|(q_1, a, q_2) \in \phi, s \in A\}$. Thus $L(\tau) = L$.

Now consider $L^2(\tau)$. Note that for $B \in L^2(\tau)$, the last, k^{th} row may not be in $L(\tau)$.

Let $B_{k,k}$ denote the set of all two dimensional blocks in $L^2(\tau)$ of size $k \times k$ and $B'_{p,k}$ denote the set of all $p \times k$ blocks $B \in L^2(\tau)$ with $B[p][] \in L(\tau)$ ($B[p][]$ denotes p^{th} row of block B). Thus it must be true that $|B_{k,k}| \leq |B'_{k-1,k}||A|^k \leq |B'_{k,k}||A|^k$. Hence,

$$h(L^2(\tau)) = \limsup_{k\to\infty} \frac{1}{k^2} \log |B_{k,k}| \tag{4.1}$$
$$\leq \limsup_{k\to\infty} \frac{1}{k^2} \log(|B'_{k,k}||A|^k) \tag{4.2}$$
$$= \limsup_{k\to\infty} \frac{1}{k^2} \log |B'_{k,k}| \tag{4.3}$$

Now, since $|B'_{k,k}| \leq |B_{k,k}|$, have that

$$h(L^2(\tau)) = \limsup_{k\to\infty} \frac{1}{k^2} \log(|B'_{k,k}|)$$

Also, $|B'_{k,k}| = |L(\tau) \cap A^k|^k$ since $B'_{k,k} = \{B \in A^{**} | B[i][] \in L(\tau) \cap A^k, i = 1 \dots k\}$. Thus

$$h(L^2(\tau)) = \limsup_{k\to\infty} \frac{1}{k} \log(|L(\tau) \cap A^k|) = h(L)$$

□

Corollary 4.2. *Let L be a regular language. Then the entropy of L is an upper bound for the entropy of $L^2(\tau)$ for any transducer τ with $L(\tau) = L$.*

Proof. Let $L \in Reg$ and let $\tau = (A, Q, \phi, q_0, F)$ be such that $L(\tau) = L$. Then consider $\tau' = (A, Q, \phi', q_0, F)$, with $\phi' = \{(q, a, s, p) | s \in A$ and $(q, a, t, p) \in \phi$, for some $t \in A\}$. Hence $\phi \subseteq \phi'$ and $h(L^2(\tau)) \leq h(L^2(\tau'))$. By the proof of the proposition 4.2, $h(L^2(\tau')) = h(L(\tau))$ and claim follows. □

5 Conclusion

In this work we have attempted to analyze sequences that can be obtained by iterating a transducer on words of its input language. We called such sequences transducer recognizable. For the sequences that are not transducer recognizable we have developed a notion of recognition in context and shown that all finite and periodic sequences are recognized in context. As noted in the introduction, transducer recognizable sequences can be related to the blocks constructed of Wang tiles. Each of the Wang tiles corresponds to a specific transition in the transducer. When the transducer is deterministic, each tile is uniquely determined by its left and bottom edge. This is a beneficial property when process of assembly of Wang tiles in blocks is simulated by DNA strands, as it helps to reduce the number of incorrect partial tilings [13].

To each nondeterministic transducer we associated a two dimensional language $L^2(\tau)$. In the case when $L^2(\tau)$ is a local picture language, we investigated properties of $L(\tau)$ that this condition induces. We have shown a relation between the entropy of $L^2(\tau)$ and $L(\tau)$. This relation suggests a wide range of the possible values for the entropy of $L^2(\tau)$.

Characterization of patterns that can be generated by iteration of transducers may be of interest in applications, in particular, in algorithmic self-assembly of two-dimensional arrays with DNA tiles. Also, it may be of interest to investigate some decidability questions for these languages as well: for example, given a transducer generated language, is there an $m \times n$-block in the language for every m, n? The relationship of the class of transducer generated languages with the class of unambiguous (or non-deterministic) picture languages as well as transitivity and mixing properties of these languages remain to be investigated as well.

References

[1] L. M. Adleman, Molecular Computation of Solutions to Combinatorial Problems, *Science, 266(1994), 1021-1024.*

[2] A. V. Aho, J. D. Ullman, The theory of parsing, translation and compiling. Volume I, *Prentice-Hall, 1972.*

[3] B. Chakraborty, N. Jonoska, N.C. Seeman, Programmable tranducer by DNA self-assembly, *submitted*.

[4] P. J. Denning, Machines, Languages, and Computation. *Prentice-Hall, 1978*

[5] E. Dolzhenko, N. Jonoska, On Complexity of Two Dimensional Languages Generated by Transducers, *submitted.*

[6] B. Grunbaum, G.C. Shephard, Tilings and patterns. *W. H. Freeman and Company, 1987.*

[7] N. Jonoska, et. al., Transducers with Programmable Input by DNA Self-assembly,*Lecture Notes in Computer Science, Springer 2950(2004), 219-240.*

[8] N. Jonoska and J.B. Pirnot, Transitivity in Two-Dimensional Local Languages Defined by Dot Systems. *International Journal of Foundations of Computer Science, 17(2006), 435-464.*

[9] V. Manca, C. Martin-Vide, Gh. Păun, New computing paradigms suggested by DNA computing: computing by carving, *BioSystems 52(1999) 47-54.*

[10] Gh. Păun, On the iteration of gsm mappings, *Rev. Roum. Math. Pures Appl.,*

23(1978), 921-937.

[11] Rozenberg, Salomaa (Eds). Handbook of Formal Languages: Volume 3. Beyond Words. *2002.*

[12] H. Wang, Notes on class tiling problems. *Fundamental Mathematics, 82(1975), 295-305.*

[13] E. Winfree, Algorithmic Self-Assembly of DNA: Theoretical Motivations and 2D Assembly Experiments. *Journal of Biomolecular Structure and Dynamics,11 (S2) (2000), 263-270.*

[14] E. Winfree, F. Liu, L.A. Wenzler, N.C. Seeman, Design and self-assembly of two-dimensional DNA crystals, *Nature, 394(1998), 539-544*.