Fundamenta Informaticae XXI (2001) 1001–1016
DOI 10.3233/FI-2016-0000
IOS Press

Copyful Streaming String Transducers

Emmanuel Filiot*
Université libre de Bruxelles
efiliot@ulb.ac.be

Pierre-Alain Reynier†
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Abstract. Copyless streaming string transducers (copyless SST) have been introduced by R. Alur and P. Černý in 2010 as a one-way deterministic automata model to define transductions of finite strings. Copyless SST extend deterministic finite state automata with a set of variables in which to store intermediate output strings, and those variables can be combined and updated all along the run, in a linear manner, i.e., no variable content can be copied on transitions. It is known that copyless SST capture exactly the class of MSO-definable string-to-string transductions, and are as expressive as deterministic two-way transducers. They enjoy good algorithmic properties. Most notably, they have decidable equivalence problem (in PSpace).

On the other hand, HDT0L systems have been introduced for a while, the most prominent result being the decidability of the equivalence problem. In this paper, we propose a semantics of HDT0L systems in terms of transductions, and use it to study the class of deterministic copyful SST. Our contributions are as follows:

- (i) HDT0L systems and total deterministic copyful SST have the same expressive power,
- (ii) the equivalence problem for deterministic copyful SST and the equivalence problem for HDT0L systems are inter-reducible, in quadratic time. As a consequence, equivalence of deterministic SST is decidable,
- (iii) the functionality of non-deterministic copyful SST is decidable,
- (iv) determining whether a non-deterministic copyful SST can be transformed into an equivalent non-deterministic copyless SST is decidable in polynomial time.

*Thanks for something to somebody
†Thanks for something else to somebody else

Keywords: words, transducers, equivalence problem

1. Introduction

The theory of languages is extremely rich and important automata-logic correspondences have been shown for various classes of logics, automata, and structures. There are less known automata-logic connections in the theory of transductions. Nevertheless, important results have been obtained for the class of MSO-definable transductions, as defined by Courcelle. Most notably, it has been shown by J. Engelfriet and H.J. Hoogeboom that MSO-definable (finite) string to string transductions are exactly those transductions defined by deterministic two-way transducers [1]. This result has then been extended to ordered ranked trees by J. Engelfriet and S. Maneth, for the class of linear-size increase macro tree transducers [2] and recently to nested word-to-word transductions [3]. MSO-definable transductions of finite strings have also been characterized by a new automata model, that of (copyless) streaming string transducers, by R. Alur and P. Černý [4].

Copyless streaming string transducers (SST) extend deterministic finite state automata with a finite set of string variables $X, Y, \ldots$. Each variable stores an intermediate string output and can be combined and updated with other variables. Along the transitions, a finite string can be appended or prepended to a variable, and variables can be reset or concatenated. The variable updates along the transitions are formally defined by variable substitutions and the copyless restriction is defined by considering only linear substitutions. Therefore, variable updates such as $X := XX$ are forbidden by the copyless restriction. The SST model has then been extended to other structures such as infinite strings [5], trees [6], and quantitative languages [7].

Two examples of SST are depicted in Figure 1:

- The SST T_0 depicted on the left realizes the function f_0 mapping any input word $u \in \Sigma^*$ to the word $u\overline{u}$, where $\overline{u}$ is the mirror image of the word u. Indeed, when the input word u has been read by the automaton, the variable X contains the word u, while the variable Y contains the word $\overline{u}$. Hence, the final output, defined as XY, is equal to the concatenation $u\overline{u}$. It is worth noting that this SST is copyless.

- The SST T_1 depicted on the right realizes the function f_1 mapping any input word $u = a^n$, with $n \geq 1$, to the output word a^{2^n}. This SST is copyful.

One of the most important and fundamental problem in the theory of transducers is the *equivalence problem*, which asks, given two transducers, whether they realize the same transduction. This problem is well known to be decidable for rational functions, and more generally for MSO-definable transductions [8], and hence for copyless SST (in PSpace, as shown in [9]). The problem gets undecidable when the transducers define binary relations instead of functions: it is already undecidable for non-deterministic finite state transducers [10], a strict subclass of non-deterministic SST. However, it was unknown whether, in the functional case, decidability still holds without the copyless restriction, as mentioned in [11], an extended version of [6].

HDT0L systems allow to define languages by means of morphism iteration: a sequence of indices $i_1, \ldots, i_k$ induces a composition of morphisms (one morphism for each index i_j) which, applied on an

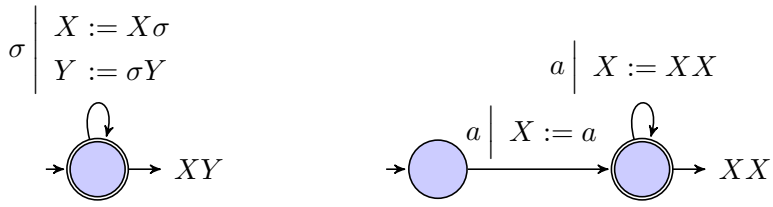

Figure 1. Two streaming string transducers T_0 (left) and T_1 (right).

initial and fixed word, produces a new word. An important result related to HDT0L systems has been obtained in the 80's, see [12]. It states that the equivalence of finite-valued transducers over HDT0L languages is decidable, with unknown complexity.

In this paper, we build a tight connection between HDT0L systems and streaming string transducers. To this end, we introduce a new semantics of HDT0L systems, viewed as transducers. This allows us to prove that (total) copyful SST and HDT0L systems (seen as transducers) have the same expressive power, with back and forth transformations of quadratic complexity. As a corollary of this result, we obtain that the equivalence problem of copyful SST and the equivalence problem of HDT0L systems are inter-reducible, in quadratic time. This result has two consequences:

- first, the decidability of (deterministic) SST equivalence directly follows from [12],
- second, the functionality problem for non-deterministic (copyful) SST is decidable.

Last, we study the class of transductions defined by copyless SST (even in the non-deterministic case), and show that it corresponds to transductions defined by (copyful) SST of *linear size increase* (the length of any output word is linearly bounded by the length of the input). We show that the linear size increase property can be decided in polynomial time: give a (copyful) SST, it is decidable in PTIME whether it defines a transduction of linear size increase. This implies that given a copyful SST (determministic or not), it is decidable in PTIME whether it is equivalent to a copyless SST. This complexity bound is obtained by a reduction to a boundedness problem for products of matrices studied by Mandel and Simon [15].

Related Work The decidability of equivalence for copyful and deterministic SST is a consequence of a result from [13] about the decidability of equivalence for deterministic top-down tree-to-string transducers which, casted to monadic trees (i.e. strings), yields a formalism expressively equivalent to copyful deterministic SSTs. The equivalence problem for copyful deterministic SST has also been shown to be decidable, with a self-contained proof, in [14]. With respect to the equivalence problem, our contribution in this paper is rather to show that HDT0L systems and SST are essentially the same formalisms, and as a consequence, known results [12] can be reused to obtain decidability of equivalence. See also Remark 4.3 for a more detailed comparison with [14].

Organization of the paper. We introduce the models of streaming string transducers and HDT0L systems in Section 2. In Section 3, we prove that these two models are equi-expressive. We apply

this result to prove the decidability of the equivalence of copyful SST and of the functionality of non-deterministic SST in Section 4. Last, in Section 5, we study the subclass of transductions defined by copyful SST of linear-size increase.

2. Preliminaries

For all finite alphabets Σ, we denote by Σ^* the set of finite words over Σ, and by ϵ the empty word. Given two alphabets Σ and Γ, a *transduction* R from Σ^* to Γ^* is a subset of $\Sigma^* \times \Gamma^*$. It is *functional* if it defines a function, i.e. for all $w \in \Sigma^*$, there exists at most one $v \in \Gamma^*$ such that $(w, v) \in R$. The domain of R, denoted by $Dom(R)$, is the set $Dom(R) = \{w \in \Sigma^* \mid \exists v \in \Gamma^*, (w, v) \in R\}$. A transduction is *total* if $Dom(R) = \Sigma^*$. Given two finite alphabets Σ, Γ, a morphism from Σ^* to Γ^* is a mapping $h : \Sigma^* \to \Gamma^*$ such that $h(uv) = h(u)h(v)$ for any two words $u, v \in \Sigma^*$.

2.1. Streaming String Transducers

Let $\mathcal{X}$ be a finite set of variables denoted by $X, Y, \ldots$ and Γ be a finite alphabet. A substitution s is defined as a mapping $s : \mathcal{X} \to (\Gamma \cup \mathcal{X})^*$. Let $\mathcal{S}_{\mathcal{X},\Gamma}$ be the set of all substitutions. Any substitution s can be extended to $\hat{s} : (\Gamma \cup \mathcal{X})^* \to (\Gamma \cup \mathcal{X})^*$ in a straightforward manner. The composition $s_1 \circ s_2$ (or $s_1 s_2$ for short) of two substitutions $s_1, s_2 \in \mathcal{S}_{\mathcal{X},\Gamma}$ is defined as the standard function composition $\hat{s_1} \circ s_2$, i.e. $(s_1 s_2)(X) = (\hat{s_1} s_2)(X) = \hat{s_1}(s_2(X))$ for all $X \in \mathcal{X}$.

Definition 2.1. A non-deterministic streaming string transducer (NSST for short) is a tuple $T = (\Sigma, \Gamma, Q, Q_0, Q_f, \Delta, \mathcal{X}, \rho, s_0, s_f)$ where:

- Σ and Γ are finite alphabets of input and output symbols,
- Q is a finite set of states,
- $Q_0 \subseteq Q$ is a set of initial states,
- $Q_f \subseteq Q$ is a set of final states,
- $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation,
- $\mathcal{X}$ is a finite set of variables,
- $\rho : \Delta \to \mathcal{S}_{\mathcal{X},\Gamma}$ is a variable update function,
- $s_0 : \mathcal{X} \to \Gamma^*$ is the initial function that gives the initial content of the variables,
- $s_f : Q_f \to (\mathcal{X} \cup \Gamma)^*$ is the final output function, which gives what is output for each final state.

The concept of a run of an NSST is defined in an analogous manner to that of a finite state automaton: it is a finite sequence $r \in (Q\Sigma)^*Q$, denoted $r = q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \ldots q_{n-1} \xrightarrow{\sigma_n} q_n$, such that $(q_i, \sigma_{i+1}, q_{i+1}) \in \Delta$ for all $0 \leq i < n$. It is accepting if $q_0 \in Q_0$ and $q_n \in Q_f$. The sequence

of substitutions $\overline{s} = \langle s_i \rangle_{1 \leq i \leq n}$ in $\mathcal{S}_{\mathcal{X},\Gamma}$ of r is defined for all $1 \leq i \leq n$, by $s_i = \rho(q_{i-1}, \sigma_i, q_i)$. We define the *substitution induced by* r as the element $s_r = s_1 \dots s_n$.

If r is accepting, the output of r, denoted by $\mathit{Out}(r) \in \Gamma^*$, is defined as

$$\mathit{Out}(r) = s_0 s_r s_f(q_n) \text{ where } \langle s_i \rangle_{1 \leq i \leq n} \in (\mathcal{S}_{\mathcal{X},\Gamma})^* \text{ is the sequence of substitutions of } r\}.$$

For all words $w \in \Sigma^*$, the output of w by T, denoted by $T(w)$, is

$$T(w) = \{\mathit{Out}(r) \mid r \text{ is an accepting run of } T \text{ on } w\}$$

The *domain* of T, denoted by $\mathit{Dom}(T)$, is defined as the set of words w such that $T(w) \neq \emptyset$. The transduction $[\![T]\!]$ defined by T is the relation from Σ^* to Γ^* given by the set of pairs (w, v) such that $v \in T(w)$.

Deterministic and functional SST A deterministic SST (SST for short) is an NSST such that $|Q_0| = 1$ and for all $p \in Q, \sigma \in \Sigma$, there exists at most one $q \in Q$ such that $(p, \sigma, q) \in \Delta$.When an SST is deterministic, we identify Q_0 with q_0, and given $t \in \Delta$, we write $\rho(t) = s$ instead of $\rho(t) = \{s\}$.

In the following, the streaming string transducers we consider are deterministic, unless they are explicitly stated to be non deterministic.

In [4, 16], the variable updates are required to be *copyless*, i.e. for every variable $X \in \mathcal{X}$, and for every transition $t \in \Delta$, X occurs at most once in $\rho(t)(X_1), \dots, \rho(t)(X_n)$ where $\{X_1, \dots, X_n\} = \mathcal{X}$. One of the main result of [4] is to show that this restriction, as well as determinism, allows one to capture exactly the class of MSO-definable transductions.

It is worth noting that any SST, since it is deterministic, defines a functional transduction. More generally, we say that an NSST T is *functional* if $[\![T]\!]$ is functional. It is known that functional NSST with copyless update are no more expressive than (deterministic) SST with copyless update [16]. We show a similar result for (copyful) SST:

Proposition 2.2. Functional NSST and SST are equi-expressive.

Proof:
Let $T = (\Sigma, \Gamma, Q, Q_0, Q_f, \Delta, \mathcal{X}, \rho, s_0, s_f)$ be a functional NSST. Without loss of generality, we assume that T is trim, *i.e.* every state of T is reachable from some initial state, and co-reachable from an accepting state. Any SST can be made trim by filtering out the states that do not have this property (which is decidable in PTime).

The main idea is to realize a subset construction on T (a similar construction was given in [5]). On states, the subset construction is just as the subset construction for NFA. On variables, one needs to duplicate each variable as many times as the number of states. The invariant property is the following: after reading a word w, if there exists a run ρ of T on w leading to q, then for all $X \in \mathcal{X}$ such that there exists an accepting continuation w' of w (*i.e.* $ww' \in \mathit{Dom}(T)$) whose output uses the content of X after ρ, then X_q and X have the same content after reading w. There might be several runs of T leading to q, but since T is trim and functional, then content of F does not depend on the chosen run. Hence the invariant is well-defined.

Formally, we define an equivalent SST $T' = (\Sigma, \Gamma, Q', q'_0, Q'_f, \Delta', \mathcal{X}', \rho', s'_0, s'_f)$ such that the tuple $(\Sigma, \Gamma, Q', q'_0, Q'_f, \Delta')$ is the DFA resulting from the classical subset construction (in particular $Q' = 2^Q$) and such that:

- $\mathcal{X}' = \mathcal{X} \times Q$ (each variable is denoted by X_q)
- $\forall t' = (Q_1, \sigma, Q_2) \in \Delta', \forall q_2 \in Q_2, \forall X \in \mathcal{X}$, $\rho'(t')(X_{q_2}) = \text{rename}_{q_1}(\rho(t)(X))$ for some $q_1 \in Q_1$ such that $t = (q_1, \sigma, q_2) \in \Delta$, where rename_{q_1} is the identity morphism on Σ and replaces any $Y \in \mathcal{X}$ by Y_{q_1}. As explained before, the functionality of T entails that the choice of q_1 is not important (a different choice would give the same value to X_{q_2}). This choice can be made canonical by using some order on the states of T.
- $\forall P \in Q'_f$, $\rho'(P) = \text{rename}_q(\rho(q))$ for some $q \in P \cap Q_f$. Once again, by functionality of T, the choice of q does not matter and can be made canonical.

□

We say that an SST T is *total* if $[\![T]\!]$ is total. We also show that regarding the equivalence problem, considering total SST is harmless, as one can modify a SST in linear time in order to make it total.

Proposition 2.3. Given two SST T, T', one can build in linear time two total SST T_{tot} and T'_{tot} such that $[\![T]\!] = [\![T']\!]$ iff $[\![T_{tot}]\!] = [\![T'_{tot}]\!]$.

Proof:
Let $\# \notin \Gamma$. Any (partial) SST T can be transformed into an SST T_{tot} that defines the following transduction: $[\![T_{tot}]\!](u) = [\![T]\!](u)$ if $u \in \mathit{Dom}(T)$, and $[\![T_{tot}]\!](u) = \#$ otherwise. This is achieved using a new variable $X_\#$ whose content is always $\#$, and by completing the rules of T by adding an accepting sink state q_{sink}. We also modify final states and the final output function: states that were not final are declared final, and the final output function associates with these states the variable $X_\#$. □

2.2. HDT0L Systems

Lindenmayer introduced in the sixties a formal grammar in order to model the developement process of some biological systems [17]. We consider here a particular class of these systems, called HDT0L systems (HDT0L stands for Deterministic 0-context Lindenmayer systems with Tables and with an additional Homomorphism).

Definition 2.4. (HDT0L System)
An HDT0L system over Σ and Γ is defined as a tuple $H = (\Sigma, A, \Gamma, v, h, (h_\sigma)_{\sigma \in \Sigma})$ where:

- Σ, A and Γ are finite alphabets,
- $v \in A^*$ is the initial word,
- h is a morphism from A^* to Γ^*,

- for each $\sigma \in \Sigma$, h_σ is a morphism from A^* to A^*.

The equivalence problem for HDT0L systems asks, given $H = (\Sigma, A, \Gamma, v, h, (h_\sigma)_{\sigma\in\Sigma})$ and $G = (\Sigma, A, \Gamma, w, g, (g_\sigma)_{\sigma\in\Sigma})$ two HDT0L systems, whether the following formula holds:

$$\forall k \geq 1 \forall \sigma_1 \in \Sigma \ldots \forall \sigma_k \in \Sigma,\ h(h_{\sigma_1} \ldots h_{\sigma_k}(v)) = g(g_{\sigma_1} \ldots g_{\sigma_k}(w)).$$

This problem is known to be decidable [12], with unknown complexity. The original proof of [12] is based on Ehrenfeucht's conjecture and Makanin's algorithm. Honkala provided a simpler proof in [18], based on Hilbert's Basis Theorem.

In order to transfer this decidability result to SST, we introduce a semantics of HDT0L systems in terms of transductions.

Definition 2.5. (Transduction realized by an HDT0L system)
Let $H = (\Sigma, A, \Gamma, v, h, (h_\sigma)_{\sigma\in\Sigma})$ be an HDT0L system. We define $[\![H]\!]$ as a (total) transduction from Σ^* to Γ^* defined by $[\![H]\!](\sigma_1 \ldots \sigma_k) = h(h_{\sigma_1} \ldots h_{\sigma_k}(v))$.

Example 2.6. Let us consider the function f_0 introduced in the introduction. We define an HDT0L $H_0 = (\Sigma, A, \Sigma, v_0, h, (h_\sigma)_{\sigma\in\Sigma})$ such that $[\![H_0]\!] = f_0$, with $A = \{\$_1, \$_2, a, b\}$, $\Sigma = \{a, b\}$, $v_0 = \$_1\$_2$, and the morphisms are defined as follows:

$$\begin{array}{lcl} h: & a \to a \\ & b \to b \\ & \$_1 \to \epsilon \\ & \$_2 \to \epsilon \end{array} \qquad \begin{array}{lcl} h_a: & a \to a \\ & b \to b \\ & \$_1 \to \$_1 a \\ & \$_2 \to a\$_2 \end{array} \qquad \begin{array}{lcl} h_b: & a \to a \\ & b \to b \\ & \$_1 \to \$_1 b \\ & \$_2 \to b\$_2 \end{array}$$

For instance, we have the following derivation:

$$\begin{array}{rcl} [\![H_0]\!](abb) & = & hh_{abb}(\$_1\$_2) = hh_{ab} \circ h_b(\$_1\$_2) = hh_{ab}(\$_1 bb\$_2) = hh_a(\$_1 bbbb\$_2) \\ & = & h(\$_1 abbbba\$_2) = abbbba \end{array}$$

We can now rephrase the result of [12] as follows:

Theorem 2.7. [12] Given two HDT0L systems H_1, H_2 over Σ and Γ, it is decidable whether $[\![H_1]\!] = [\![H_2]\!]$.

In the next section, we show that HDT0L systems and total SST define the same class of transductions.

3. SST and HDT0L systems are equi-expressive

Let Σ and Γ be two alphabets. In this section, we always consider that SST and HDT0L systems are over Σ and Γ. We prove the following theorem:

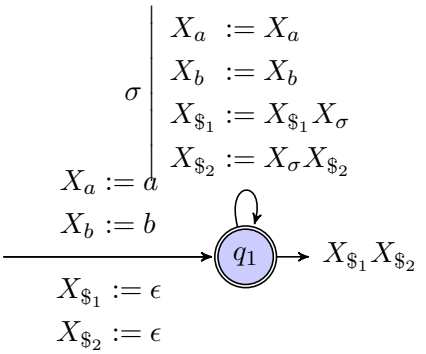

Figure 2. The SST T_2 equivalent to the HDT0L system H_0 of Example 2.6.

Theorem 3.1. HDT0L systems over Σ and Γ and total SST define the same class of transductions. Moreover, the constructions are effective in both directions, in quadratic time.

A direct consequence of this result is:

Corollary 3.2. The equivalence problems for HDT0L systems and for (copyful) streaming string transducers are inter-reducible in quadratic time.

We prove successively the two directions of Theorem 3.1.

Lemma 3.3. For all HDT0L systems H, there exists an equivalent (total) SST T with only one state. Moreover, T can be constructed in linear time.

Proof:
Let $H = (\Sigma, A, \Gamma, v, h, (h_\sigma)_{\sigma\in\Sigma})$ be an HDT0L system. We construct a total SST T over Σ and Γ such that $[\![T]\!] = [\![H]\!]$. The SST has one state q, both initial and accepting. Its set of variables is the set $\mathcal{X} = \{X_a \mid a \in A\}$. Its transitions are defined by $q \xrightarrow{\sigma} q$ for all $\sigma \in \Sigma$.

To define the update functions, we first introduce the morphism $\text{rename}_X : A^* \to \mathcal{X}^*$ defined for all $a \in A$ by $\text{rename}_X(a) = X_a$. Then, the update function ρ is defined, for all $\sigma \in \Sigma$ and $a \in A$ by $\rho((q, \sigma, q))(X_a) = \text{rename}_X(h_\sigma(a))$.

Finally, the initial function is defined by $s_0(X_a) = h(a)$ for all $a \in A$, and the final function s_f by $s_f(q) = \text{rename}_X(v)$. □

Example 3.4. For the HDT0L system H_0 of Example 2.6, we obtain the SST T_2 depicted in Figure 2.

We now prove the converse.

Lemma 3.5. For all total SST T, there exists an equivalent HDT0L system H, which can be constructed in quadratic time.

Proof:
Let $T = (\Sigma, \Gamma, Q, q_0, Q_f, \Delta, \mathcal{X}, \rho, s_0, s_f)$ be a total SST (remember that by default an SST is deterministic). We define an equivalent HDT0L H as follows.

We construct the finite alphabet $A = \{\alpha_q \mid \alpha \in \Gamma \cup \mathcal{X}, q \in Q\}$ (which is of quadratic size in the size of T, this is where the quadratic complexity stated above comes from). For every $q \in Q$, we consider the morphism $\text{subscript}_q : (\Gamma \cup \mathcal{X})^* \to A^*$ defined for all $\alpha \in \Gamma \cup \mathcal{X}$ by $\text{subscript}_q(\alpha) = \alpha_q$.

As T is total, we have that $Q_f = Q$. We consider an enumeration $q_1, \ldots, q_n$ of Q. We define the initial word v as follows:

$$v = \text{subscript}_{q_1}(s_f(q_1)) \ldots \text{subscript}_{q_n}(s_f(q_n))$$

We define the morphism $h : A^* \to \Gamma^*$ as follows:

$$\begin{array}{rlcll} h: & \gamma_{q_0} & \to & \gamma & \text{with } \gamma \in \Gamma \\ & X_{q_0} & \to & s_0(X) & \text{with } X \in \mathcal{X} \\ & \alpha_q & \to & \epsilon & \text{with } q \neq q_0 \text{ and } \alpha \in \Gamma \cup \mathcal{X} \end{array}$$

Last, given $\sigma \in \Sigma$ we define the morphism $h_\sigma : A^* \to A^*$ as follows. Given a state q, we define the set $Pre_q^\sigma \subseteq Q$ as the set of states p such that $(p, \sigma, q) \in \Delta$.

We define: (by convention, the product over the empty set gives the empty word)

$$\begin{array}{rlcl} \forall \gamma_q \in A, & h_\sigma(\gamma_q) & = & \Pi_{p \in Pre_q^\sigma} \text{subscript}_p(\gamma) \\ \forall X_q \in A, & h_\sigma(X_q) & = & \Pi_{p \in Pre_q^\sigma} \text{subscript}_p(\rho(p, \sigma, q)(X)) \end{array}$$

Intuitively, the HDT0L system simulates the computations of the SST in a backward manner, starting from the final states. These computations are encoded using the labelling of symbols by states. One can easily prove by induction on the length of some input word w that after reading w, for every state q, the projection of $h_w(v)$ on the subalphabet $\text{subscript}_q(\Gamma \cup \mathcal{X})$ encodes the run of the SST on w starting in state q, which is unique since T is deterministic. The morphism h then simply erases parts of the computations that did not reach the initial state q_0. □

Connections between SSTs and one-state SSTs We point out the following results, which follow from Lemma 3.3 and Lemma 3.5.

Corollary 3.6. For all SST T, one can construct in quadratic time a (total) SST T' **with only one state** which coincides with T on its domain, i.e. $[\![T]\!] = [\![T']\!]|_{Dom(T)}$.

Proof:
If T is total, then the result is a direct consequence of the successive application of Lemma 3.5 and Lemma 3.3. Note that in this case, T' has only one state.

If T is not total, we make it total as in the proof of Proposition 2, and obtain a total SST S which can be converted into a single state SST S'. □

A direct consequence of the previous corollary is the following result:

Corollary 3.7. For all SST T, one can construct in polynomial time an equivalent SST T' such that the underlying input DFA of T' is the minimal complete DFA recognizing $\mathit{Dom}(T)$.

Proof:
By Corollary 3.6, we can construct a one-state SST T' which coincides with T on its domain. Then, we minimize the underlying input DFA of T (which recognizes $\mathit{Dom}(T)$) into a minimal complete DFA A_{min}. Finally, we construct an SST T'' equivalent to T with underlying DFA A_{min}, as a kind of product of T' and A_{min}: if $A_{min} = (\Sigma, P, p_0, P_f, \delta)$ and $T' = (\Sigma, \Gamma, \{q\}, q, \{q\}, \{(q, \sigma, q) \mid \sigma \in \Sigma\}, \mathcal{X}, \rho, s_0, s_f)$, then we let $T'' = (\Sigma, \Gamma, P, p_0, P_f, \delta, \mathcal{X}, \rho'', s_0, s''_f)$ where:

- $\rho''(p, \sigma, p') = \rho(q, \sigma, q)$ for all $(p, \sigma, p') \in \delta$,
- $s''_f(p_f) = s_f(q)$ for all $p_f \in P_f$.

□

Remark 3.8. By these results, any total SST is equivalent to some one-state SST. A direct transformation of total SST into one-state SST was also given in [14] (Proposition 8). Here, we obtain this result as a direct corollary of the back and forth transformations between SST and HDT0L systems.

4. Applications: SST Equivalence and Functionality of NSST

Based on the correspondence between SST and HDT0L systems, and the fact that the HDT0L system equivalence problem is decidable, we show that that the SST equivalence and functionality problems are decidable.

Theorem 4.1.
1. Given two SST T and T', it is decidable whether they are equivalent, i.e. $[\![T]\!] = [\![T']\!]$.
2. Given an NSST T, it is decidable whether T is functional.

Proof:
The first statement is straightforward by Theorem 3.1, Theorem 2.7 and Proposition 2.3.

To prove the second statement, we reduce the functionality problem to the equivalence of two (deterministic) SST T_1 and T_2. Let $T = (\Sigma, \Gamma, Q, Q_0, Q_f, \Delta, \mathcal{X}, \rho, s_0, s_f)$ be an NSST. We extend the alphabet Σ with pairs of rules of T as follows: $\Sigma' = \Sigma \times \Delta^2$. Now, T_1 and T_2 are defined as the square of T: they run on words w' over Σ', and make sure that the sequence of transitions are valid runs of T on the Σ-projections of w'. In addition, T_i simulates T on the $(i+1)$-th component, for all $i = 1, 2$, by following the transitions defined on the input letters. Clearly, T_1 and T_2 have the same domain, are deterministic, and are equivalent iff all pairs of accepting runs of T on the same input word produce the same output, i.e., iff T is functional. The conclusion follows from statement 1. □

Conversely, as equivalence problems of HDT0L and of SST are inter-reducible, we can also use known results about SST equivalence to prove new results on HDT0L. We say that an HDT0L system $H = (\Sigma, A, \Gamma, v, h, (h_\sigma)_{\sigma \in \Sigma})$ is *copyless* if for each $\sigma \in \Sigma$, the morphism h_σ is linear, *i.e.* each element $a \in A$ appears at most once in $\{h_\sigma(a) \mid a \in A\}$.

Theorem 4.2. For copyless HDT0L systems, the equivalence problem can be solved in PSpace.

Proof:
The construction of Lemma 3.3 which transforms an HDT0L system into an equivalent SST, when applied to a copyless HDT0L system, yields a copyless SST. Inspecting this construction, for all $\sigma \in \Sigma$ and $a \in A$, we have $\rho((q,\sigma,q))(X_a) = \text{rename}_X(h_\sigma(a))$. Let $\{a_1,\dots,a_n\} = A$ be some enumeration of A. Then

$$\begin{aligned}\rho((q,\sigma,q))(X_{a_1})\dots\rho((q,\sigma,q))(X_{a_n}) &= \text{rename}_X(h_\sigma(a_1))\dots\text{rename}_X(h_\sigma(a_n)) \\ &= \text{rename}_X(h_\sigma(a_1\dots a_n))\end{aligned}$$

Since h_σ is copyless, $h_\sigma(a_1\dots a_n)$ does not contain twice the same symbol, and hence the word $\text{rename}_X(h_\sigma(a_1\dots a_n))$ does not contain twice the same variable. In other words, $\rho((q,\sigma,q))$ is copyless. □

Remark 4.3. (Complexity)
It has been shown in [14] that the equivalence problem for (copyful) SST is in the Ackermann complexity class. Unlike our result on the decidability of this problem, which is based on the known decidability of the HDT0L equivalence problem [12] (without any bound given), the latter result is self-contained and goes via a reduction to the equivalence problem of polynomial automata. The purpose of our paper was rather to show that HDT0L systems and SST are essentially the same objects (and hence that decidability of SST equivalence can be directly obtained using known results). Combining the result of [14] with our correspondence between HDT0L and SSTs (Theorem 3.1), one obtains that the equivalence problem for HDT0L systems lies in the complexity class Ackermann. To the best of our knowledge, no upper bound was known for this problem. However, it is still open whether this bound is tight.

5. SST of Linear Size Increase

A transduction $R \subseteq \Sigma^* \times \Gamma^*$ is *linear size increase (LSI)* if for some constant $K \geq 0$, for all $(u,v) \in R$, $|v| \leq K|u|$. For instance, transductions defined by ϵ-free finite state transducers (without variables) are LSI, by taking K as the length of the longest word occurring on a single transition. It is not difficult to see that it is also the case of transductions defined by copyless NSST:

Proposition 5.1. ([16])
Any transduction defined by a copyless NSST is linear size increase.

Proof:
It suffices to take $K = |\mathcal{X}|M$ where $|\mathcal{X}|$ is the number of variables of the NSST, and M is the length of the maximal number of output symbols occurring in the rhs of a substitution $\rho(t)$, for $t \in \Delta$ a transition, or the initial function s_0, or the terminal function s_f. Formally, $K = \max\{K_1, K_2, K_3\}$ where for π_Γ the projection on alphabet Γ, we have:

$$\begin{aligned}K_1 &= \max\{|\pi_\Gamma(\rho(t)(X))| \mid t \in \Delta, X \in \mathcal{X}\} \\ K_2 &= \max\{|s_0(X)| \mid X \in \mathcal{X}\} \\ K_3 &= \max\{|\pi_\Gamma(s_f(q))| \mid q \in Q_f\}\end{aligned}$$

□

An objective of this section is to show that the converse also holds for transductions defined by NSST: any LSI transduction defined by an NSST is definable by some copyless NSST. We also give an effective characterisation of the LSI transductions defined by NSST, which allows one to decide in PTIME whether a given NSST is actually equivalent to a copyless NSST. In the functional case, the subclass of copyless NSST (which is known to be equivalent to copyless deterministic SST [5]) is of great interest, as it exactly corresponds to the class of regular functions which enjoys multiple characterizations (MSO transductions and deterministic two-way transducers for instance) and has been widely studied in the literature (see for instance [19] for a survey). Again in the functional case, similar definability questions have been investigated for tree transductions defined by deterministic macro tree transducers, where the LSI (semantical) restriction is shown to be decidable [20].

To prove the equivalence between NSST of linear size increase and copyless NSST, we use an intermediate class called bounded copy NSST [5]. Intuitively, the bounded copy restriction allows for a bounded number of copies of the same variable. It is formally defined through a natural notion of *variable flow*, which forms what is called the *flow transition monoid* [21, 22].

Let T be an SST with states Q and variables $\mathcal{X}$. The *flow transition monoid* M_T of T is a set of square matrices over non-negative integers enriched with a new absorbing element $\perp$. The matrices are indexed by elements of $Q \times \mathcal{X}$. Given an input word u, the image of u in M_T is the matrix m such that for all states p, q and all variables X, Y, $m[p, X][q, Y] = n \in \mathbb{N}$ if, and only if, there exists a run r of T on u from state p to state q, and X occurs n times in $s_r(Y)$, where s_r is the substitution induced by r. In this case, we say that (p, X) flows n times to (q, Y) along u and we may also write $(p, X) \xrightarrow{u|n} (q, Y)$. Respectively, $m[p, X][q, Y] = \perp$ iff there is no run of T on u from state p to state q. Then, an NSST T is *bounded copy* if M_T is finite. Note that T is copyless iff M_T contains only matrices whose values are in $\{0, 1, \perp\}$.

Theorem 5.2. Given a transduction $R \subseteq \Sigma^* \times \Gamma^*$ defined by some (copyful) NSST, the following statements are equivalent:

1. R is definable by a bounded-copy NSST
2. R is definable by a copyless NSST
3. R is linear size increase

Moreover, it is decidable in PTIME whether a given (copyful) NSST defines a transduction of linear size increase.

Proof:
Let $T = (\Sigma, \Gamma, Q, q_0, Q_f, \Delta, \mathcal{X}, \rho, s_0, s_f)$ be an NSST. Consider a run $q_0 \xrightarrow{\sigma_1} q_1 \dots q_{n-1} \xrightarrow{\sigma_n} q_n = p$ of T starting in the initial state. By definition of the semantics of T, after this run, one can associate, in state p, with variable X the content $\nu(X) \in \Gamma^*$ defined as $s_0 s_1 \dots s_n(X)$ where $s_i = \rho(q_{i-1}, \sigma_i, q_i)$. We use the notation $q_0 \xrightarrow{\sigma_1 \dots \sigma_n} (p, \nu)$ to describe this fact, and may remove the label $\sigma_1 \dots \sigma_n$ when it is useless.

We also say that a pair $(p, X) \in Q \times \mathcal{X}$ is *co-accessible* whenever there exists a word u, a state $q_f \in Q_f$ and a variable $Y \in \mathcal{X}$ such that Y appears in the final output $s_f(q_f)$ and $(p, X) \xrightarrow{u|n} (q_f, Y)$ with $n \geq 1$.

We introduce the following notations:

- we let $\mathit{val}(p, X) = \{\nu(X) \in \Gamma^* \mid \text{ there exists a run } q_0 \to (p, \nu)\}$
- we let $\mathit{Inf} = \{(p, X) \mid (p, X) \text{ is co-accessible and } \mathit{val}(p, X) \text{ is infinite}\}$

We now consider the set of square matrices M_T^{Inf} indexed by elements of *Inf* and obtained by restricting the matrices of M_T to *Inf*. Consider three pairs $(p, X), (q, Y), (s, Z)$ such that $(p, X) \xrightarrow{u_1|n_1} (q, Y) \xrightarrow{u_2|n_2} (s, Z), n_1, n_2 \geq 1$ and $(p, X), (s, Z) \in \mathit{Inf}$. Then we necessarily have $(q, Y) \in \mathit{Inf}$. This implies that the elements of M_T^{Inf} can be generated from the (finite) set of matrices $\{m_a^{\mathit{Inf}} \mid a \in \Sigma\}$, where, for each $a \in \Sigma$, m_a is the image of letter a in the monoid M_T, and m_a^{Inf} is the restriction of m_a to *Inf*.

We claim that the four following properties are equivalent:

1. $[\![T]\!]$ is definable by a bounded-copy NSST,
2. $[\![T]\!]$ is definable by a copyless NSST,
3. $[\![T]\!]$ is linear size increase,
4. M_T^{Inf} is finite.

Before proving the claim, we show that it implies decidability, thanks to property 4. First, the set *Inf* is computable in polynomial time (fixpoint computation in the set of pairs $(p, X) \in Q \times \mathcal{X}$). Second, the (finite) set of matrices $\{m_a^{\mathit{Inf}} \mid a \in \Sigma\}$ can be computed in polynomial time and the finiteness of M_T^{Inf} can thus be decided in polynomial time, as shown by Mandel and Simon in [15].

We turn to the proof of the claim, and to this end we prove the implications $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$.

$1 \Rightarrow 2$ We prove that every bounded-copy NSST is equivalent to some copyless NSST. This property has been proven for deterministic SST in [5, 22]. We explain how to lift it to non-deterministic SST. We consider the deterministic SST $\bar{T}$ defined over the alphabet Δ. It is obtained from T by replacing any transition $t = (p, a, q)$ by the transition (p, t, q). Then, we let $\bar{U}$ be the equivalent copyless SST over the alphabet Δ, obtained thanks to [5, 22]. Last, we let U be the copyless NSST obtained from $\bar{U}$ by modifying the transitions as follows: if $e = (s_1, t, s_2)$ is a transition of $\bar{U}$ with $t = (p, a, q)$, then we replace e by the transition (s_1, a, s_2). It is immediate to verify that U is a copyless NSST equivalent to T.

$2 \Rightarrow 3$ This implication has been proven in Proposition 5.1.

$3 \Rightarrow 4$ We prove the contraposition, by showing that if the property 4. of the claim is not satisfied, then $[\![T]\!]$ is not LSI. We thus assume that the set M_T^{Inf} is infinite. By the characterization of this property proven in [15] by Mandel and Simon, two cases may occur:

1. there exists $(p, X) \in \mathit{Inf}$ such that $(p, X) \xrightarrow{u|n} (p, X)$ with $n \geq 2$, for some word u,
2. there exist two distinct nodes $(p, X), (q, Y) \in \mathit{Inf}$ such that $(p, X) \xrightarrow{u|n_1} (p, X)$, $(p, X) \xrightarrow{u|n_2} (q, Y)$ and $(q, Y) \xrightarrow{u|n_3} (q, Y)$, with $n_1, n_2, n_3 \geq 1$, for some word u.

In the first case, as $(p, X) \in \mathit{Inf}$, we can choose two words u_1 and u_2 such that $q_0 \xrightarrow{u_1} (p, \nu)$ with $\nu(X) \neq \epsilon$, and (p, X) is co-accessible by the word u_2. Then, for every $i \geq 0$, we consider the word $v_i = u_1 u^i u_2$. We have $|[\![T]\!](v_i)| \geq |\nu(X)|.n^i \geq 2^i$. This proves that $[\![T]\!]$ is not LSI.

In the second case, for every $n \geq 1$, one has $(p, X) \xrightarrow{u^n|m} (q, Y)$ for some $m \geq n$. As $(p, X), (q, Y) \in \mathit{Inf}$, there exist words u_1, u_2 such that $q_0 \xrightarrow{u_1} (p, \nu)$ and (q, Y) is co-accessible by the word u_2. Then, for every $i \geq 0$, we consider the word $v_i = u_1 u^i u_2$. We have $|[\![T]\!](v_i)| \geq i.|\nu(X)| \geq \frac{|v_i| - |u_1 u_2|}{|u|}.|\nu(X)|$. As $(p, X) \in \mathit{Inf}$, $\nu(X)$ can be chosen to be arbitrarily large, hence $[\![T]\!]$ is not LSI.

$4 \Rightarrow 1$ The constraint expressed by property 4. of the claim precisely states that, with respect to pairs $(p, X) \in \mathit{Inf}$, the NSST is bounded-copy. We can build an equivalent bounded-copy NSST as follows. For each state p and each variable X such that (p, X) is co-accessible and not in *Inf*, we can compute the finite set of possible valuations of X, denoted as $\nu_1, \ldots, \nu_k$. Then, we consider duplicates of state p in which the valuation of X is fixed. Now, for every transition leaving state p, the substitution is modified so as to remove the reference to X, and replace it by the valuation of X. Similarly, if a variable X is such that (p, X) is not co-accessible, the substitutions can be modified so as to ensure that the only possible valuation of X in state p is ϵ. A consequence of these modifications is that for every flow of variables involving a pair not in *Inf*, the value will be 0, while flows between elements of *Inf* are unchanged. As a consequence, the resulting NSST has a finite flow transition monoid, and is thus bounded-copy.

□

6. Conclusion

Our results establish a bridge between the theory of SST and the theory of systems of iterated morphisms. It allows to solve an interesting open problem for copyful streaming string transducers, namely the decidability of the equivalence problem. For transductions defined by non-deterministic SST, we have proven the decidability of the functionality problem, as well as the linear size increase property (which entails definability by a copyless non-deterministic SST). Our results are summarised in Fig. 3, in which the arrows have to be interpreted as the definability problems of deciding whether given a transducer in the starting class, it is equivalent to a transducer in the target class.

We hope that these positive decidability results will pave the way to a further study of the class of copyful SST. As future work, we want to investigate what the theory of iterated morphisms can bring to the theory of SST, and conversely, in terms of tight complexity results.

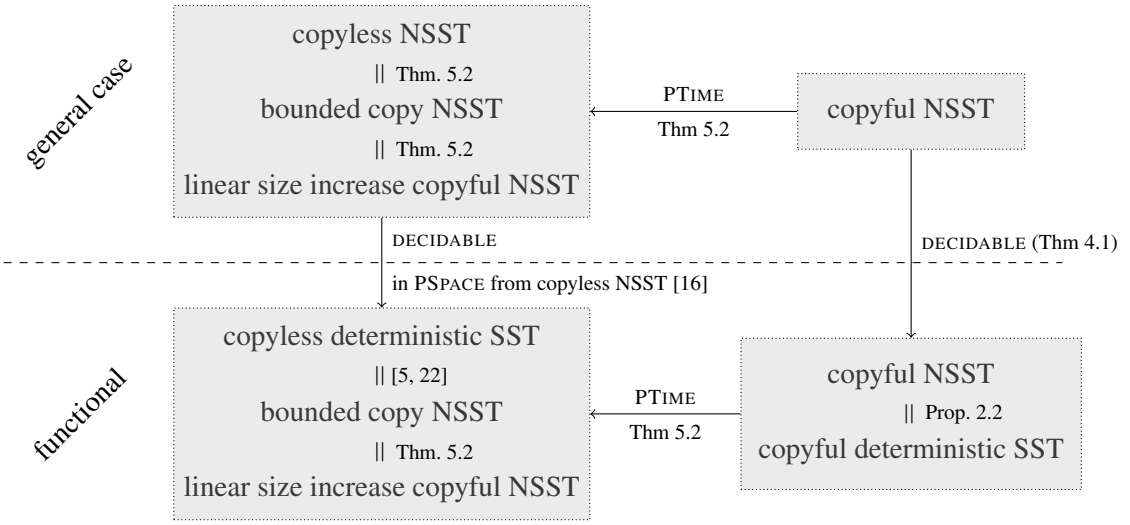

Figure 3. Comparison between transducer classes and decidability of the corresponding definability problems

References

[1] Engelfriet J, Hoogeboom HJ. MSO definable string transductions and two-way finite-state transducers. *ACM Trans. Comput. Logic*, 2001. **2**:216–254.

[2] Engelfriet J, Maneth S. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Information and Computation*, 1999. **154**(1):34–91.

[3] Dartois L, Filiot E, Reynier PA, Talbot JM. Two-Way Visibly Pushdown Automata and Transducers. In: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16. ACM, 2016 pp. 217–226.

[4] Alur R, Černý P. Expressiveness of streaming string transducers. In: FSTTCS, volume 8. 2010 pp. 1–12.

[5] Alur R, Filiot E, Trivedi A. Regular Transformations of Infinite Strings. In: LICS. IEEE. ISBN 978-1-4673-2263-8, 2012 pp. 65–74.

[6] Alur R, D'Antoni L. Streaming Tree Transducers. In: ICALP (2), volume 7392 of *LNCS*. Springer, 2012 pp. 42–53.

[7] Alur R, D'Antoni L, Deshmukh JV, Raghothaman M, Yuan Y. Regular Functions and Cost Register Automata. In: 28th Annual ACM/IEEE Symp. on Logic in Computer Science, LICS 2013. IEEE Computer Society. ISBN 978-1-4799-0413-6, 2013 pp. 13–22. doi:10.1109/LICS.2013.65. URL `http://doi.ieeecomputersociety.org/10.1109/LICS.2013.65`.

[8] Engelfriet J, Maneth S. The equivalence problem for deterministic MSO tree transducers is decidable. *Inf. Process. Lett.*, 2006. **100**(5):206–212.

[9] Alur A, Černý P. Streaming transducers for algorithmic verification of single-pass list-processing programs. In: POPL. 2011 pp. 599–610.

[10] Griffiths TV. The Unsolvability of the Equivalence Problem for Lambda-Free Nondeterministic Generalized Machines. *J. ACM*, 1968. **15**(3):409–413. doi:10.1145/321466.321473.

[11] Alur R, D'Antoni L. Streaming Tree Transducers. *CoRR*, 2011. **abs/1104.2599**. URL http://arxiv.org/abs/1104.2599.

[12] Culik II K, Karhumäki J. The Equivalence of Finite Valued Transducers (On HDT0L Languages) is Decidable. *Theor. Comput. Sci.*, 1986. **47**(3):71–84. doi:10.1016/0304-3975(86)90134-9. URL http://dx.doi.org/10.1016/0304-3975(86)90134-9.

[13] Seidl H, Maneth S, Kemper G. Equivalence of Deterministic Top-Down Tree-to-String Transducers is Decidable. In: IEEE 56th Annual Symp. on Foundations of Computer Science, FOCS 2015. IEEE Computer Society, 2015 pp. 943–962.

[14] Benedikt M, Duff T, Sharad A, Worrell J. Polynomial automata: Zeroness and applications. In: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017. 2017 pp. 1–12.

[15] Mandel A, Simon I. On Finite Semigroups of Matrices. *Theor. Comput. Sci.*, 1977. **5**(2):101–111.

[16] Alur R, Deshmukh JV. Nondeterministic Streaming String Transducers. In: ICALP, volume 6756 of *LNCS*. Springer, 2011 pp. 1–20.

[17] Lindenmayer A. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 1968. **18**:280–315.

[18] Honkala J. A short solution for the HDT0L sequence equivalence problem. *Theor. Comput. Sci.*, 2000. **244**(1-2):267–270. doi:10.1016/S0304-3975(00)00158-4. URL https://doi.org/10.1016/S0304-3975(00)00158-4.

[19] Filiot E, Reynier PA. Transducers, logic and algebra for functions of finite words. *SIGLOG News*, 2016. **3**(3):4–19. doi:10.1145/2984450.2984453. URL http://doi.acm.org/10.1145/2984450.2984453.

[20] Engelfriet J, Maneth S. Macro Tree Translations of Linear Size Increase are MSO Definable. *SIAM J. Comput.*, 2003. **32**(4):950–1006.

[21] Filiot E, Krishna SN, Trivedi A. First-order Definable String Transformations. In: 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, volume 29 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014 pp. 147–159.

[22] Dartois L, Jecker I, Reynier PA. Aperiodic String Transducers. In: Developments in Language Theory - 20th International Conference, DLT 2016, volume 9840 of *Lecture Notes in Computer Science*. Springer, 2016 pp. 125–137.