Equivalence Checking Problem for Finite State Transducers over Semigroups

Vladimir A. Zakharov(✉)

Institute for System Programming RAS,
National Research University Higher School of Economics, Moscow, Russia
zakh@cs.msu.su

Abstract. Finite state transducers over semigroups can be regarded as a formal model of sequential reactive programs. In this paper we introduce a uniform technique for checking effectively functionality, k-valuedness, equivalence and inclusion for this model of computation in the case when a semigroup these transducers operate over is embeddable in a decidable group.

1 Introduction

Finite state transducers extend the finite state automata to model functions on strings or lists, that is why they are used in fields as diverse as computational linguistics and model-based testing. In software engineering transducers provide a suitable formal model for various device drivers for manipulating with strings, transforming images, filtering dataflows, inserting fingerprints, etc. (see [1,10]). Algorithms for building compositions of transducers, checking equivalence, reducing their state space considerably enhance the effectiveness of designing, testing, verification and maintenance of such software routines.

Transducers may be used also as 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. It is significant that different sequences of actions may yield the same result. Therefore, the basic actions of a program may be viewed as generating elements of some appropriate semigroup, and the result of computation may be regarded as the 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 states 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.

A. Maletti (Ed.): CAI 2015, LNCS 9270, pp. 208–221, 2015.
DOI: 10.1007/978-3-319-23021-4_19

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 the equivalence checking problem and some related problems for finite state transducers over semigroups. The study of these problems for classical transducers that operate on words began in the early 60s. First, it was shown that the equivalence checking problem is undecidable for non-deterministic transducers [6] even over 1-letter input alphabet [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. It was proved that it is decidable in polynomial time whether the cardinality of the image of every word by a given transducer is bounded [15] and whether it is bounded by a given integer k [7]. The equivalence checking problem was shown also to be decidable for deterministic [3], functional (single-valued) transducers [2,13], and k-valued transducers [5,16]. In a series of papers [4,11,12,14] a construction to decompose k-valued transducers into a sum of functional and unambiguous ones was developed and used for checking bounded valuedness, k-valuedness and equivalence of finite state transducers over words.

This paper offers an alternative technique for the analysis of finite state transducers over semigroups. To check the equivalence of transducers π_1 and π_2 we associate with them a Labeled Transition System Γ_{π_1,π_2}. Each path in this LTS represents all possible runs of π_1 and π_2 on the same input word. Every node u of Γ_{π_1,π_2} keeps track of the states of π_1 and π_2 achieved at reading some input word and the *deficiency* of the output words computed so far. If both transducers reach their final states and the deficiency of their outputs is nonzero then this indicates that π_1 and π_2 produce different images for the same word, and, hence, they are not equivalent. The nodes of Γ_{π_1,π_2} that capture this effect are called *rejecting* nodes. Thus, the equivalence checking of π_1 and π_2 is reduced to checking the reachability of rejecting nodes in LTS Γ_{π_1,π_2}. We show that one needs to analyze only a bounded fragment of Γ_{π_1,π_2} to certify (un)reachability of rejecting nodes. The size of this fragment is polynomial of the size of π_1 and π_2 if both transducers are deterministic, and single-exponential if they are k-bounded. The same approach is applicable for checking k-valuedness of transducers over semigroups.

Initially, this LTS-based approach was introduced and developed in [17] for equivalence checking sequential programs in polynomial time. The concept of deficiency and a similar way of its application to the analysis of classical transducers was independently introduced in [4] and used in [12,14] under the names "Advance or Delay Action" (ADA), or "Lead or Delay Action" (LDA). The main advantage of our approach (apart from the fact that it is applicable to a more general type of transducers) is twofold. First, unlike one used in [11,16], it does not require a pre-processing (decomposition) of transducers to be analyzed and can be applied to any given transducers at once. Second, the checking procedure does not rely on the specific features of internal structures (like the analysis of strongly connected components used in [12,14]) of transducers under consideration and makes a plain depth-first search of rejecting nodes in the corresponding LTS. Complexity issues of our technique are shortly discussed in the conclusion.

2 Preliminaries

Given a finite *alphabet* A, denote by A^* the set of all finite *words* over A. A *finite state automaton* over A is 5-tuple $\mathcal{A} = \langle A, Q, init, F, \varphi \rangle$, where Q is a finite set of *states*, *init* is an *initial state*, F is a subset of *final states*, and $\varphi, \varphi \subseteq Q \times A \times Q$, is a *transition relation*. An automaton $\mathcal{A}$ *accepts* a word $w = a_1 a_2 \dots a_n$ if there exists a sequence of states $q_0, q_1, \dots, q_n$ such that $q_0 = init$, $q_n \in F$, and $(q_{i-1}, a_i, q_i) \in \varphi$ holds for every $i, 1 \leq i \leq n$. A *language* $L(\mathcal{A})$ of $\mathcal{A}$ is the set of all words accepted by $\mathcal{A}$. We write $\mathcal{A}[q]$ for the automaton $\langle A, Q, q, F, \varphi \rangle$ which has q for the initial state.

Let $S = (B, \cdot, e)$ be a semigroup generated by a set of elements B and having the identity element e. A *finite state transducer* over S is 6-tuple $\pi = \langle A, S, Q, q_0, F, T \rangle$, where Q is a finite set of states, q_0 is an initial state, F is a subset of final states, and $T, T \subseteq Q \times A \times S \times Q$, is a finite transition relation. Quadruples (q, a, s, q') in T are called *transitions* and denoted by $q \xrightarrow{a/s} q'$. We will denote by $\mathcal{A}_\pi$ the underlying finite state automaton $\langle A, Q, q_0, F, \varphi_\pi \rangle$, where $\varphi_\pi = \{(q, a, q') : q \xrightarrow{a/s} q' \text{ for some } s \text{ in } S\}$.

A *run* of π on a word $w = a_1 a_2 \dots a_n$ is a sequence of transitions of the form

$$q_i \xrightarrow{a_1/s_1} q_{i+1} \xrightarrow{a_2/s_2} \cdots \xrightarrow{a_{n-1}/s_{n-1}} q_{i+n-1} \xrightarrow{a_n/s_n} q_{i+n} \ . \tag{1}$$

The element $s = s_1 \cdot s_2 \cdots s_n$ of the semigroup S is called an *image* of w, and the pair (w, s) is called the *label* of a run (1). We will use notation $q_i \xrightarrow{w/s} q_{i+n}$ for a run of a transducer. If $q_i = q_0$ then (1) is an *initial* run. If $q_{i+n} \in F$ then (1) is a *final* run. A run which is both initial and final is called *complete*. By $Lab(\pi)$ we denote the *transduction* relation realized by π which is the set of labels (w, s) of all complete runs of π. A state q is *useful* if at least one complete run passes via q. In what follows we will assume that all states of the transducers under consideration are useful; in [4] such transducers are called *trim*. A transducer π is *deterministic* if for every letter a and a state q the set T contains at most one transition of the form $q \xrightarrow{a/s} q'$. A transducer π is k-*valued*, where k is a positive integer, if for every input word w the transduction relation $Lab(\pi)$ contains at most k labels of the form (w, s). A 1-valued transducer π is also called *functional*. Transducers π' and π'' are *equivalent* ($\pi' \sim \pi''$ in symbols) if $Lab(\pi') = Lab(\pi'')$.

In the rest of the paper we define and study procedures for checking equivalence and k-valuedness of finite state transducers over a semigroup S which can be embedded in a group. A semigroup S is *embeddable* in a group G if this group includes a semigroup S' isomorphic to S. The set of necessary and sufficient conditions for the embeddability of a semigroup in a group were given in [9]. The conditions are countably infinite in number and no finite subset will suffice. In fact, a free semigroup is embeddable in a free group, and any commutative semigroup can be embedded in a group iff it is cancellative. Without loss of generality in what follows it will be assumed that the transducers under consideration operate over a finitely generated decidable group G (i.e. there exists an algorithm for checking whether two words in the generators of G represent the same element), and, given an element s, we write s^- for the element of G which is inverse of s.

3 Equivalence Checking Deterministic Transducers

Let $\pi = \langle A, G, Q, q_0, F, T\rangle$ and $\pi' = \langle A, G, Q', q'_0, F', T'\rangle$ be deterministic transducers over a finitely generated decidable group G. To check their equivalence we define the Labeled Transition System (LTS) $\Gamma^0_{\pi,\pi'} = \langle Q \times Q' \times G, \Rightarrow\rangle$. The nodes of $\Gamma^0_{\pi,\pi'}$ are triples of the form (q, q', g), where $q \in Q$, $q' \in Q'$, and $g \in G$. The third component g in this triple is called a *deficiency* (of initial runs arriving at the states q and q').

The transition relation $\Rightarrow$ is defined as follows: for every pair of nodes $v_1 = (q_1, q'_1, g_1)$ and $v_2 = (q_2, q'_2, g_2)$, and for every letter a a relation $v_1 \overset{a}{\Rightarrow} v_2$ holds iff $q_1 \xrightarrow{a/s} q_2$ and $q'_1 \xrightarrow{a/s'} q'_2$ are transitions in π and π' respectively, and $g_2 = s^- g_1 s'$.

Given a word $w = a_1a_2 \dots a_n$ and a pair of nodes $v = (q_1, q'_1, g_1)$ and $u = (q_2, q'_2, g_2)$ we write $v \overset{w}{\Rightarrow} u$ as shorthand notation of a sequence $v \overset{a_1}{\Rightarrow} v_1 \overset{a_2}{\Rightarrow} \cdots \overset{a_{n-1}}{\Rightarrow} v_{n-1} \overset{a_n}{\Rightarrow} u$ which is called a *path* in $\Gamma^0_{\pi,\pi'}$. In this case we say that a node u is *reachable* from a node v. It is easy to see that $v \overset{w}{\Rightarrow} u$ holds iff $q_1 \xrightarrow{w/s} q_2$, $q'_1 \xrightarrow{w/s'} q'_2$, and $s^- g_1 s' = g_2$.

The node $v_{src} = (q_0, q'_0, e)$, where e is the identity element of G, is called the *source node* of $\Gamma^0_{\pi,\pi'}$. Denote by $V^0_{\pi,\pi'}$ the set of nodes reachable in LTS $\Gamma^0_{\pi,\pi'}$ from v_{src}. A node (q, q', g) is called *rejecting* if it satisfies at least one of the following requirements:

1. both q and q' are final states of π and π', and $g \neq e$;
2. exactly one of the states q or q' is final;
3. for some letter a only one of the states q or q' has a a-transition, whereas the other state does not.

The set of all rejecting nodes of LTS $\Gamma^0_{\pi,\pi'}$ is denoted by $R^0_{\pi,\pi'}$.

Lemma 1. *Deterministic transducers π and π' are equivalent iff $V^0_{\pi,\pi'} \cap R^0_{\pi,\pi'} = \emptyset$.*

Proof. Follows immediately from the definitions of LTS $\Gamma^0_{\pi,\pi'}$, $V^0_{\pi,\pi'}$, $R^0_{\pi,\pi'}$ and the equivalence $\sim$ in view of the fact that π and π' are both deterministic and trim. □

Thus, the equivalence checking of deterministic trim transducers is reduced to the searching of rejecting nodes in the set of reachable nodes of LTS $\Gamma^0_{\pi,\pi'}$. Next we show how to cut down the search space.

Lemma 2. *If the set $V^0_{\pi,\pi'}$ contains a pair of nodes $v_1 = (q, q', g_1)$ and $v_2 = (q, q', g_2)$ such that $g_1 \neq g_2$ then $V^0_{\pi,\pi'} \cap R^0_{\pi,\pi'} \neq \emptyset$.*

Proof. Suppose, to the contrary, that $V^0_{\pi,\pi'} \cap R^0_{\pi,\pi'} = \emptyset$, and, hence, $\pi \sim \pi'$. By the definition of $\Gamma^0_{\pi,\pi'}$ there is such a word w_0 that $q_0 \xrightarrow{w_0/s_0} q$, $q'_0 \xrightarrow{w_0/s'_0} q'$, and $g_1 = s_0^- s'_0$. Since the state q is useful, there is such a word w that $q_0 \xrightarrow{w_0/s_0} q \xrightarrow{w/s} p$ is a complete run of π. As far as $\pi \sim \pi'$ and π' is deterministic, the run $q'_0 \xrightarrow{w_0/s'_0} q' \xrightarrow{w/s'} p'$ of π' is complete and $s_0 s = s'_0 s'$. Hence, $g_1 = s(s')^-$. Inasmuch as $q \xrightarrow{w/s} p$ and $q' \xrightarrow{w/s'} p'$, there is a path $v_2 \stackrel{w}{\Rightarrow} (p, p', g)$ in $\Gamma^0_{\pi,\pi'}$, where $g = s^- g_2 s'$. Having in mind that (p, p', g) is in $V^0_{\pi,\pi'}$, both states p and p' are final, and assuming that $V^0_{\pi,\pi'} \cap R^0_{\pi,\pi'} = \emptyset$, we arrive at the equality $g = e$. Therefore, $g_2 = s(s')^- = g_1$ which contradicts the premise of Lemma. □

By Lemmata 1 and 2, to check the equivalence of deterministic trim transducers π and π' it is sufficient to analyze at most $|Q||Q'| + 1$ nodes reachable from the source node of LTS $\Gamma^0_{\pi,\pi'}$. This consideration brings us to

Theorem 1. *The equivalence problem for deterministic transducers over finitely generated decidable group G is decidable. Moreover, if the word problem for G is decidable in polynomial time then the equivalence problem for deterministic transducers over G is decidable in polynomial time as well.*

4 Checking Functional Transducers

To check the functionality of a transducer $\pi = \langle A, G, Q, q_0, F, T\rangle$ we also take advantage of LTSs. Let $\pi = \langle A, G, Q, q_0, F, T\rangle$ and $\pi' = \langle A, G, Q', q'_0, F', T'\rangle$ be a pair of transducers. Define a LTS $\Gamma^1_{\pi,\pi'} = \langle Q \times Q' \times G, \Rightarrow\rangle$ as follows: for every pair of nodes $v_1 = (q_1, q'_1, g_1)$ and $v_2 = (q_2, q'_2, g_2)$, and for every letter a, a relation $v_1 \stackrel{a}{\Rightarrow} v_2$ holds iff there exist transitions $q_1 \xrightarrow{a/s} q_2$ and $q'_1 \xrightarrow{a/s'} q'_2$ such that $g_2 = s^- g_1 s'$, and $L(\mathcal{A}_\pi[q_2]) \cap L(\mathcal{A}_{\pi'}[q'_2]) \neq \emptyset$. The set of all nodes of LTS $\Gamma^1_{\pi,\pi'}$ reachable from the source node (q_0, q_0, e) is denoted by $V^1_{\pi,\pi'}$. We say that (q_1, q_2, g) is a *rejecting* node if q_1 and q_2 are final states, and $g \neq e$. The set of all rejecting nodes of LTS $\Gamma^1_{\pi,\pi'}$ is denoted by $R^1_{\pi,\pi'}$. The lemmata below can be proved using the same reasoning as in previous section.

Lemma 3. *A transducer π is functional iff $V^1_{\pi,\pi} \cap R^1_{\pi,\pi} = \emptyset$.*

Lemma 4. *If the set $V^1_{\pi,\pi}$ includes a pair of nodes $v_1 = (q, p, g_1)$ and $v_2 = (q, p, g_2)$ such that $g_1 \neq g_2$ then $V^1_{\pi,\pi} \cap R^1_{\pi,\pi} \neq \emptyset$.*

As it follows from Lemmata 3 and 4, to check functionality of a transducer π one needs only to analyze at most $|Q|^2 + 1$ nodes reachable from the source node of $\Gamma^1_{\pi,\pi}$.

Theorem 2. *The functionality of transducers over finitely generated decidable group G can be checked effectively. Moreover, if the word problem for G is decidable in polynomial time then the functionality checking can be performed in polynomial time as well.*

The equivalence of functional transducers π and π' can be checked in the same way by means of LTS $\Gamma^1_{\pi,\pi'}$. But now we need to check in advance that $L(\mathcal{A}_\pi[q_0]) = L(\mathcal{A}_{\pi'}[q'_0])$ since, unlike the case of deterministic transducers, the nodes (q_1, q_2, g) in $\Gamma^1_{\pi,\pi'}$ such that exactly one of the states q_1 and q_2 is final can not be regarded as rejecting ones.

Lemma 5. *If $L(\mathcal{A}_\pi[q_0]) = L(\mathcal{A}_{\pi'}[q'_0])$ then functional transducers π and π' are equivalent iff $V^1_{\pi,\pi'} \cap R^1_{\pi,\pi'} = \emptyset$.*

Lemma 6. *If the set $V^1_{\pi,\pi'}$ includes a pair of nodes $v_1 = (q, q', g_1)$ and $v_2 = (q, q', g_2)$ such that $g_1 \neq g_2$ then $V^1_{\pi,\pi'} \cap R^1_{\pi,\pi'} \neq \emptyset$.*

Theorem 3. *The equivalence problem for functional transducers over finitely generated decidable group G is decidable. Moreover, if the word problem for G is decidable in polynomial time then the equivalence problem for functional transducers is PSPACE-complete.*

5 Checking 2-Valuedness of Transducers

The LTS-based techniques put forward in Sections 3 and 4 for checking the equivalence of deterministic and functional transducers can be developed further to cope with the analysis of k-valued transducers. For the sake of clarity we consider in details only the case of $k = 2$; the same arguments supplied with a bit more cumbersome combinatorics gives a general solution to the checking problems for k-valued finite state transducers.

We begin with checking 2-valuedness of transducers over a decidable group. Given a transducer $\pi = \langle A, G, Q, q_0, F, T\rangle$ define a LTS $\Gamma^2_\pi = \langle Q \times (Q \times G)^2, \Rightarrow \rangle$ as follows: for every pair of nodes $v_1 = (q_1, (q_2, g_{12}), (q_3, g_{13}))$ and $v_2 = (q'_1, (q'_2, g'_{12}), (q'_3, g'_{13}))$, and a letter a, a transition $v_1 \overset{a}{\Rightarrow} v_2$ takes place if there exist transitions $q_1 \xrightarrow{a/s_1} q'_1$, $q_2 \xrightarrow{a/s_2} q'_2$, and $q_3 \xrightarrow{a/s_3} q'_3$ such that the equalities $g'_{12} = s_1^- g_{12} s_2$ and $g'_{13} = s_1^- g_{13} s_3$ hold, and $L(\mathcal{A}_\pi[q'_1]) \cap L(\mathcal{A}_\pi[q'_2]) \cap L(\mathcal{A}_\pi[q'_3]) \neq \emptyset$.

A triple of states (q_1, q_2, q_3) will be called a *type* of a node $(q_1, (q_2, g_{12}), (q_3, g_{13}))$. As in the case of 1-valuedness, we define the set V^2_π of all nodes reachable in LTS Γ^2_π from the source node $(q_0, (q_0, e), (q_0, e))$. From the definitions of Γ^2_π and V^2_π it follows that a node $v = (q_1, (q_2, g_{12}), (q_3, g_{13}))$ is in V^2_π iff there exists such a word w that $q_0 \xrightarrow{w/s_1} q_1$, $q_0 \xrightarrow{w/s_2} q_2$, $q_0 \xrightarrow{w/s_3} q_3$, and $g_{12} = s_1^- s_2$, $g_{13} = s_1^- s_3$.

The set R^2_π of rejecting nodes includes all such nodes $(q_1, (q_2, g), (q_3, h))$ that q_1, q_2, q_3 are final states, and $g \neq e$, $h \neq e$, $g \neq h$ hold.

Lemma 7. *A transducer π is 2-valued iff $V^2_\pi \cap R^2_\pi = \emptyset$.*

Proof. Follows from the definitions of V^2_π, R^2_π, and 2-valuedness property. □

Now we need to cut off the space of V^2_π for searching the rejecting nodes. This is achieved by means of the following two lemmata. Their proofs are based on the pigeonhole principle and basic group-theoretic properties.

Lemma 8. *Suppose that V^2_π includes four nodes $v_i = (q, (q', g'_i), (q'', g''_i))$, $1 \le i \le 4$, of the same type such that the inequalities $g'_i \ne g'_j$, $g''_i \ne g''_j$, and $g'_i(g''_i)^- \ne g'_j(g''_j)^-$ hold for every pair of indices i, j, $1 \le i < j \le 4$. Then $V^2_\pi \cap R^2_\pi \ne \emptyset$.*

Proof. Since all nodes v_i, $1 \le i \le 4$, are in V^2_π then $L(\mathcal{A}_\pi[q]) \cap L(\mathcal{A}_\pi[q']) \cap L(\mathcal{A}_\pi[q'']) \ne \emptyset$. Hence, there exists such a word w that $q \xrightarrow{w/s} p$, $q' \xrightarrow{w/s'} p'$, and $q'' \xrightarrow{w/s''} p''$ are final runs of the transducer π. Then, by definition of Γ^2_π, the set of reachable nodes V^2_π includes four nodes $u_i = (p, (p', s^- g'_i s'), (p'', s^- g''_i s''))$, $1 \le i \le 4$. If u_1 is not a rejecting node then at least one of the equalities hold: $s^- g'_1 s' = e$, $s^- g''_1 s'' = e$, or $s^- g'_1 s' = s^- g''_1 s''$. Without loss of generality consider the case of $s^- g'_1 s' = e$ (two other possibilities are treated in the similar way). Since $g'_1 \ne g'_2$, this case implies $s^- g'_2 s' \ne e$. Therefore, if u_2 is not a rejecting node then this is due to one of the equalities $s^- g''_2 s'' = e$, or $s^- g'_2 s' = s^- g''_2 s''$. Consider the case of $s^- g''_2 s'' = e$ (the other possibility is treated similarly). Since $g'_1 \ne g'_3$ and $g''_2 \ne g''_3$, the above equalities $s^- g'_1 s' = e$ and $s^- g''_2 s'' = e$ imply $s^- g'_3 s' \ne e$ and $s^- g''_3 s'' \ne e$. Therefore, if u_3 is not a rejecting node then $s^- g'_3 s' = s^- g''_3 s''$. But, taking into account that $g'_1 \ne g'_4$, $g''_2 \ne g''_4$, and $g'_3(g''_3)^- \ne g'_4(g''_4)^-$, the equalities $s^- g'_1 s' = e$, $s^- g''_2 s'' = e$, and $s^- g'_3 s' = s^- g''_3 s''$ bring us to the conclusion that $s^- g'_4 s' \ne e$, $s^- g''_4 s'' \ne e$, and $s^- g'_4 s' \ne s^- g''_4 s''$, which means that $v_4 \in R^2_\pi$. □

Lemma 9. *Let $v_i = (q, (q', g'_i), (q'', g''_i))$, $1 \le i \le 4$, be four pairwise different nodes in LTS Γ^2_π that satisfy one of the following requirements:*

a) $g'_i = g'_j$ holds for every pair i, j, $1 \le i < j \le 4$;
b) $g''_i = g''_j$ holds for every pair i, j, $1 \le i < j \le 4$;
c) $(g'_i)^- g''_i = (g'_j)^- g''_j$ holds for every pair i, j, $1 \le i < j \le 4$.

If a rejecting node is reachable from v_4 then some rejecting node is reachable from one of the nodes v_1, v_2, v_3.

Proof. We consider only the case when all nodes satisfy the first requirement $g'_i = g'$ for every $i, 1 \le i \le 4$. The similar reasoning is adequate for the other alternatives.

Suppose that a rejecting node $u_4 = (p, (p', h'), (p'', h''_4))$ is reachable from v_4 through some word w. Then there are three final runs $q \xrightarrow{w/s} p$, $q' \xrightarrow{w/s'} p'$, and $q'' \xrightarrow{w/s''} p''$ of π such that $h' = s^- g' s'$ and $h''_4 = s^- g''_4 s''$. Since u_4 is a rejecting node, we have $h' \ne e$.

The definition of Γ^2_π guarantees that for every i, $1 \le i \le 3$, there is a path from the node v_i to the node $u_i = (p, (p', h'), (p'', h''_i))$, where $h''_i = s^- g''_i s''$. If $u_1 \notin R^2_\pi$ then either $h''_1 = e$ or $(h')^- h'' = e$. Consider only the case $h''_1 = e$ (the other possibility is treated in the same way). Since $g''_2 \ne g''_1$ and $g''_1 \ne g''_3$, we have $h''_2 \ne e$ and $h''_3 \ne e$. Therefore, if $u_2 \notin R^2_\pi$ then $(h')^- h''_2 = e$. But, as far as $g''_2 \ne g''_3$, it is true that $(h')^- h''_3 \ne e$. Thus, we conclude that u_3 is a rejecting node. □

With Lemmata 8 and 9 in hand we are able to prove

Theorem 4. *If G is a finitely generated decidable group then 2-valuedness is a decidable property of transducers over G.*

Proof. By Lemma 7 we can check 2-valuedness of a transducer π through the reachability analysis of rejecting nodes in LTS Γ^2_π. To this end we introduce a depth-first search of rejecting nodes. It begins with the source node $(q_0, (q_0, e), (q_0, e))$ and keeps track of useful nodes only. Suppose that at some step the traversal reaches a node $v = (q, (q', g'), (q'', g''))$ in Γ^2_π which has not been visited yet. Then the following 4 cases are possible.

1) If v is a rejecting node then the search stops and announces that π is not 2-valued.

2) Otherwise, check if there exist 3 previously visited useful nodes $v_i = (q, (q', g'_i), (q'', g''_i))$, $1 \leq i \leq 3$, of the same type as v that satisfy one of the following requirements:

a) $g' = g'_i$ for every $i,\ 1 \leq i \leq 3$;

b) $g'' = g''_i$ for every $i,\ 1 \leq i \leq 3$;

c) $(g')^- g'' = (g'_i)^- g''_i$ for every $i,\ 1 \leq i \leq 3$.

If so then v is regarded as useless and a backtracking step is made from this node.

3) Otherwise, if 27 useful nodes $v_i = (q, (q', g'_i), (q'', g''_i))$, $1 \leq i \leq 27$, of the same type as v has been already visited then the search stops and announces that π is not 2-valued.

4) Otherwise, the node v is regarded as *useful*, and the search procedure continues its depth-first traversal of LTS Γ^2_π.

If the search backtracks finally to the source node then π is recognized 2-valued.

As it can be seen from the definition of the search procedure, it always terminates at visiting at most $27|Q|^3$ useful nodes of Γ^2_π. Lemma 9 guarantees that by skipping useless nodes we do not miss possible paths to some rejecting nodes. This certifies the completeness of our search. To prove its correctness we need to show that case 3) of the search is correct. Indeed, simple combinatorial considerations disclose that if we have 28 nodes (v and $v_i,\ 1 \leq i \leq 27$) such that neither 4 nodes of them fall under the premise of Lemma 9 (i.e., the nodes are useful) then this set of nodes includes a quadruple of nodes that satisfy the assumptions of Lemma 8. □

Corollary 1. *If the word problem for a group G is decidable in polynomial time then 2-valuedness property of transducers over G can be checked in polynomial time.*

Both Lemmata 8 and 9, as well as the decision procedure defined in Theorem 4 can be readily extended to the case of an arbitrary k: the nodes of LTS Γ^2_π are $(k+1)$-tuples $(q_0, (q_1, h_1), \ldots, (q_k, h_k))$, and to certify the reachability of a rejecting node in Γ^2_π it suffices to visit at most $\binom{k+1}{2}^{\binom{k+1}{2}} |Q|^{k+1} + 1$ useful nodes.

6 Checking the Equivalence of 2-Valued Transducers

Instead of solving the equivalence checking problem for finite state transducers we study a more general inclusion checking problem: given a pair of transducers π and π' check whether $Lab(\pi') \subseteq Lab(\pi)$. The LTS-based approach is invoked once again.

Let $\pi = \langle A, G, Q, q_0, F, T\rangle$ and $\pi' = \langle A, G, Q', q_0', F', T'\rangle$ be a pair of trim 2-valued transducers. Clearly, if $Lab(\pi') \subseteq Lab(\pi)$ then $L(\mathcal{A}_{\pi'}) \subseteq L(\mathcal{A}_\pi)$. Therefore, in this section we deal only with the case of π and π' such that $L(\mathcal{A}_{\pi'}) \subseteq L(\mathcal{A}_\pi)$.

To define an LTS $\Gamma^3_{\pi,\pi'}$ corresponding to the inclusion checking problem for transducers π and π' we introduce a concept of block of states. Let $\widehat{Q}$ be some multiset of states of transducer π. Then a *block of states in* $\widehat{Q}$ is any maximal (i.e., inextensible) subset B of $\widehat{Q}$ such that $\bigcap_{q\in B} L(\mathcal{A}_\pi[q]) \neq \emptyset$, i.e. some word is accepted by every automaton $\mathcal{A}_\pi[q]$, $q \in B$, but no such words are accepted by an automaton $\mathcal{A}_\pi[q']$ for any $q', q' \in \widehat{Q} \setminus B$.

LTS $\Gamma^3_{\pi,\pi'} = \langle V, \Rightarrow\rangle$ is defined as follows. The set of nodes V consists of all such pairs $u = (q', X)$, where $q' \in Q'$, and $X = \{(q_1, g_1), \dots, (q_m, g_m)\} \subseteq Q \times G$, that satisfy the requirement $L(\mathcal{A}_{\pi'}[q']) \cap \bigcap_{i=1}^{m} L(\mathcal{A}_\pi[q_i]) \neq \emptyset$. The pair $(q', \{q_1, \dots, q_m\})$ will be referred to as a *type* of the node u. For every letter a and a pair of nodes $u = (q', X)$ and $v = (p', Y)$ of types (q', B_u) and (p', B_v) respectively a transition $u \overset{a}{\Rightarrow} v$ takes place iff

1. there is transition $q' \xrightarrow{a/s'} p'$ in the transducer π',
2. B_v is a block of states in the multiset $\widehat{Q} = \{\widehat{q} : \exists q\ (q \in B_u \text{ and } q \xrightarrow{a/s} \widehat{q}\ )\ \}$, and
3. a pair (p, h) is in Y if and only if $p \in B_v$ and there exists such a pair (q, g) in X that $q \xrightarrow{a/s} p$ is a transition of transducer π and $h = (s')^- gs$.

As usual, given a word w we write $u \overset{w}{\Rightarrow} v$ for the composition of corresponding 1-letter transitions of LTS. The node $v_{src} = (q_0', \{(q_0, e)\})$ is the *source* node of LTS $\Gamma^3_{\pi,\pi'}$. By $V^3_{\pi,\pi'}$ we denote the set of all nodes reachable from v_{src}. A node (q', X) such that $q' \in F'$, and for every pair (q, g) in X either $q \notin F$, or $g \neq e$, is called a *rejecting* node. The set of rejecting nodes of $\Gamma^3_{\pi,\pi'}$ is denoted by $R^3_{\pi,\pi'}$. The intended meaning of LTS $\Gamma^3_{\pi,\pi'}$ with regard to the inclusion checking of π and π' is clarified in the propositions below.

Proposition 1. *Let w_0 and w_1 be arbitrary words, and $q_0' \xrightarrow{w_0/s_0'} q_1' \xrightarrow{w_1/s_1'} q_2'$ be a complete run of transducer π'. Then there exists such a node $v = (q_1', X)$ that $v_{src} \overset{w_0}{\Rightarrow} v$ and for every complete run $q_0 \xrightarrow{w_0/s_0} q_1 \xrightarrow{w_1/s_1} q_2$ of transducer π the multiset X includes a pair $(q_1, (s_0')^- s_0)$.*

Proposition 2. *Suppose that* $v_{src} \stackrel{w_0}{\Rightarrow} (q', X)$. *Then there exist such a word* w_1 *and a complete run* $q'_0 \xrightarrow{w_0/s'_0} q'_1 \xrightarrow{w_1/s'_1} q'_2$ *of transducer* π' *that for every complete run* $q_0 \xrightarrow{w_0/s_0} q_1 \xrightarrow{w_1/s_1} q_2$ *of transducer* π *the multiset* X *includes a pair* $(q_1, (s'_0)^- s_0)$.

Both propositions can be proved by induction on the length of w_0 relying on the definition of transition relation $\Rightarrow$ only. The correctness of these propositions is due to the fact that the type of every reachable node is specified as block of states.

Lemma 10. $Lab(\pi') \subseteq Lab(\pi) \iff V^3_{\pi,\pi'} \cap R^3_{\pi,\pi'} = \emptyset$.

Proof. Follows from Propositions 1,2 above and the definition of rejecting node. □

We show that, even though the set $V^3_{\pi,\pi'}$ may be infinite, only finitely many nodes must be checked to verify (un)reachability of rejecting nodes.

Consider an arbitrary reachable node v of type (q', B). Since the transducer π is 2-valued, for every state q of π at most two copies of q may occur in the multiset B. Therefore, $|B| \leq 2|Q|$, and the total number of types of reachable nodes in $\Gamma^3_{\pi,\pi'}$ does not exceed $|Q'|3^{|Q|}$.

Consider the language $L = L(\mathcal{A}_{\pi'}[q']) \cap \bigcap_{q \in B} L(\mathcal{A}_\pi[q])$; it will be called a *language of type* (q', B). By definition of $\Gamma^3_{\pi,\pi'}$, this language is non-empty. The set of types of all reachable nodes can be divided into three classes depending on the properties of L. A type (q', B) will be called *A-type* iff there exists such a word w in L which has two different images s'_1 and s'_2 of w in two final runs $q' \xrightarrow{w/s'_1} p'_1$ and $q' \xrightarrow{w/s'_2} p'_2$ of transducer π'. A type (q', B) will be called *B-type* iff it does not belong to the class A and there exist a state q in the multiset B and a word w in L which has two different images s_1 and s_2 in two final runs $q \xrightarrow{w/s_1} p_1$ and $q \xrightarrow{w/s_2} p_2$ of transducer π. All other types will be called *C-types*. Lemmata below elucidate some properties of these classes that are crucial for the solution of the inclusion checking problem.

Lemma 11. *Suppose that* $Lab(\pi') \subseteq Lab(\pi)$, *and* (q', B) *be a A-type. Then at most* $2^{|B|}$ *nodes of this type are reachable from the source node.*

Proof. Let L be the language of type (q', B). Consider an arbitrary node $v = (q', X)$ of type (q', B) such that $v_{src} \stackrel{w_0}{\Rightarrow} v$, and an arbitrary pair (q, g) from X. Since (q', B) is A-type, there exists such a word w in L which has different images s'_1 and s'_2 in two final runs $q' \xrightarrow{w/s'_1} p'_1$ and $q' \xrightarrow{w/s'_2} p'_2$ of transducer π'. By definition of L, the transducer π has a final run $q \xrightarrow{w/s} q_1$. Notice, that the elements s'_1, s'_2 and s depend on the type (q', B) and the state q only. By Proposition 2, transducers π and π' have initial runs $q_0 \xrightarrow{w_0/s_0} q$ and $q'_0 \xrightarrow{w_0/s'_0} q'$ such that $g = (s'_0)^- s_0$. Then the transducer π' has two complete runs $q'_0 \xrightarrow{w_0/s'_0} q' \xrightarrow{w/s'_1} p'_1$ and $q'_0 \xrightarrow{w_0/s'_0} q' \xrightarrow{w/s'_2} p'_2$, and the transducer π has a complete run $q_0 \xrightarrow{w_0/s_0} q \xrightarrow{w/s} q_1$.

Since π is a 2-valued transducer, $s'_0 s'_1 \neq s'_0 s'_2$, and $Lab(\pi') \subseteq Lab(\pi)$, we may be sure that at least one of the equalities $s_0 s = s'_0 s'_1$ or $s_0 s = s'_0 s'_2$ holds. Hence, either $g = s'_1 s^-$, or $g = s'_2 s^-$. The assertion of the Lemma follows from the fact that both possible values of g depend on the type (q', B) and the state q only. □

Lemma 12. *Suppose that $Lab(\pi') \subseteq Lab(\pi)$, and (q', B) is a B-type. Then at most $3^{|B|}$ nodes of this type are reachable from the source node.*

Proof. Let L be the language of type (q', B). Consider an arbitrary node $v = (q', X)$ of type (q', B) such that $v_{src} \overset{w_0}{\Rightarrow} v$. Let a pair (q, g) in X be such that for some word w in L final runs $q \xrightarrow{w/s_1} p_1$ and $q \xrightarrow{w/s_2} p_2$ of transducer π yield different images of w. Consider an arbitrary pair (p, h) in X. Since $w \in L$, there exist final runs $p \xrightarrow{w/s} p_3$ and $q' \xrightarrow{w/s'} p'$ of π and π'. By referring to Proposition 2 we conclude the following. Since $Lab(\pi') \subseteq Lab(\pi)$, exactly one of the equalities $s' = gs_1$ or $s' = gs_2$ holds. Since π is a 2-valued transducer, exactly one of the equalities $gs_1 = hs$ or $gs_2 = hs$ is valid. Hence, either $h = s's^-$, or $h = s'(s_1)^- s_2 s^-$, or $h = s'(s_1)^- s_2 s^-$. The assertion of Lemma follows from the fact that these possible values of h depend on the type (q', B) and the states q and p only. □

Let (q',B) be a C-type, where $B = \{q_1, \ldots, q_m\}$, and L be the language of this type. Associate with (q',B) any word w_0 from L and consider a final run $q' \xrightarrow{w_0/s'} p'$ of transducer π' and final runs $q_i \xrightarrow{w_0/s_i} p_i$ for every i, $1 \leq i \leq m$. The tuple $(s', s_1, \ldots, s_m)$ of elements in S will be called a w_0-*characteristics* of the type (q', B). This characteristics will help us to narrow the search space. Suppose that $u = (q', \{(q_1, g_1), \ldots, (q_m, g_m)\})$ is a reachable node of the C-type (q', B). If $s' \neq g_i s_i$ holds for every i, $1 \leq i \leq m$, then, by definition of LTS $\Gamma^3_{\pi,\pi'}$, a rejecting node is reachable from u. We will say that such a node u is *pre-rejecting* node of the type (q', B). Otherwise, the set X can be split into two subsets $X_0 = \{(q_i, g_i) : s' = g_i s_i, 1 \leq i \leq m\}$ and $X_1 = \{(q_j, g_j) : s' \neq g_j s_j, 1 \leq j \leq m\}$ such that $X_0 \neq \emptyset$. We will use a notation $(q', X_0 \oplus X_1)$ for such a node u. Note that since π is a 2-valued transducer, $g_i s_i = g_j s_j$ holds for every two pairs (q_i, g_i), (q_j, g_j) from X_1.

Lemma 13. *Let (q', B) be a C-type, $B = \{q_1, \ldots, q_m\}$, and $k = 2m$. Suppose that $k+1$ nodes $u_1 = (q', X_0 \oplus X_1), \ldots, u_{k+1} = (q', X_0 \oplus X_{k+1})$ of type (q', B) are reachable from the source node. Then a rejecting node is reachable from one of the nodes $u_1, \ldots, u_{k+1}$ iff a rejecting node is reachable from one of the nodes $u_1, \ldots, u_k$.*

Proof. Let $(s', s_1, \ldots, s_m)$ be a characteristics of the type (q', B). Assume that $X_0 = \{(q_1, g_1), \ldots, (q_\ell, g_\ell)\}$ and $X_j = \{(q_{\ell+1}, g_{\ell+1j}), (q_m, g_{mj})\}$ for every j, $1 \leq j \leq k+1$.

Suppose that $u_{k+1} \overset{w}{\Rightarrow} v$ holds for some rejecting node v and a word w. Then, by definition of $\Gamma^3_{\pi,\pi'}$, the transducer π' has a final run $q' \xrightarrow{w/s'} p'$ and for every i, $1 \leq i \leq m$, the transducer π either has no final runs on the word w from the state q_i, or every final run $q_i \xrightarrow{w/t_i} p_i$ yields an image t_i of w such that $s' \neq g_{ik+1}t_i$ (actually, at most two such images t_{i1} and t_{i2} are possible due to the fact that π is a 2-valued transducer). We analyze the worst case when the second alternative is achieved for every state q_i, $1 \leq i \leq m$. Thus, we have at most $2(m-1)$ elements $t_{i\sigma}$, $\sigma \in \{1,2\}$ from G that are images of w on final runs from the states $q_{\ell+1}, \dots, q_m$.

If a rejecting node is not reachable from, say, a node u_1 then for some (q_i, g_{i1}) from X_1 and for some image t of the word w the equality $s' = g_{i1}t$ holds, i.e. $g_{i1} = s't^-$. Recall that for any other pair (q_j, g_{j1}) we have $g_{i1}s_i = g_{j1}s_j$, i.e. $g_{j1} = s't^-s_is_j^-$. This means that the image t completely defines all elements g_{j1}, $\ell+1 \leq j \leq m$, in X_1. Clearly, different images of the word w define the elements in the different sets X_i. Since the amount of images of w does not exceed $2(m-1) < k$, there exists such node u_i, $1 \leq i \leq k$, that $s' \neq g_{ji}t_{j\sigma}$ holds for every component (q_j, g_{ji}) of X_i and image(s) $t_{j\sigma}$ of the word w. The latter means that a rejecting node is reachable from u_i. □

Theorem 5. *If G is a finitely generated decidable group G then inclusion problem $Lab(\pi') \subseteq Lab(\pi)$ for 2-valued transducers over G is decidable.*

Proof. The search of rejecting nodes in $\Gamma^3_{\pi,\pi'}$ begins with the source node v_{src}. Suppose that at some step the traversal reaches a node $u = (q', X)$ of a type (q', B), and u has not been visited yet. Then the following 6 cases are possible.
1) If $u \in R^3_{\pi,\pi'}$ then the search stops and announces that π does not include π'.
2) Otherwise, if (q', B) is a A-type and $2^{|B|}$ nodes of the same type have been already visited then the search stops and announces that π does not include π'.
3) Otherwise, if (q', B) is a B-type and $3^{|B|}$ nodes of the same type have been already visited then the search stops and announces that π does not include π'.
4) Otherwise, if (q', B) is a C-type, and u is a pre-rejecting node of this type then the search stops and announces that π does not include π'.
5) Otherwise, if (q', B) is a C-type, $u = (q', X_0 \oplus X_1)$, and $2|B|$ nodes of the form $u_i = (q', X_0 \oplus X_{1i})$ have been already visited then the search backtracks from u.
6) Otherwise, the search procedure continues its depth-first traversal of LTS Γ^2_{π}. If the backtracking ends in the source node then the inclusion $Lab(\pi') \subseteq Lab(\pi)$ holds.

Termination, correctness and completeness of this search procedure follow from Lemmata 10-13. As it can be seen from the description of the search procedure, to check the inclusion $Lab(\pi') \subseteq Lab(\pi)$ less than $|Q'|8^{|Q|}$ nodes of LTS $\Gamma^3_{\pi,\pi'}$ have to be analyzed. □

Corollary 2. *The equivalence checking problem for 2-valued transducers over finitely generated decidable group G is decidable. Moreover, if the word problem for G is decidable in polynomial time then the equivalence checking problem for 2-valued transducers over G is decidable in single exponential time.*

The same approach is applicable to equivalence checking of k-valued transducers for an arbitrary k. But till now the author did not find adequate means for presenting the general solution of this problem in short terms; this remains the topic for further research.

7 Conclusion

The complexity of checking procedures defined in Sections 3-6 depends on the complexity of the word problem for a group G. The time complexity of our algorithms for the cases when G is the free group is estimated below on the following parameters: n (number of states), m (number of transitions), and ℓ (maximal length of the outputs of transitions).

- deterministic equivalence checking: $O(\ell n^3)$,
- functionality checking: $O(\ell m^2 n^2)$,
- k-valuedness checking: $O((k+1)^{2(k+1)^2} \ell m^{k+1} n^{k+1})$,
- functional equivalence checking: $2^{O(n)}$;
- 2-valued equivalence checking: $2^{O(n \log m)}$.

One can compare these complexity estimates with previously known upper bounds for the complexity of k-valuedness checking $O(2^{(k+1)^4} \ell m^{k+1} n^{k+1})$ obtained in [12] and equivalence checking of k-valued transducers $2^{O(\ell k^5 n^{k+4})}$ presented in [14]. As is easy to see, even the best known algorithms for the analysis of k-valued transducers have the complexity which is exponential of k. So, an open question is if it is possible to check k-valuedness and equivalence of nondeterministic transducers in time polynomial of k.

The author would like to thank the anonymous referees whose keen and valuable comments helped him to improve the original version of the paper.

References

1. Alur, R., Cerny, P.: Streaming transducers for algorithmic verification of single-pass list-processing programs. In: Proc. of 38th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, pp. 599–610 (2011)
2. Blattner, M., Head, T.: Single-valued a-transducers. Journal of Computer and System Sciences **15**, 310–327 (1977)
3. Blattner, M., Head, T.: The decidability of equivalence for deterministic finite transducers. Journal of Computer and System Sciences **19**, 45–49 (1979)
4. Beal, M.-P., Carton, O., Prieur, C., Sakarovitch, J.: Squaring transducers: an efficient procedure for deciding functionality and sequentiality. Theoretical Computer Science **292** (2003)
5. Culik, K., Karhumaki, J.: The equivalence of finite-valued transducers (on HDTOL languages) is decidable. Theoretical Computer Science **47**, 71–84 (1986)
6. Griffiths, T.: The unsolvability of the equivalence problem for ε-free nondeterministic generalized machines. Journal of the ACM **15**, 409–413 (1968)
7. Gurari, E., Ibarra, O.: A note on finite-valued and finitely ambiguous transducers. Mathematical Systems Theory **16**, 61–66 (1983)

8. Ibarra, O.: The unsolvability of the equivalence problem for Efree NGSM's with unary input (output) alphabet and applications. SIAM Journal on Computing **4** (1978)
9. Malcev, A.I.: Uber die Einbettung von assoziativen Systemen. Gruppen, Rec. Math. (Mat. Sbornik) N.S. **6**, 331–336 (1939)
10. Veanes, M., Hooimeijer, P., Livshits, B., et al.: Symbolic finite state transducers: algorithms and applications. In: Proc. of the 39th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (2012)
11. Sakarovitch, J., de Souza, R.: On the decomposition of k-valued rational relations. In: Proc. of 25th International Symposium on Theoretical Aspects of Computer Science, pp. 621–632 (2008)
12. Sakarovitch, J., de Souza, R.: On the decidability of bounded valuedness for transducers. In: Ochmański, E., Tyszkiewicz, J. (eds.) MFCS 2008. LNCS, vol. 5162, pp. 588–600. Springer, Heidelberg (2008)
13. Schutzenberger, M.P.: Sur les relations rationnelles. In: Brakhage, H. (ed.) GI-Fachtagung 1975. LNCS, vol. 33, pp. 209–213. Springer, Heidelberg (1975)
14. de Souza, R.: On the decidability of the equivalence for k-valued transducers. In: Ito, M., Toyama, M. (eds.) DLT 2008. LNCS, vol. 5257, pp. 252–263. Springer, Heidelberg (2008)
15. Weber, A.: On the valuedness of finite transducers. Acta Informatica **27**, 749–780 (1989)
16. Weber, A.: Decomposing finite-valued transducers and deciding their equivalence. SIAM Journal on Computing **22**, 175–202 (1993)
17. Zakharov, V.A.: An efficient and unified approach to the decidability of equivalence of propositional program schemes. In: Proc. of the 25th International Colloquium "Automata, Languages and Programming", pp. 247–258 (1998)