УДК 519.717

On the minimization and equivalence checking of sequential reactive systems

Temerbekova G.G. (Lomonosov Moscow State University)

Zakharov V.A.

(National Research University Higher School of Economics (HSE))

Finite state transducers over semigroups can be regarded as a formal model of sequential reactive programs. In some cases verification of such programs can be reduced to minimization and equivalence checking problems for this model of computation. To solve efficiently these problems certain requirements are imposed on a semigroup these transducers operate on. Minimization of a transducer over a semigroup is performed in three stages: at first the greatest common left-divisors are computed for all states of a transducer, next a transducer is brought to a reduced form by pulling all such divisors "upstream", and finally a minimization algorithm for finite state automata is applied to the reduced transducer. As a byproduct of this minimization technique we obtain an equivalence checking procedure for transducers operating on certain classes of semigroups.

Keywords: reactive system, transducer, semigroup, minimization, equivalence checking

1. Introduction

Finite state transducers extend finite state automata to model functions on strings or lists. That is why they are used in fields as diverse as computational linguistics [9] and model-based testing [1, 18]. In software engineering transducers provide a suitable formal model for various device drivers for manipulating with strings, transforming images, filtering dataflows. Transducers also found a usage in regular model checking of parameterized distributed systems. In some formal models of these systems configurations are modeled as words over finite alphabet and a transition relation is specified by a finite state transducers [21]. The more succinct is the presentation of these transducers, the more efficient are regular model checking algorithms. The authors of [17] proposed models of communication protocols as regular transducers operating on bit strings and set up the verification problem as equivalence checking between the protocol transducer and the specification transducer. These considerations show that algorithms for building compositions of transducers, checking equivalence, reducing their state space considerably enhance the effectiveness of designing, verification and maintenance of software routines.

Transducers can take on the role of simple models of sequential reactive programs. These programs operate in the interaction with the environment permanently receiving data (requests) from it. At receiving a piece of data such program performs a sequence of actions. When certain control points are achieved a program outputs the current results of computation as a response. Since different sequences of actions may yield the same result we need a more sensitive interpretation of the outputs than just words in some alphabet. Basic actions of a program are interpreted as generating elements of an appropriate semigroup, and the result of computation is a composition of actions performed by the program.

Imagine, for example, that a radio-controlled robot moves on the earth surface. It can make one step moves in any of 4 directions N, E, S, W. When such robot receives a control signal syg in a state q it must choose and carry out a sequence of steps (say, N, N, W, S), and enter to the next state q'. At some distinguished state q_{fin} robot reports its current location. The most simple model of computation which is suitable for designing such a robot and analyzing its behaviour is non-deterministic finite state transducer operating on free Abelian group of rank 2. These considerations give rise to the concept of a transducer which has some finitely generated semigroup S for the set of outputs.

In this paper we study minimization and equivalence checking problems for finite state transducers operating on certain semigroups. The study of these problems for classical transducers over words began in the early 60s. First, it was shown that the equivalence checking problem is undecidable for non-deterministic transducers [8]. But the undecidability displays itself only in the case of unbounded transduction when an input word may have arbitrary many images. At the next stage bound-valued transducers were studied. The equivalence checking problem was shown to be decidable for deterministic [3], functional transducers [2, 15], and k-valued transducers [5, 20]. In a series of papers [12, 13, 16] a construction to decompose k-valued transducers into a sum of functional and unambiguous ones was developed and used for checking k-valuedness and equivalence of finite state transducers over words. An alternative approach which is applicable to a more wide class of transducers was introduced in [23]. It was shown that the equivalence checking problem is decidable for k-valued transducers operating on any semigroup S which is embeddable in a decidable group.

The minimization problem for finite state transducers over words was considered in [11], but only in [10] an admissible solution to this problem was obtained. Later a minimization algorithm proposed by Mohri was corrected and improved in [4, 14]. In [7] an attempt was made to adapt this result to weighted transducers. An alternative approach to this problem was suggested in [24]: it was shown that minimization of finite state transducers operating on decidable groups can be achieved through the using of equivalence checking algorithms developed in [23].

In this paper a minimization technique proposed by M. Mohri [10] is extended to cover the case of finite state transducers operating on ordered semigroups. Minimization of a transducer π over a semigroup S is performed in three stages. At the first stage for every state q we compute the greatest common left-divisor $GCD(\pi, q)$ of all those elements of S that are the results of the runs of π beginning in q. At the next stage we pull all $GCD(\pi, q)$ ”upstream” to obtain such a transducer π' that $GCD(\pi', q)$ for every state q is the neutral element of S. A transducer enjoying this property can be minimized by considering its underlying finite automaton only and by applying any minimization algorithm for finite automata (see, e.g. [19]). As a byproduct of this minimization technique we obtain also an equivalence checking procedure for transducers operating on S.

This work is supported by the Basic Research Program at the National Research University Higher School of Economics in 2016 and by RFBR grants N 16-01-00546.

2. Transducers as models of sequential reactive systems

Let $\mathcal{C}$ and $\mathcal{A}$ be two finite sets. The elements of $\mathcal{C}$ are called *signals*; they may be viewed as abstractions of messages (control instructions, instrument readings, etc.) received by a reactive system from its environment. Finite sequences of signals (words over alphabet $\mathcal{C}$) are called *signal flows*. As usual, the set of signal flows is denoted by $\mathcal{C}^*$. We write uv for concatenation of signal flows u and v.

The elements of $\mathcal{A}$ are called *basic actions*; they may be viewed as abstractions of operations (data processings, movements, etc.) performed by a reactive system in response to received messages. Finite sequences of basic actions (words over $\mathcal{A}$) are called *compound actions*.

Actions are interpreted over semigroups. Consider a semigroup $(S, e, \circ)$ generated by the set $\mathcal{A}$, where e is the neutral element, and $\circ$ is a composition operation. The elements of S may be regarded as *data states*. Every basic action $a, a \in \mathcal{A}$, always terminates and when been applied to a data state $s, s \in S$, yields the result $s \circ a$. Every compound action $g = a_1 a_2 \dots a_k$ is interpreted as the composition $a_1 \circ a_2 \circ \cdots \circ a_k$. In order to distinguish a compound action g from its interpretation we denote the latter by $[g]_S$ and skip the index S when a semigroup is clearly assumed from the context.

A deterministic *finite state transducer* over a set of signals $\mathcal{C}$ and a set of basic actions $\mathcal{A}$ is a labeled transition system $\pi = (\mathcal{C}, \mathcal{A}, Q, q_0, F, T, g_0)$, where Q is a finite set of *control states*, $q_0, q_0 \in Q$, is an *initial state*, $F, F \subseteq Q$ is a subset of *output states*, $T, T : Q \times \mathcal{C} \to Q \times \mathcal{A}^*$, is a *transition function*, and $g_0, g_0 \in \mathcal{A}^*$, is an *initializing action.* Every quadruple (q, c, q', g) such that $T(q, c) = (q', g)$ is called a *transition* which is depicted as $q \xrightarrow{c,g} q'$. By the size $|\pi|$ of a transducer π we mean the number $|Q|$ of its control states.

A *run* of π on a signal flow $w = c_1c_2 \dots c_n$ is a sequence of transitions

$$q \xrightarrow{c_1,g_1} q_1 \xrightarrow{c_2,g_2} q_2 \xrightarrow{c_3,g_3} \cdots \xrightarrow{c_n,g_n} q' \ . \tag{1}$$

We denote this run by $q \xrightarrow{w,h}_* q'$, where $h = g_1g_2 \dots g_n$. If q is the initial state then the run is called *initial*, and if $q' \in F$ then the run is called *output.* If a run is both initial and output then it is called *complete.* When $q \xrightarrow{w,h}_* q$ is a complete run of π and the actions are interpreted over a semigroup S the element $[g_0h]$ is called the *result* of the run.

Finite state transducers can be used as formal models of sequential reactive systems. At the beginning of the computation a reactive system executes an initializing action g_0. At each step of its computation it receives a signal c from the environment and performs a transition $q \xrightarrow{c,g} q'$ by passing its control to a state q' and executing an action g. When a system turns out to be in an output state it displays an achieved result of its computation to an outside observer and continues its interaction with the environment. A behaviour of such a reactive system is completely specified by a partial function $\pi : \mathcal{C}^* \to S$ such that

$$\pi(w) \quad = \quad \begin{cases} [g_0h], \ \text{if there exists a complete run } q \xrightarrow{w,h}_* q' \text{ of } \pi, \\ \text{undefined, otherwise,} \end{cases}$$

for every signal flow w.

Transducers π_1 and π_2 are S-*equivalent* ($\pi_1 \sim_S \pi_2$ in symbols) iff $\pi_1(w) = \pi_2(w)$ holds for every signal flow w. A transducer π' is called S-*minimal* if $|\pi'| \le |\pi|$ holds for any S-equivalent transducer π. The minimization problem for transducer over a semigroup S is to build, given an arbitrary transducer π, a S-minimal transducer π' such that $\pi' \sim_S \pi$.

With every transducer $\pi = (\mathcal{C}, \mathcal{A}, Q, q_0, F, T, g_0)$ operating on a semigroup S one can associate a deterministic finite state automaton $A_\pi = (\mathcal{C} \times S, Q, q_0, F, \varphi)$ over a (possibly infinite) alphabet of pairs $\mathcal{C} \times S$; its transition function $\varphi : Q \times (\mathcal{C} \times S) \to Q$ is specified as follows: $\varphi(q, (c, s)) = q' \iff T(q, c) = (q', g) \wedge s = [g]$. Such an automaton takes at its input a finite sequence of pairs $\alpha = (c_1, s_1), (c_2, s_2), \dots, (c_n, s_n)$ and accepts it at reaching an output state q'. Clearly, A_π accepts α iff the transducer π has a complete run (1) such that $[g_i] = c_i$ for every $i, 1 \le i \le n$. Let $L(A_\pi)$ be the set of all sequences $\alpha, \alpha \in (\mathcal{C} \times S)^*$, accepted by A_π. Transducers π' and π'' are *strongly equivalent* on a semigroup S (in symbols $\pi' \approx_S \pi''$) iff $L(A'_\pi) = L(A''_\pi)$.

It is easy to see that if transducers π' and π'' have the same initializing action (i.e. $[g'_0] = [g''_0]$) and $\pi' \approx_S \pi''$ then $\pi_1 \sim_S \pi_2$. In general case the converse is not true. The key idea of our minimization technique is that of finding, given a certain semigroup S, a subclass of reduced transducers such that

1. for every transducer π one can effectively construct a reduced S-equivalent transducer π' such that $|\pi| = |\pi'|$,
2. for any pair of reduced transducers π' and π'' it is true that $\pi' \sim_S \pi''$ iff $\pi' \approx_S \pi''$ and $[g'_0] = [g''_0]$.

Then to minimize a transducer π one needs only to build an equivalent reduced transducer π' and then apply any of the well-known techniques [19] for minimization of a deterministic finite state automaton $A_{\pi'}$. This approach can be used also for equivalence checking of finite state transducers operating on certain semigroups: to check whether $\pi_1 \sim_S \pi_2$ it is sufficient to build S-equivalent reduced transducers π'_1 and π''_2 and then check the equivalence of deterministic finite state automata $A_{\pi'_1}$ and $A_{\pi'_2}$.

3. Ordered semigroups

In this section we will impose certain requirements on a semigroup S to solve efficiently the minimization problem for transducers operating on such a semigroup.

Let a binary relation $\preceq_S$ on S be defined as follows: $s_1 \preceq_S s_2 \iff \exists s : s_1 \circ s = s_2$. A semigroup S is called *ordered* iff $(S, \preceq_S)$ is a partially ordered set. Sometimes we will skip the underscore symbol S if it is clear from the context. Our first requirement is

Req1: $(S, \preceq)$ is a well-founded lattice such that the greatest lower bound is effectively computable for every pair of elements $[h]$ and $[g]$, where $g, h \in \mathcal{A}^*$.

Denote by $s_1 \vee s_2$ and $s_1 \wedge s_2$ the greatest lower bound and the least upper bound of elements s_1 and s_2 respectively. Actually, $s_1 \vee s_2$ is the greatest common left-divisor of s_1 and s_2, and $s_1 \wedge s_2$ is the lowest common multiple of s_1 and s_2. From the definition of $\preceq$ it follows that $s \circ s_1 \vee s \circ s_2 = s \circ (s_1 \vee s_2)$. The neutral element e of S is the least element in $(S, \preceq)$ but this lattice may have no maximal elements. We add to S a new virtual element τ such that

$s \circ \tau = \tau \circ s = \tau$ holds for any element s in S. Clearly, $s \preceq \tau$ holds for every $s, s \in S$. Let $S_\tau = S \cup \{\tau\}$ Thus, if S meets the requirement **Req1** then $(S_\tau, \preceq)$ is a complete lattice. For any subset S' of S_τ we write $\bigvee S'$ for the greatest lower bound of S'.

Req2: There exists an algorithm for solving equations of the form $[g] \circ X = [h]$ for every pair of actions $g, h \in \mathcal{A}^*$.

It is easy to see that if a semigroup S satisfies the requirements **Req1** and **Req2** then the word problem "$[g] \overset{?}{=} [h]$" is decidable in S.

A semigroup S is called *left cancellative* iff $s \circ s' = s \circ s'' \Rightarrow s' = s''$ holds for every triple of elements s, s', s''. Our final requirement is

Req3: S is a left cancellative semigroup.

Many semigroups widely used in computer science, including free monoids, partially commutative monoids (traces) [6], a semigroup of conservative substitutions [22], etc. meet the requirements **Req1–Req3** listed above.

4. Greatest common divisors

Our minimization algorithm comprises three stages. At the first stage it figures out for every control state q the greatest common divisor of all results of all output runs that start in q.

A control state q of transducer π is *useful* if it is traversed by at least one complete run. It easy to see that useless states do not affect the function $\pi(\cdot)$ and by deleting all useless states with the incoming and outcoming transition we obtain an equivalent transducer π'. We will assume without loss of generality that all control states of transducers are useful.

Let π be a finite state transducer such that $Q = \{q_1, q_2, \ldots, q_n\}$. Consider an arbitrary control state q_i of a transducer π and a set

$$S(\pi, q_i) = \{[h] : q_i \xrightarrow{w,h}_* q_j, q_j \in F\} \ .$$

of results computed by the output runs started in the state q_i. We say that the element $gcd(\pi, q_i) = \bigvee S(\pi, q_i)$ is the *greatest common divisor* of the state q_i and use a notation $GCD(\pi)$ for the tuple $\langle gcd(\pi, q_1), gcd(\pi, q_2), \ldots, gcd(\pi, q_n)\rangle$.

To compute the greatest common divisors of all control states of π we introduce an operator $\Psi_\pi : S_\tau^n \to S_\tau^n$ as follows. For every tuple $\langle s_1, s_2, \ldots, s_n\rangle$ in S_τ^n we assume that

$\Psi_\pi(s_1, s_2, \dots, s_n) = \langle s'_1, s'_2, \dots, s'_n \rangle$, where

$$s'_i \; = \; \begin{cases} e, \text{ if } q_i \text{ is an output state,} \\ \bigvee\{[g] \circ s_j : T(q_i, c) = (q_j, g), c \in \mathcal{C}\}, \text{ otherwise,} \end{cases}$$

for every $i, 1 \le i \le n$.

The partial order $\preceq$ can be extended on the set of tuples S^n_τ in the usual way:

$$\langle s_1, s_2, \dots, s_n \rangle \preceq \langle s'_1, s'_2, \dots, s'_n \rangle \iff \forall i : s_i \preceq s'_i \; .$$

Proposition 1. If a semigroup S meets the requirement **Req1** then Ψ_π is monotone operator.

This proposition follows immediately from the definition of Ψ. Since S_τ is a complete lattice, the operator Ψ_π by Knaster-Tarsky theorem has the greatest fixed point $gfp(\Psi_\pi)$. By Kleene theorem the greatest fixed point of Ψ_π is the limit of the descending sequence $\top \succeq_S \Psi_\pi(\top) \succeq_S \Psi_\pi(\Psi_\pi(\top)) \succeq_S \dots$, where $\top = \langle \tau, \tau, \dots, \tau \rangle$. Since by **Req1** the partially ordered set $(S, \preceq)$ is well-founded, $\Psi^k_\pi(\top) = \Psi^{k+1}_\pi(\top) = gfp(\Psi_\pi)$ holds eventually for some k.

Proposition 2. If a semigroup S meets the requirement **Req1** then $gfp(\Psi_\pi) = GCD(\pi)$.

Proof. 1). If $q_i \in F$ then $e \in S(\pi, q_i)$ and, hence, $gcd(\pi, q_i) = e$. If q_i is not an output control state then $S(\pi, q_i) = \bigcup\limits_{c \in \mathcal{C}} \{[g] \circ [h] : T(q_i, c) = (q_j, g), h \in S(\pi, q_j)\}$. Therefore,

$$gcd(\pi, q_i) = \bigvee\{[g] \circ GCD(\pi, q_j) : T(q_i, c) = (q_j, g), c \in \mathcal{C}\}$$

by left-distributivity of $\circ$ over $\vee$. Hence, $GCD(\pi)$ is a fixed point of the operator Ψ_π.

2). Suppose that $gfp(\Psi_\pi) = \langle s'_1, s'_2, \dots, s'_n \rangle$ and $q_i \xrightarrow{w,h}_* q_j$ is an arbitrary output run of π. It could be shown by induction on the length of this run that $s'_i \preceq [h]$. If $s_i = s_j$ then by definition of Ψ_π we have $s'_i = e \preceq [h]$ for any action h. Consider a case of a run $q_i \xrightarrow{w,h}_* q_j = q_i \xrightarrow{c,g} q_k \xrightarrow{w',h'}_* q_j$. Then by induction hypothesis and by definition of Ψ_π we have $s'_i \preceq [g] \circ s'_k \preceq [g] \circ [h'] = [h]$. Therefore, $s'_i \preceq [h]$ holds for every $[h]$ in $S(\pi, q_i)$. This implies that $s'_i \preceq gcd(\pi, q_i)$ for every $i, 1 \le i \le n$. Thus, $gfp(\Psi_\pi) \le GCD(\pi)$. □

5. Reduced transducers

At the next stage our minimization algorithm brings a finite state transducer to a reduced form. We say that a transducer π operating on a semigroup S which satisfies **Req1** is *reduced* iff $GCD(\pi) = \langle e, e, \dots, e \rangle$.

Theorem 1. If a semigroup S meets requirements **Req1–Req2** then every transducer π can be effectively transformed into a reduced transducer π' such that $\pi \sim_S \pi'$ and $|\pi| = |\pi'|$.

Proof. Suppose that q_1 is the initial control state of π. For an arbitrary transition $q_i \xrightarrow{c,g} q_j$ in π consider an equation $gcd(\pi, q_i) \circ X = [g] \circ gcd(q_j)$. Since $S(\pi, q_i) \supseteq \{[g] \circ s : s \in S(\pi, q_j)\}$, by definition of $GCD(\pi)$ we have $gcd(\pi, q_i) \preceq [g] \circ gcd(\pi, q_j)$. Hence, the equation above always has a solution. By **Req2** this solution $X = g'$ can be computed effectively. A transducer π' is obtained from π by the replacement of every transition $q_i \xrightarrow{c,g} q_j$ with a transition $q_i \xrightarrow{c,g'} q_j$ and by the replacement of the initializing action g_0 of π with an initializing action g_0' such that $[g_0'] = [g_0] \circ gcd(\pi, q_1)$.

The relationship $gcd(\pi, q_i) \circ g' = g \circ gcd(\pi, q_j)$ between transitions $q_i \xrightarrow{c,g} q_j$ and $q_i \xrightarrow{c,g'} q_j$ in transducers π and π' can be extended to the runs of these transducers. Consider an arbitrary pair of corresponding runs of π and π' on some signal flow $w = c_1 c_2 \dots c_{m-1} c_m$:

$$q_1 \xrightarrow{c_1,g_1} q_2 \xrightarrow{c_2,g_2} \cdots q_{m-1} \xrightarrow{c_{m-1},g_{m-1}} q_m \xrightarrow{c_m,g_m} q_{m+1},$$
$$q_1 \xrightarrow{c_1,g_1'} q_2 \xrightarrow{c_2,g_2'} \cdots q_{m-1} \xrightarrow{c_{m-1},g_{m-1}'} q_m \xrightarrow{c_m,g_m'} q_{m+1}.$$

Then by definition of π' we have the following chain of equalities:

$$\begin{aligned}
&[g_1 \dots g_{m-1} g_m] \circ gcd(\pi, q_m) = [g_1 g_2 \dots g_{m-1}] \circ [g_m] \circ [gcd(\pi, q_{m+1})] = \\
&= [g_1 g_2 \dots g_{m-1}] \circ [gcd(\pi, q_m)] \circ [g_m'] = [g_1 g_2 \dots g_{m-2}] \circ gcd(\pi, q_{m-1}) \circ [g_{m-1}' g_m'] = \cdots \\
&\cdots = [g_1] \circ [gcd(\pi, q_2)] \circ [g_2' \dots g_{m-1}' g_m] = gcd(\pi, q_1) \circ [g_1' \dots g_{m-1}' g_m'] \ .
\end{aligned}$$

To make sure that $\pi \sim_S \pi'$ it should be noticed first that both functions $\pi(\cdot)$ and $\pi'(\cdot)$ have the same domain. Consider then an arbitrary signal flow w such that $\pi(w)$ is defined. Let $q_1 \xrightarrow{w,h}_* q_m$ and $q_1 \xrightarrow{w,h'}_* q_m$ be complete runs of π and π' on w. Since $gcd(\pi, q_m) = e$ due to $q_m \in F$, the following chain of equalities holds:

$$\pi(w) = [g_0] \circ [h] = [g_0] \circ [h] \circ [gcd(\pi, q_m)] = [g_0] \circ [gcd(\pi, q_1)] \circ [h'] = [g_0'] \circ [h'] = \pi'(w) \ .$$

Hence, $\pi(w) = \pi'(w)$ for every signal flow w.

To make certain that π' is a reduced transducer consider an arbitrary control state q_i in π' (which is also a control state in π) and $gcd(\pi', q) = \bigvee\{[h'] : q_i \xrightarrow{w,h'}_* q, q \in F\}$. Relying on the relationship between the corresponding runs of transducers π and π' and on the fact that $gcd(\pi, q) = e$ holds for any final state q it is easy to notice that

$$\begin{aligned}
&gcd(\pi, q_i) \circ gcd(\pi', q_i) = \bigvee\{gcd(\pi, q_i) \circ [h'] : q_i \xrightarrow{w,h'}_* q, q \in F\} = \\
&= \bigvee\{[h] \circ gcd(\pi, q) : q_i \xrightarrow{w,h}_* q, q \in F\} = gcd(\pi, q) \ .
\end{aligned}$$

Since S is an ordered semigroup, $gcd(\pi, q_i) \circ gcd(\pi', q_i) = gcd(\pi, q)$ implies $gcd(\pi', q_i) = e$. □

6. Minimization of reduced transducers

At the final stage to minimize reduced transducers we apply any of minimization techniques for deterministic finite state automata. This consideration is based on the close relationships between reduced transducers and finite state automata revealed in the propositions below.

Proposition 3. Suppose that a semigroup S meets the requirements **Req1, Req3**. Let π' and π'' be a pair of reduced S-equivalent transducers such that g_0' and g_0'' are their initializing actions. Then $[g_0'] = [g_0'']$.

Proof. Consider an arbitrary pair $q_0' \xrightarrow{w,h'}_* q'$ and $q_0'' \xrightarrow{w,h''}_* q''$ of complete runs of π' and π'' on some signal flow w. Since $\pi' \sim_S \pi''$, it is true that $[g_0'h'] = [g_0''h'']$. The latter means that $[g_0'] \wedge [g_0''] \neq \tau$. Hence, by definition of the least upper bound in the lattice $(S, \preceq)$ there exists a triple of elements s, s', s'' such that $[g_0'] \wedge [g_0''] = [g_0'] \circ s' = [g_0''] \circ s''$ and $[g_0'h'] = [g_0''h''] = ([g_0'] \wedge [g_0'']) \circ s$. Thus, $[g_0'rc[h'] = [g_0'] \circ s' \circ s$ and $[g_0''h''] = [g_0''] \circ s'' \circ s$. Since S satisfies the requirement **Req3**, these equalities imply $[h'] = s' \circ s$ and $[h''] = s'' \circ s$. It should be noticed that the elements s' and s'' depend on g_0' and g_0'' only. Therefore, the conclusion can be made that $s' \preceq gcd(\pi', q_0')$ and $s'' \preceq gcd(\pi'', q_0'')$. But once π' and π'' are reduced transducers, we have $gcd(\pi', q_0') = gcd(\pi'', q_0'') = e$. Hence, $s' = s'' = e$. This means that $g_0' = g_0''$ □

Proposition 4. Suppose that a semigroup S meets the requirements **Req1, Req3** and let π' and π'' be a pair of reduced S-equivalent transducers. Suppose also that $q_0' \xrightarrow{w,h'}_* q_1' \xrightarrow{c,g'} q_2'$ and $q_0'' \xrightarrow{w,h''}_* q_1'' \xrightarrow{c,g''} q_2''$ are initial runs of π' and π'' on some signal flow wc, where $c \in \mathcal{C}$. Then $[g'] = [g'']$.

The proof of Proposition 4 follows the same line of reasoning as that of Proposition 3. These propositions bring us to

Theorem 2. If a semigroup S satisfies the requirements **Req1, Req3** then for any pair of reduced transducers π' and π'' it is true that

$$\pi' \sim_S \pi'' \iff \pi' \approx_S \pi'' \wedge [g_0'] = [g_0''],$$

where g_0' and g_0'' are initializing actions of π' and π''.

Theorems 1 and 2 provide a solution to both minimization problem and equivalence checking problem for deterministic finite state transducers operating on a semigroup S which satisfies the requirements **Req1–Req3**. To verify the S-equivalence of transducers π_1 and π_2 it is sufficient to minimize both transducers and then check that these S-minimal transducers are isomorphic.

7. Conclusions

Complexity issues of the minimization problem for finite state transducers over semigroups that fall into the scope of requirements **Req1–Req3** is a topic for further research since the complexity depends greatly on the individual algebraic properties of a lattice $(S, \preceq)$.

One may also wonder how much important for minimization problem are the requirements **Req1–Req3**. Some ordered semigroups of actions arising in program modeling are not left-cancellative, and their lattices $(S, \preceq)$ are not well-founded. It would be interesting to study to build effectively S-minimal transducers for such semigroups.

References

1. Alur R., Cerny P. Streaming transducers for algorithmic verification of single-pass list-processing programs // Proc. of 38-th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. 2011. p. 599-610.
2. Blattner M, Head T. Single-valued transducers // Journal of Computer and System Sciences. 1977. Vol. 15. p. 310-327.
3. Blattner M, Head T. The decidability of equivalence for deterministic finite transducers // Journal of Computer and System Sciences. 1979. Vol. 19. p. 45-49.
4. Beal M.-P., Carton O. Computing the prefix of an automaton // Theoretical Informatics and Applications. 2000. Vol. 34. p. 503-514.
5. Culik K., Karhumaki J. The equivalence of finite-valued transducers (on HDTOL languages) is decidable // Theoretical Computer Science. 1986. Vol. 47. p. 71-84.
6. Diekert V., Metivier Y. Partial commutation and traces // Handbook of formal languages. 1997. Vol. 3. p. 457-533.
7. Eisner J. Simpler and more general minimization for weighted finite-state automata // Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. 2003. Vol. 1. p. 64-71.
8. Griffiths T. The unsolvability of the equivalence problem for ε-free nondeterministic generalized machines // Journal of the ACM. 1968. Vol. 15. p. 409-413.
9. Mohri M. Finite-state transducers in language and speech processing // Computational Linguistics. 1997. Vol. 23. p. 269-311.
10. Mohri M. Minimization algorithms for sequential transducers // Theoretical Computer Science. 2000. Vol. 234. p. 177-201.

11. Reutenauer C., Schuzenberger M.P. Minimization of rational word functions // SIAM Journal of Computing. 1991. Vol. 30. p. 669-685.
12. Sakarovitch J., de Souza R. On the decomposition of k-valued rational relations // Proc. of 25th International Symposium on Theoretical Aspects of Computer Science. 2008. p.621-632.
13. Sakarovitch J., de Souza R. On the decidability of bounded valuedness for transducers // Proc. of the 33rd International Symposium on MFCS. 2008. p. 588-600.
14. Shofrutt C. Minimizing subsequential transducers: a survey // Theoretical Computer Science. 2003. Vol. 292. p. 131-143.
15. Schutzenberger M. P. Sur les relations rationnelles // Proc. of Conference on Automata Theory and Formal Languages. 1975. p. 209-213.
16. de Souza R. On the decidability of the equivalence for k-valued transducers // Proc. of 12th International Conference on Developments in Language Theory.(2008. p. 252-263.
17. Thakkar J., Kanade A., Alur R. A transducer-based algorithmic verification of retransmission protocols over noisy channels // Proc. of IFIP Joint International Conference on Formal Techniques for Distributed Systems. Lecture Notes in Computer Science. 2013. Vol. 7892. p. 209-224
18. Veanes M., Hooimeijer P., Livshits B., et al. Symbolic finite state transducers: algorithms and applications // Proc. of the 39th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. ACM SIGPLAN Notices. 2012. Vol. 147. p. 137-150.
19. Watson B.W. A taxonomy of finite automata minimization algorithm // Computing Science Report. Eindhoven University of Technology. 2005. Vol. 93/44. 32 p.
20. Weber A. Decomposing finite-valued transducers and deciding their equivalence // SIAM Journal on Computing. 1993. Vol. 22. p. 175-202.
21. Wolper P., Boigelot B. Verifying systems with infinite but regular state spaces// Proc. 10th Int. Conf. on Computer Aided Verification (CAV-1998). Lecture Notes in Computer Science. 1998. Vol. 1427. p. 88-97.
22. Zakharov V.A. On the decidability of the equivalence problem for orthogonal sequential programs // Grammars. 1999. Vol. 2. p. 271-281.
23. Zakharov V.A.: Equivalence checking problem for finite state transducers over semigroups // Proc. of the 6-th International Conference on Algebraic Informatics (CAI-2015). Lecture Notes in Computer Science. 2015. Vol. 9270. p. 208-221.

24. Zakharov V.A., Podymov V.V. On the application of equivalence checking algorithms for program minimization // Proc. of the Institute for System Programming. 2015. Vol. 27. p. 145-174.