Two-way automata and transducers with planar behaviours are aperiodic

Lê Thành Dũng (Tito) Nguyễn^{1*} Camille Noûs² Cécilia Pradic³

- ¹ Laboratoire de l'informatique du parallélisme (LIP), École normale supérieure de Lyon, France
- ² Laboratoire Cogitamus
- ³ Department of Computer Science, Swansea University, Wales

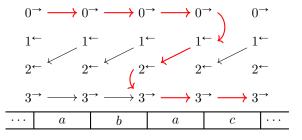
We consider a notion of planarity for two-way finite automata and transducers, inspired by Temperley-Lieb monoids of planar diagrams. We show that this restriction captures star-free languages and first-order transductions.

Keywords: aperiodic regular functions, planar diagrams

1 Introduction

This paper considers a notion of *planarity* – in the topological sense – for two-way deterministic finite automata. To the best of our knowledge, little has been said about it; Hines (2006) suggested the definition in a workshop talk, with motivations coming from outside automata theory, but he did not investigate it further. Unlike the notion of "planar automaton" that seems to occur more frequently in the literature, e.g. (Book and Chandra, 1976; Bonfante and Deloup, 2019), ours is concerned not with the state transition chart, but with another graphical representation of the behaviour of a two-way automaton.

Example 1.1. The following diagram corresponds to a two-way automaton – defined in Example 2.3 – running on an input containing a factor abca. Assuming that the automaton arrives on this factor from the left in state 0^{\rightarrow} , its run will continue as indicated by the path in red, exiting on the right in state 3^{\rightarrow} .



^{*}Supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

ISSN subm. to DMTCS © 2023 by the author(s) Distributed under a Creative Commons Attribution 4.0 International License

By "planarity", we refer to the lack of crossings⁽ⁱ⁾ in such diagrams drawn on the plane. This is the case in the above example.

Our main result entails that planar deterministic two-way automata recognize exactly the class of *star-free* languages. We actually show a strengthening of this result on *transducers* – finite-state machines that compute string-to-string functions.

Theorem 1.2. Let $f: \Sigma^* \to \Gamma^*$. The following are equivalent:

- f is a first-order transduction.
- f is computed by some planar deterministic two-way finite transducer.
- f is computed by some planar reversible(ii) two-way finite transducer.

By forgetting the output information of a two-way transducer, one may obtain a two-way automaton recognizing the domain of the partial function computed by the transducer. Thus, we immediately get Corollary 1.3 below for automata and languages. To make sense of this, keep in mind that first-order transductions are to regular functions – those computed by deterministic two-way transducers (2DFTs), without the planarity condition – what star-free languages are to regular languages (see e.g. Muscholl and Puppis, 2019, p. 9). (While the name "first-order" refers to logic, we will not use logic in this paper.)

Corollary 1.3. *Let* $L \subseteq \Sigma^*$. *The following are equivalent:*

- L is a star-free language.
- L is recognized by some planar deterministic two-way finite automaton (planar 2DFA).
- L is recognized by some planar reversible two-way finite automaton.

Plan of the paper After a discussion of motivations and related work, we introduce in Section 2 the automata model and the notion of planarity under scrutiny. We then embark on the proof of Theorem 1.2, which contains two main thrusts corresponding to the equivalence between the first and last items.

Section 3 shows that functions computed by planar 2DFTs are first-order transductions. To do so, we appeal to a result of Carton and Dartois (2015) stating that *aperiodic* 2DFTs are exactly as expressive as first-order transductions (extending Schützenberger's seminal theorem relating star-free languages and aperiodic monoids, whose history is recounted in (Straubing, 2018)). In order to show that the monoid of behaviours of a planar 2DFT is necessarily aperiodic, we simply embed it into a generic monoid that depends only on the (ordered and directed) state space of the 2DFT rather than the fine details of its transition functions. The key feature that makes this so simple is that planarity is a property that is preserved by composition of transitions (whereas aperiodic elements of a monoid are, in general, not closed under product).

Then, in Section 4, we show that the functions realizable by planar reversible two-way transducers (2RFTs) form a class that is closed under composition and contains some "simple" functions. We can then conclude by factoring any first-order transduction into a composition of such simple functions (this involves the decomposition theorem of Krohn and Rhodes (1965)). Here the interesting aspect is that planarity of transitions is preserved by known methods (Dartois et al., 2017) to compose 2RFTs.

⁽i) There is an unfortunate clash of terminology here with the *crossing sequences* from Shepherdson (1959). Therefore, we shall avoid the word "crossing" altogether.

⁽ii) A reversible machine is both deterministic and "co-deterministic", as we will explain.

1.1 Related work

Planar diagram monoids and geometry of interaction A device amounting to an "undirected" variant of our planar 2RFA was first proposed by Hines (2006). He had previously connected (Hines, 2003) the monoids of 2DFA and 2RFA behaviours to a category-theoretic account of the so-called *geometry of interaction* (GoI)⁽ⁱⁱⁱ⁾ semantics of linear logic. Then planarity comes up naturally when considering GoI for non-commutative linear logic, as noted for instance by Abramsky (2007) who also explains how the monoids of endomorphisms of his categorical semantics coincide with Kauffman's (1990) diagram monoids. (The monoids of behaviours in (Hines, 2006) are finite quotients^(iv) of Kauffman monoids, called "Jones monoids" or "Temperley-Lieb^(v) monoids" (see e.g. East, 2021).)

In an earlier work (Nguyễn and Pradic, 2020), we exhibited a connection between non-commutative linear logic and star-free languages. It is thanks to this that we could guess what effect the planarity restriction would have on the computational power of two-way automata and transducers. Relatedly, Auinger (2014) has shown that Jones monoids generate precisely the pseudovariety of aperiodic monoids – a result that is morally very close to our characterization of star-free languages by planar 2RFA, and whose proof uses the algebraic version of the Krohn–Rhodes decomposition.

Relationship with previous automata models As we already said in our opening words, our planar behaviours are entirely unrelated to the *planar automata* of Book and Chandra (1976). Their notion of planarity relates to usual one-way machines when regarded as graphs, so it includes, for instance, the minimal DFA of the non-star-free language $(aa)^*$.

That said, a special case of our planarity condition has appeared before in the literature on automata. Indeed, when instantiated to deterministic one-way automata, it becomes *monotonicity*: all transitions are partial monotone functions with respect to a common linear order on states. The variant with total instead of partial transitions has been introduced and dubbed "monotonic automata" by Ananichev and Volkov $(2004)^{(vi)}$ in the context of synchronization problems. The study of the monoid \mathcal{O}_n of all partial monotone functions on the finite linear order $\{1 < 2 < \cdots < n\}$ – which contains the monoids of behaviours of (partial) monotonic automata, i.e. one-way planar automata, with n states – is actually even older. In particular, Higgins (1995) exhibits an aperiodic monoid that does not divide any \mathcal{O}_n for $n \in \mathbb{N}$, which entails that some star-free language cannot be recognized by any partial monotonic automaton. Thus, two-wayness is necessary to recognize all star-free languages with planar automata.

As for our definition of reversibility for automata, it does coincide with the widespread one in the literature. Dartois, Fournier, Jecker, and Lhote (2017) were the first to study it in the context of two-way transducers, but there had been several earlier works proving that reversibility does not affect the power of two-way automata, and of their generalizations to trees and graphs – for an overview, see (Martynova and Okhotin, 2023, Introduction). Finally, on a technical level, this paper is very much tied to the existing theory of machine models for first-order transductions as developed in (Filiot, Krishna, and Trivedi, 2014; Carton and Dartois, 2015; Dartois, Jecker, and Reynier, 2018; Bojańczyk, Daviaud, and Krishna, 2018).

⁽iii) Further connections between categorical versions of the GoI and automata theory have been investigated by Katsumata (2008) and by Clairambault and Murawski (2019). We may also note that *inverse monoids* have been linked to both GoI (Goubault-Larrecq, 2011) and two-way automata (Baudru, Dando, Lhote, Monmege, Reynier, and Talbot, 2022).

⁽iv) An element of a Kauffman monoid consists not only of a planar diagram, but also of a natural number of free-floating cycles, making the monoid countably infinite; the quotient throws these cycles away.

⁽v) The *Temperley-Lieb algebras* mentioned in the titles of (Hines, 2006; Abramsky, 2007) admit a presentation based on Kauffman monoids. They appear in knot theory and statistical physics.

⁽vi) See for instance (Fernau, Haase, and Hoffmann, 2022, Section 7) for an overview of subsequent work on monotonic automata.

2 Preliminaries

Notations Alphabets Σ , Γ consist of non-empty finite sets. By convention, we assume that we have distinguished elements \triangleright and \triangleleft that do not belong to any alphabet and we set $\Sigma_{\triangleright \triangleleft} = \Sigma \cup \{\triangleright, \triangleleft\}$. Given two sets A and B, we write $A \rightarrow B$ for the set of partial functions from A to B and $A \rightleftharpoons B$ for partial injections from A to B. Finally, given a relation \rightarrow , we write \rightarrow^* for its reflexive transitive closure.

Definition 2.1. A two-way nondeterministic automaton (2NFA) \mathcal{A} over the alphabet Σ consists of:

- A finite set Q of states equipped with a direction map ρ: Q → {-1,1}.
 The pair (Q, ρ) is sometimes called a directed set of states and we write Q[→] for ρ⁻¹(1) and Q[←] for ρ⁻¹(-1). Elements of Q[→] are called forward states and those of Q[←] backward states.
- An initial state $q_0 \in Q^{\rightarrow}$ and a set^(vii) of final states $F \subseteq Q$.
- A family of transition relations $\delta \colon \Sigma_{\triangleright \triangleleft} \to \mathcal{P}(Q \times Q)$.

A configuration of the automaton is a triple $(u,q,v) \in \Sigma^*_{\triangleright \triangleleft} \times Q \times \Sigma^*_{\triangleright \triangleleft}$. The immediate successor relation \rightarrow_{δ} is defined as follows

```
\begin{array}{lll} (u,q,av) & \rightarrow_{\delta} & (ua,r,v) & \text{whenever } (q,r) \in \delta(a), q \in Q^{\rightarrow} \text{ and } r \in Q^{\rightarrow} \\ (u,q,av) & \rightarrow_{\delta} & (u,r,av) & \text{whenever } (q,r) \in \delta(a), q \in Q^{\rightarrow} \text{ and } r \in Q^{\leftarrow} \\ (ua,q,v) & \rightarrow_{\delta} & (u,r,av) & \text{whenever } (q,r) \in \delta(a), q \in Q^{\leftarrow} \text{ and } r \in Q^{\leftarrow} \\ (ua,q,v) & \rightarrow_{\delta} & (ua,r,v) & \text{whenever } (q,r) \in \delta(a), q \in Q^{\leftarrow} \text{ and } r \in Q^{\rightarrow} \end{array}
```

A word $w \in \Sigma^*$ is said to be accepted by \mathcal{A} if we have $(\varepsilon, q_0, \triangleright w \triangleleft) \to_{\delta}^* (\triangleright w \triangleleft, q_f, \varepsilon)$ for some $q_f \in F$. That is, a run of the 2NFA starts in the initial state on the left of the input string, and to accept a word is to reach a final state after overrunning the right-hand side marker.

We consider only *deterministic* machines in this paper, i.e. we require that the transition relations are actually partial functions. In fact, we shall study a further restriction that is *reversibility*. At the formal level this amounts to requiring that the inverse of the transition functions also be partial functions.

Definition 2.2. An automaton $(Q, \rho, q_0, F, \delta)$ over Σ is called *deterministic* (resp. *reversible*) if δ maps each letter $a \in \sigma$ to a *partial function* (resp. *injection*).

The languages recognized by 2DFA are exactly the regular languages – that is, 2DFA are equivalent to one-way automata (Rabin and Scott, 1959, Theorem 15). Nowadays, the best-known construction for translating 2DFA to one-way DFA is due to Shepherdson (1959); it is the one that we shall work with^(viii).

Example 2.3. We define here the two-way automaton whose behaviour on a sample input is depicted in Example 1.1. The input alphabet is $\{a, b, c\}$, and the set of states is $Q^{\rightarrow} = \{0^{\rightarrow}, 3^{\rightarrow}\}$, $Q^{\leftarrow} = \{1^{\leftarrow}, 2^{\leftarrow}\}$ and $F = \{3^{\rightarrow}\}$. The transition function is specified the following input/output table:

	a		c		\triangleleft
<u>0</u> →	0-	0→	1←	0→	
1←	2←	2←	2←		
2←		$3 \rightarrow$			
$3 \rightarrow$	$3 \rightarrow$	$3 \rightarrow$	1← 2← 3→	$3 \rightarrow$	$3 \rightarrow$

 $[\]overline{\text{(vii)}}$ We could take F to be a singleton or replace q_0 by a set without changing any result of the paper.

⁽viii) Technically speaking, we use two-sided behaviours as in (Birget, 1989) so as to discuss monoids while Shepherdson only considered one-sided behaviours; the combinatorics remain similar.

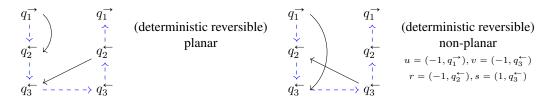


Fig. 1: Planar and non-planar behaviours for $Q^{\rightarrow} = \{q_1^{\rightarrow}\}$ and $Q^{\leftarrow} = \{q_2^{\leftarrow}, q_3^{\leftarrow}\}$.

It recognizes the language $\{a,b\}^*b\{a,b\}c\{a,b,c\}^*$ by first looking for the first occurrence of c, then doubling back to check a b occurs two positions earlier. This automaton:

- is deterministic, as seen from the fact that each cell of the table contains at most one output;
- is not reversible, because the transition specified for b has two different states sent to 3^{\rightarrow} .

We now formalize the central restriction on 2DFA at play in this paper, which is planarity. As per Figure 1, it means that we want to enforce that each transition profile, as drawn in our example pictures, does not contain any sort of crossings. Example 2.3 is a (paradigmatic) example of a planar 2DFA.

Definition 2.4. Let (Q, ρ) be a directed set of states. For any binary relation $f \subseteq Q \times Q$, we define the *transition profile* $\mathcal{G}(\rho, f)$ as the directed graph with vertices $Q \times \{-1, 1\}$ and edges

$$\{(q, -\rho(q)) \to (r, \rho(r)) \mid (q, r) \in f\}$$

Definition 2.5. Let Q be a finite set. For every total order < over Q, extend < to a total order over $Q \times \{-1,1\}$ by setting (q,i) < (r,j) if and only if either

$$i = -1$$
 and $j = 1$ or $q < r$ and $i = j = 1$ or $q > r$ and $i = j = -1$

Call a transition $f \in \mathcal{P}(Q^2)$ planar with respect to the order < and polarity function ρ if there is no ordered sequence of vertices u < r < v < s such that either $u \to v$ or $v \to u$ is in $\mathcal{G}(\rho, f)$ and either $r \to s$ or $s \to r$ is in $\mathcal{G}(\rho, f)$.

A 2NFA $(Q, \rho, q_0, F, \delta)$ over alphabet Σ is called planar when there is a total order < over Q such that, for every input letter $a \in \Sigma$, $\delta(a)$ is planar with respect to < and ρ .

Remark 2.6. This definitions says that we must choose a total order on states which will constrain how all transitions will be drawn in the plane. This is stronger than merely requiring each transition profile to individually be a planar graph. In fact, it corresponds to the planarity of a *combinatorial map* (see Mohar and Thomassen, 2001, Chapter 4) whose underlying graph is obtained from the transition profile by forgetting the edge directions.

Our results will primarily concern two-way *transducers*, a notion that generalizes two-way automata by realizing binary relations between Σ^* and Γ^* instead of languages, i.e. sets of words. Since we are going to restrict our attention to deterministic machines the relations will always be partial functions $\Sigma^* \to \Gamma^*$.

Definition 2.7. A two-way nondeterministic transducer (2NFT) \mathcal{T} with input alphabet Σ and output alphabet Γ consists of a tuple $(Q, \rho, q_0, F, \delta)$ with (Q, ρ) a directed set of states, $q_0 \in Q^{\rightarrow}$, $F \subseteq Q$ and, most importantly, with $\delta \colon \Sigma \to \mathcal{P}(Q \times \Gamma^* \times Q)$.

The notion of configuration is defined in the same way as Definition 2.1, except that now the immediate successor relation \xrightarrow{w}_{δ} is tagged by a word $w \in \Gamma^*$. It is defined in the expected way, analogous to the case of 2NFA. We then say that (u,v) is in the relation realized by \mathcal{T} if we have a sequence of configurations $(\varepsilon, q_0, \triangleright u \lhd) \xrightarrow{v_1} \dots \xrightarrow{v_n} (\triangleright u \lhd, q_f, \varepsilon)$ such that $q_f \in F$ and $v = v_1 \dots v_n$.

One can consider the 2NFA $[\mathcal{T}]$ over Σ obtained by forgetting the output, i.e. $(Q, \rho, q_0, F, \lfloor \delta \rfloor)$ where $\lfloor \delta \rfloor(a) = \{(q,r) \mid \exists w. \ (q,w,r) \in \delta\}$ and check that the language recognized by $\lfloor \mathcal{T} \rfloor$ is exactly the domain of the relation defined by \mathcal{T} . We can use the map $\mathcal{T} \mapsto \lfloor \mathcal{T} \rfloor$ to lift all notions relevant to automata to transducers: we say that \mathcal{T} is deterministic/reversible/planar when $|\mathcal{T}|$ is.

A 2DFT \mathcal{T} with input alphabet Σ^* and output alphabet Γ^* defines a partial function $\Sigma^* \to \Gamma^*$. Let us write $\mathcal{T} \colon \Sigma^* \to \Gamma^*$ and conflate \mathcal{T} with the partial function it computes in notations when convenient.

3 Planar 2DFTs can only compute first-order transductions

To show that planar 2DFTs can only compute first-order transductions, we use a characterization of the latter due to Carton and Dartois (2015). To recall it, we need to start with *aperiodic monoids*, the algebraic counterpart of star-free languages (as mentioned in the introduction).

Definition 3.1. A finite monoid M is aperiodic if there is $n \in \mathbb{N}$ such that for every $x \in M$, $x^{n+1} = x^n$.

To make full use of this algebraic condition, we associate finite monoids to our machines. The idea is to look at the transition profiles generated by the transition relations δ and their composition.

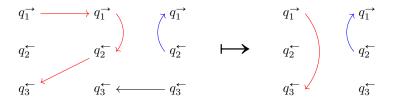


Fig. 2: Composition of two transitions.

Definition 3.2. Consider a directed set of states (Q, ρ) . Given two transitions $f, g \in \mathcal{P}(Q^2)$, consider the digraph $\mathcal{G}(\rho, f, g)$ by taking the disjoint union of $\mathcal{G}(\rho, f)$ and $\mathcal{G}(\rho, g)$ and identifying $Q \times \{1\}$ in the former with $Q \times \{-1\}$ in the latter. (Formally, $\mathcal{G}(\rho, f, g)$ is defined as having vertices $Q \times \{-1, 0, 1\}$ and the edge $(u, i) \to (v, j)$ if and only if either $(u, 2i + 1) \to (v, 2j + 1)$) is an edge of $\mathcal{G}(\rho, f)$ or $(u, 2i - 1) \to (v, 2j - 1)$) is in $\mathcal{G}(\rho, g)$.)

The composition f*g is then defined as the transition relating q to r if and only if there is a path from $(q, -\rho(q))$ to $(r, \rho(r))$ in the digraph $\mathcal{G}(\rho, f, g)$. This way, $\mathcal{G}(\rho, f*g)$ contains an edge $u \to v$ if and only if $\mathcal{G}(\rho, f, g)$ contains a path from u to v (see Figure 2 for an illustration). Composition can be shown to be associative, with the identity relation as unit element (see e.g. Birget, 1989): this yields a finite monoid $\mathbf{Beh}(Q, \rho)$ with carrier $\mathcal{P}(Q^2)$.

Given a 2NFA $\mathcal{A} = (Q, \rho, q_0, F, \delta)$ over alphabet Σ , define $\mathbf{Beh}(\mathcal{A})$ as the least submonoid of $\mathbf{Beh}(Q, \rho)$ containing $\{\delta(a) \mid a \in \Sigma\}$. For 2NFTs, set $\mathbf{Beh}(\mathcal{T}) = \mathbf{Beh}(|\mathcal{T}|)$.

When \mathcal{T} is a 2DFT, (Carton and Dartois, 2015, Def. 4) define its transition monoid $\mathbf{Beh}(\mathcal{T})$ in the same way^(ix), so we may use their characterization as our reference definition^(x) of first-order transductions.

Definition 3.3. A first-order transduction is a string-to-string function computed by a 2DFT \mathcal{T} whose monoid of behaviours $\mathbf{Beh}(\mathcal{T})$ is aperiodic.

Next, we want to show that the monoid of behaviours of a planar 2DFT is necessarily aperiodic. As submonoids of aperiodic monoids are also aperiodic, it suffices to show that the following class of monoids defined from directed set of states are aperiodic.

Definition 3.4. Given a directed set of states (Q, ρ) and a total order < on Q, call $\mathbf{TL}(Q, \rho, <)$ the subset of $\mathbf{Beh}(Q, \rho)$ containing only transitions that are both deterministic (i.e. partial functions) and planar with respect to ρ and <.

Theorem 3.5. $\mathbf{TL}(Q, \rho, <)$ is an aperiodic submonoid of $\mathbf{Beh}(Q, \rho)$.

Proof: First, we show that it is a submonoid. We leave the reader to mechanically check that the unit element is deterministic planar. Concerning closure under composition, it is topologically intuitive: $^{(xi)}$ for $f,g \in \mathbf{TL}(Q,\rho,<)$, the combinatorial map (cf. Remark 2.6) $\mathcal{G}(\rho,f,g)$ is planar because it is obtained by gluing two planar maps along a common boundary, identifying vertices; by determinism, it consists of a disjoint union of rooted trees that can be drawn in the plane in a pairwise non-crossing fashion, and this drawing is homotopy equivalent to $\mathcal{G}(\rho,f*g)$ (fixing the vertices $Q\times\{-1,1\}$), by contracting the trees to stars. Note that determinism is necessary here: Figure 3 shows that planar non-deterministic behaviours do not compose.

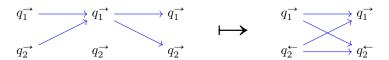


Fig. 3: A non-planar composition of two planar non-deterministic behaviours.

Let us now show aperiodicity. Let $f \in \mathbf{TL}(Q, \rho, <)$ and let us partition f into

$$f_{\mathfrak{S}} = f \cap (Q^{\longrightarrow} \times Q^{\leftarrow}) \qquad f_{\mathfrak{S}} = f \cap (Q^{\leftarrow} \times Q^{\longrightarrow}) \qquad f_{\longleftrightarrow} = f \backslash (f_{\mathfrak{S}} \cup f_{\mathfrak{S}})$$

We shall consider variants of f_{\bigcirc} and f_{\bigcirc} where "partial identities" have been added:

$$f_{\dashv} = f_{\circlearrowleft} \cup \{ (q,q) \in Q^2 \mid \nexists r : (q,r) \in f_{\circlearrowleft} \} \qquad f_{\vdash} = f_{\circlearrowleft} \cup \{ (q,q) \in Q^2 \mid \nexists r : (q,r) \in f_{\circlearrowleft} \}$$

The point is that we have the following identity (which, for now, does not depend on planarity):

$$f = f_{\dashv} * f_{\leftrightarrow} * f_{\vdash} \quad \text{hence} \quad \forall n \in \mathbb{N}, \ f^{n+1} = f_{\dashv} * (f_{\leftrightarrow} * f_{\vdash} * f_{\dashv})^n * f_{\leftrightarrow} * f_{\vdash}$$

⁽ix) The presentation is different, but the result is easily seen to be isomorphic.

⁽x) The main result of (Carton and Dartois, 2015) is that this definition is equivalent to another one based on first-order logic, hence the name of the function class. The papers cited at the very end of the introduction provide several alternative characterizations.

⁽xi) Our argument here is somewhat informal. It could be made rigorous using the Heffter–Edmonds–Ringel principle (cf. Mohar and Thomassen, 2001, Theorem 3.2.4) relating combinatorial maps and actual topology. In principle, there should probably also exist a purely combinatorial proof from Definition 2.5, but it would likely be cumbersome.

Let $g = f_{\leftrightarrow} * f_{\vdash} * f_{\dashv}$. The above identity shows that if the sequence $(g^n)_{n \ge 1}$ is eventually constant, then so is $(f^n)_{n \ge 1}$ – and the latter is what we want, according to the definition of aperiodicity.

Observe that $g \subseteq Q^{\rightarrow} \times Q^{\rightarrow} \cup Q^{\leftarrow} \times Q^{\leftarrow}$. Thus, g is the union of a partial function $g_{\rightarrow} : Q^{\rightarrow} \to Q^{\rightarrow}$ with the transpose of a partial function $g_{\leftarrow} : Q^{\leftarrow} \to Q^{\leftarrow}$, and these two components do not interact in a composition. Since g is planar (as a product of planar transitions), these partial functions are both monotone. So, to conclude, it suffices to invoke the well-known fact that the monoid of partial monotone functions on a finite total order is aperiodic (see e.g. Nguyễn and Pradic, 2020, Lemma 3.2).

Remark 3.6. An alternative proof of aperiodicity can also be obtained by characterizing some of *Green's relations* over $\mathbf{TL}(Q, \rho, <)$, which we sketch now. The \mathcal{R} -class of any element of $f \in \mathbf{TL}(Q, \rho, <)$ (which corresponds to the left ideal generated by f) is fully determined by $f_{\mathfrak{D}}$ and the set $f_r \in \mathcal{P}(\mathcal{P}(Q))$ defined as

$$f_r = \{f_{\leftrightarrow}(\{q\}) \mid q \in Q^{\leftarrow}\} \cup \{f_{\leftrightarrow}^{-1}(\{q\}) \mid q \in Q^{\rightarrow}\}\}$$

The intuition is that f_{\odot} and f_r describes the left part of a transition diagram, plus information on which vertices might end up being identified by f on the other side of the diagram. This is the only data that can be preserved by an invertible multiplication on the right, and conversely any element is reachable by multipying by a suitable planar partial bijection. Similarly, the \mathcal{L} -class (right ideal generated by f) is determined by f_{\odot} and a set f_l defined in a dual manner. Then it can be observed that the resulting joint map $f \mapsto (f_{\odot}, f_l, f_{\odot}, f_r)$ is injective because of planarity. So in particular, every class \mathcal{H} -class (an intersection of an \mathcal{R} and a \mathcal{L} -class) is a singleton, which is equivalent to the monoid being aperiodic.

Note that for a monoid of reversible transitions, the problem can reduce to the aperiodicity of the Jones monoids. In that setting the Green relations can be characterized using similar (although slightly simpler) maps as shown in Lau and FitzGerald (2006, Theorem 3.5, (iii-v)).

In any case, we can deduce the first half of the main theorem.

Corollary 3.7. Every planar 2DFT computes a first-order transduction.

Proof: This reduces to Theorem 3.5, since the behaviour monoid $\mathbf{Beh}(\mathcal{T})$ of a planar transducer \mathcal{T} with directed set of states (Q, ρ) ordered by < will always be a submonoid of $\mathbf{TL}(Q, \rho, <)$.

4 Planar 2RFTs compute all first-order transductions

The objective of this section is to establish the other half of Theorem 1.2.

Theorem 4.1. Any first-order transduction can be computed by a planar 2RFT.

Let us walk through the proof of the theorem, delegating important lemmas to further subsections. We use two results from the literature to characterize first-order transductions as compositions of simpler functions. The first is Krohn and Rhodes's (1965) decomposition theorem, specialized to the following standard statement found in e.g. (Nguyễn and Pradic, 2020, Theorem 4.8) about *sequential functions* – these are the functions computed by *one-way* deterministic transducers (i.e. 2DFTs with $\rho(Q) = \{1\}$).

Theorem 4.2 (Aperiodic Krohn–Rhodes decomposition). Any aperiodic sequential function $f: \Sigma^* \to \Gamma^*$ can be realized as a composition $f = f_1 \circ \ldots \circ f_n$ (with $f_i: \Delta_i^* \to \Delta_{i-1}^*$, $\Delta_0 = \Gamma$ and $\Delta_n = \Sigma$) where each function f_i is computed by some aperiodic sequential transducer with 2 states.

We can show that planar 2RFTs compute all aperiodic sequential functions by establishing that that the functions computed by planar 2RFTs are closed under composition (Lemma 4.4), and by providing an encoding of aperiodic 2-state transducers (Lemma 4.5).

Then we use another result, which is derived from the characterization of first-order transductions by aperiodic streaming string transducers (Filiot et al., 2014; Dartois et al., 2018).

Lemma 4.3 (rephrasing of (Bojańczyk, Daviaud, and Krishna, 2018, Lemma 4.8)^(xii)). Every first-order transduction can be decomposed as $f \circ reverse \circ g \circ reverse \circ h$ where f is computed by a monotone register transducer and the functions g and h are aperiodic sequential.

The reverse function simply refers to a function that reverses its input and can easily be implemented with a 3-state reversible planar transducer. Monotone register transducers are a variation of single-state streaming stream transducers (Alur and Černý, 2010) with a monotonicity constraint on register assignments. So we can conclude the proof using an encoding of monotone register transducers in planar 2RFTs (Lemma 4.7) and re-using the closure under composition of planar 2RFTs.

Our remaining task is to prove the aforementioned lemmas. The following subsections will be concerned with Lemma 4.4, Lemma 4.5 and Lemma 4.7 respectively.

4.1 Closure under composition

The proof of (Dartois et al., 2017, Theorem 1) gives an efficient construction to compose 2RFTs reminiscent of the wreath product of semigroups. The basic idea is that one may simply take a cartesian product of the states, and a transition is composed of:

- a first component that simulates the first transducer in the composition,
- and a second component that simulates the second transducer on the production given by the first
 component, signaling to the simulation of the first transducer to proceed with its computation or
 rewind according to its needs (it is because of rewinding that reversibility is required).

It turns out that this construction also readily preserves planar behaviours. We illustrate this in Figure 4 and thus can show the following.

Lemma 4.4. Let $\mathcal{T}_1: \Sigma^* \to \Gamma^*$ and $\mathcal{T}_2: \Gamma^* \to \Delta^*$ be planar 2RFTs. There is a planar 2RFT computing the composition of the partial functions $\mathcal{T}_2 \circ \mathcal{T}_1$.

Proof: If $\mathcal{T}_1 = (Q, \rho_1, q_0, F_1, \delta_1)$ and $\mathcal{T}_2 = (R, \rho_2, r_0, F_2, \delta_2)$, the composition is computed by

$$\mathcal{T}_2 * \mathcal{T}_1 = (Q \times R, \rho_1 \rho_2, (q_0, r_0), F_1 \times F_2, \delta)$$

with $\delta: \Sigma_{\triangleright \triangleleft} \to \mathcal{P}((Q \times R) \times \Delta^* \times (Q \times R))$ defined as a union $\delta_{\to} \cup \delta_{\hookrightarrow} \cup \delta_{\hookrightarrow}$, where each component is the smallest such that

•
$$\delta_{\rightarrow}(a) \ni ((q,r), w, (q',r'))$$
 if $\rho_2(r) = \rho_2(r') = 1$, $(\varepsilon, r, v) \xrightarrow{w} \overset{*}{\delta_{\delta_2}}(v, r', \varepsilon)$ and $(q, v, q') \in \delta'_1(a)$

⁽xii) We use the theorem numbering from the official published version of (Bojańczyk et al., 2018), which is significantly different from the numbering in the arXiv version. Our rephrasing uses the fact that an aperiodic *rational* function can be decomposed as $reverse \circ g \circ reverse \circ h$ where g and h are aperiodic sequential; this decomposition, analogous to a theorem of Elgot and Mezei (1965) on general rational functions, can be proved directly starting from the definition of rational function with monoids that is used by Bojańczyk et al. (2018).

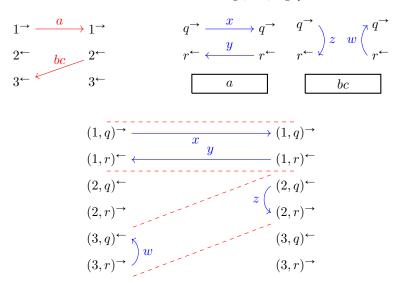


Fig. 4: Three planar transitions, one with output in $\{a, b, c\}$, two with inputs in $\{a, b, c\}$, and the transition obtained by composing the transducers they originate from.

•
$$\delta_{\leftarrow}(a) \ni ((q',r),w,(q,r')) \text{ if } \rho_2(r) = \rho_2(r') = -1, (v,r,\varepsilon) \xrightarrow{w} \overset{*}{\delta_2} (\varepsilon,r',v) \text{ and } (q,v,q') \in \delta_1'(a)$$

•
$$\delta_{\mathfrak{S}}(a) \ni ((q,r), w, (q,r'))$$
 if $\rho_2(r) = -\rho_2(r') = 1$, $(\varepsilon, r, v) \xrightarrow{w} \overset{*}{\delta_2} (\varepsilon, r', v)$ and $(q, v, q') \in \delta_1'(a)$

$$\bullet \ \delta_{ \circlearrowleft}(a) \ni ((q',r),w,(q',r')) \text{ if } \rho_2(r) = -\rho_2(r') = -1, (v,r,\varepsilon) \xrightarrow{\ w \ } ^*_{\delta_2} (v,r',\varepsilon) \text{ and } (q,v,q') \in \delta_1'(a)$$

where $\delta_1': \Sigma \to \mathcal{P}(Q \times \Gamma_{\rhd \lhd}^* \times Q)$ is the obvious modification of δ_1 required to surround the output of \mathcal{T}_1 by the end markers \rhd and \lhd . The basic idea is that in each of the four cases, we observe the global movement of \mathcal{T}_2 on the output produced by \mathcal{T}_1 (determined by the subscripts \to , \leftarrow , \hookrightarrow or \circlearrowleft), we move forward or backwards in the run of \mathcal{T}_1 over the input accordingly.

The construction is exactly the same as in the proof of (Dartois et al., 2017, Theorem 1); it is sound and preserves both determinism and reversibility for exactly the same reasons. It remains to check that the obtained transitions can still be made planar. It is the case: if δ_1 and δ_2 induce planar transitions with respect to two orders over Q and R respectively, then δ induce planar transitions with respect to their lexicographic order over $Q \times R$ as per Definition 2.5.

To do so, let us assume that we have two edges $e_0 = \{(q_0, r_0), p_0\}, ((q_2, r_2), p_2)\} \in \mathcal{G}(\rho_1 \rho_2, \lfloor \delta \rfloor r(a))$ and $e_1 = \{(q_1, r_1), p_1), ((q_3, r_3), p_3)\} \in \mathcal{G}(\rho_1 \rho_2, \lfloor \delta \rfloor (a))$ such that $((q_i, r_i), p_i) < ((q_j, r_j), p_j)$ whenever i < j. We then proceed via a case analysis:

- If it is the case that every $\rho_2(r_i)$ coincide, then projecting the R components yield edges either in $\mathcal{G}(\rho_1, |\delta_1|(a))$. In that case, we contradict the planarity of $\delta_1(a)$.
- If three of the q_i s coincide, by determinism and reversibility of \mathcal{T}_1 , it means that projecting the Q components away yields edges in $\mathcal{G}(\rho_2, \lfloor \delta_2 \rfloor(v))$ where v is the output of \mathcal{T}_1 when it reads a from state q. But then we contradict the planarity of transitions in \mathcal{T}_2 .

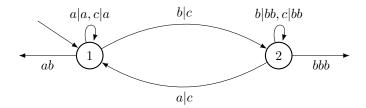


Fig. 5: A two-state aperiodic sequential transducer over the alphabet $\{a,b,c\}$ A transition $\partial(a)(q)=(q',w)$ is pictured as an arrow from q to q' labelled by a|w.

• And with that, we have covered all the cases; the first item allows us to show that either e_0 or e_1 is in $\mathcal{G}(\rho_1\rho_2, \lfloor \delta_{\bigcirc} \cup \delta_{\bigcirc} \rfloor(a))$. If e_0 is an edge of $\mathcal{G}(\rho_1\rho_2, \lfloor \delta_{\bigcirc} \cup \delta_{\bigcirc} \rfloor(a))$, it means in particular that $q_0 = q_2$ and $\rho_2(r_0) \neq \rho_2(r_2)$. So necessarily $p_0 = p_1 = p_2$, so we further have $q_0 = q_1 = q_2$, so we were actually in the second case. The case where e_1 is an edge of $\mathcal{G}(\rho_1\rho_2, \lfloor \delta_{\bigcirc} \cup \delta_{\bigcirc} \rfloor(a))$ is handled similarly.

4.2 Two-state aperiodic sequential transducers

We now turn to proving the following.

Lemma 4.5. Every two-state aperiodic sequential transducer can be computed by a planar 2RFT.

Before embarking on this proof, let us recall what is an aperiodic sequential transducer.

Definition 4.6. A (total deterministic) sequential transducer with input alphabet Σ and output alphabet Γ is a tuple (Q, q_0, ∂, o) where Q is a finite set of states, $q_0 \in Q$ is an initial state, $o: Q \to \Gamma^*$ is a final output function and $\partial: \Sigma \to Q \to Q \times \Gamma^*$ is a transition function. Let us write $\partial_1: \Sigma \to Q \to Q$ and $\partial_2: \Sigma \to Q \to \Gamma^*$ for each component of ∂ .

The function ∂ is extended to $\partial^* : \Sigma^* \to Q \to Q \times \Gamma^*$ defined by its components ∂_1^* and ∂_2^* :

$$\partial^*(\varepsilon)(q) = (q, \varepsilon) \qquad \partial_1^*(wa) = \partial_1(a) \circ \partial_1^*(w) \quad \text{and} \quad \partial_2^*(wa)(q) = \partial_2^*(w)\partial_2(a) \left(\partial_1^*(w)(q)\right)$$

Such a sequential transducer then defines a function $\Sigma^* \to \Gamma^*$ by $w \mapsto \partial_2^*(w) \cdot o(\partial_1^*(q_0))$.

This sequential transducer is said to be *aperiodic* if the monoid generated by the transition function, which is the closure of $\{\partial_1(a) \mid a \in \Sigma\}$ under composition, is aperiodic.

An example of a two-state aperiodic transducer is pictured in Figure 5. One can see that the monoid generated by its transition function is actually the (unique) maximal aperiodic submonoid of $\{1,2\}^{\{1,2\}}$ obtained by removing the only non-trivial bijection.

We now have all the definitions required to embark in the construction that will prove Lemma 4.5; before giving the details, one may want to take a look at the translation of the transducer of Figure 5, which is ran over an input in Figure 6.

Proof of Lemma 4.5: Without loss of generality, suppose that we start with an input transducer of shape $(\{1,2\},1,\partial,o)$ with input alphabet Σ and output alphabet Γ . We build a planar reversible 2DFT with the

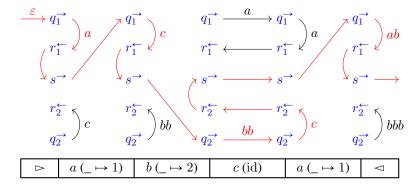


Fig. 6: A run of the 2DFT obtained by the construction in the proof of Lemma 4.5 applied to the sequential transducer pictured in Figure 5.

directed state-space $\{q_1^{\rightarrow}, r_1^{\leftarrow}, s^{\rightarrow}, r_2^{\rightarrow}, q_2^{\rightarrow}\}$, the initial state q_1^{\rightarrow} and final state s^{\rightarrow} . Its transition function δ is then defined as per the table given in Figure 7: note that for $a \in \Sigma$, $\delta(a)$ is determined by $\partial(a)$, $\delta(\triangleright)$ is fixed (essentially because we fixed that 1 was the initial state of our input sequential transducer) and $\delta(\triangleleft)$ is determined by o.

As a visual inspection of Figure 7 reveals, all those transitions are reversible and planar for the ordering $q_1^{\rightarrow} < r_1^{\leftarrow} < s^{\rightarrow} < r_2^{\rightarrow} < q_2^{\rightarrow}$ on the states.

Now it only remains to be checked that the partial functions compted by the two machines are the same. The proof of this basically relies on the following claim that holds by induction over $w \in \Sigma^*$.

$$\text{If } \partial^*(w)(1) = (i,u) \text{, then } (\varepsilon,q_1^{\rightarrow}, \rhd w) \xrightarrow{\ \ u \ \ }^* (\rhd w,q_i^{\rightarrow},\varepsilon) \text{ and } (\rhd w,r_i^{\leftarrow},\varepsilon) \xrightarrow{\ \ \varepsilon \ \ }^* (\rhd w,s^{\rightarrow},\varepsilon)$$

With that proven, we may build the following run over a word w

$$(\varepsilon,q_1^{\rightarrow},\rhd w\lhd) \xrightarrow{\partial_2^*(w)(1)} {}^*_{\delta} (\rhd w,q_i^{\rightarrow},\lhd) \xrightarrow{o(i)} {}_{\delta} (\rhd w,r_i^{\leftarrow},\lhd) \xrightarrow{\varepsilon} {}^*_{\delta} (\rhd w,s^{\rightarrow},\lhd) \xrightarrow{\varepsilon} {}_{\delta} (\rhd w\lhd,s^{\rightarrow},\varepsilon)$$
 where $i=\partial_1^*(w)(1)$, which allows to conclude.

4.3 Monotone register transducers

We now prove the final lemma required to conclude the section.

Lemma 4.7. Any string-to-string function computed by a monotone register transducer may be computed by a planar reversible 2DFT.

Before proving this, we first need to define what is a *monotone register transducer*. This will correspond to a restricted subclass of *copyless streaming string transducers* (Alur and Černý, 2010). Those machines go through their inputs in a single left-to-right pass, storing infixes of their outputs in registers that they may update by performing concatenations of previously stored values and constants. Here we will further impose that our machines have no control states, that the output corresponds to a single register and that the register updates satisfy a monotonicity condition in addition to being copyless.

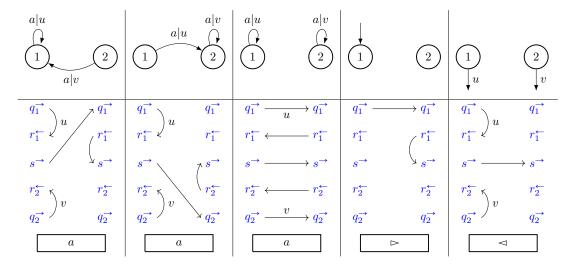


Fig. 7: Definition of the translation from an aperiodic sequential transducer to a transition function for a 2DFT computing the same function. The first row represents features of the sequential transducer and the second a corresponding transition in the obtained 2DFT. In the first three columns, transitions $\partial(a)$ are depicted in diagramatic form, then the initial state in the fourth column and the output function in the last column.

Definition 4.8. A register update for the set of registers R over the alphabet Γ is a map $\sigma \colon R \to (R \cup \Gamma)^*$ (we assume $R \cap \Gamma = \emptyset$). It induces the map $\sigma^{\dagger} \colon (\Gamma^*)^R \to (\Gamma^*)^R$ defined this way: the r-component of the output is obtained by replacing, in $\sigma(r)$, each register occurrence r' by the r'-component of the input.

We shall use a suggestive notation using an assignment symbol for register updates: we write

$$\sigma = \begin{cases} X := XbYc \\ Y := abZ \\ Z := c \end{cases} \quad \text{for} \quad \sigma(X) = XbYc, \ \sigma(Y) = abZ, \ \sigma(Z) = c$$

(with $R = \{X, Y\}$ and $a, b, c \in \Gamma$). For this update, we have $(\sigma^{\dagger}(\overrightarrow{w}))_X = w_X b w_Y c$ for any $\overrightarrow{w} \in (\Gamma^*)^R$. Now to limit ourselves to regular first-order transductions, we need to apply some restrictions to the kind of register updates we have.

Definition 4.9. Let $R = \{r_1 < \cdots < r_n\}$ be a finite and totally ordered set of registers. A register update $\sigma \colon R \to (R \cup \Gamma)^*$ is *copyless monotone* when the list of register occurrences in the concatenation $\sigma_a(r_1) \dots \sigma_a(r_n)$ is increasing (i.e. strictly monotone).

Our previous example of register update is copyless monotone for X < Y < Z: the occurrences of registers in $XbYc \cdot abZ \cdot c$ form the increasing sequence X, Y, Z. Some examples of register updates that do not satisfy this condition include

$$X := XX \qquad \begin{cases} X := Y \\ Y := X \end{cases}$$

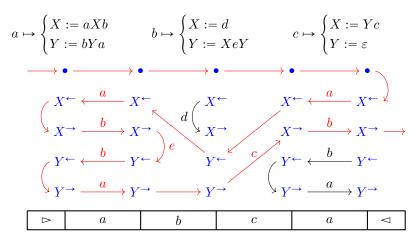


Fig. 8: A monotone register transducer and a run of the corresponding planar reversible 2DFT (where we write X^{\leftarrow} for (X, -1) and X^{\rightarrow} for (X, 1) for readability).

Definition 4.10. A monotone register transducer $\Sigma^* \to \Gamma^*$ consists of the following:

- A finite totally ordered set of registers R.
- For each input letter $a \in \Sigma$, a copyless monotone register update σ_a for R over Γ .

It computes the function $a_1 \dots a_n \in \Gamma^* \mapsto (\sigma_{a_n}^\dagger \circ \dots \circ \sigma_{a_1}^\dagger(\overrightarrow{\varepsilon}))_X$ where $\overrightarrow{\varepsilon} \in (\Gamma^*)^R$ is the tuple whose components are all equal to the empty word, and X is the minimum element of R.

We can now turn to proving that they can be simulated by planar reversible transducers; this will be a simpler variation of the construction found in the proof of (Dartois et al., 2018, Theorem 6) based on their notion of *output structure* that determines how to translate register updates into two-way behaviours; an example of applying it is given in Figure 8.

Proof of Lemma 4.7: Given a monotone register transducer with an ordered set of registers R and update function σ and input alphabet Σ , we will build a 2DFT with state space $\{\bullet\} \cup R \times \{-1,1\}$ (for some $\bullet \notin R \times \{-1,1\}$). The direction ρ over states is going to be given by the second projection and $\rho(\bullet) = 1$. Its transition relation δ will be the smallest relation satisfying the following, where r_0 is the smallest register of R:

- $\{(\bullet, \varepsilon, \bullet), ((r, -1), \varepsilon, (r, 1))\} \subseteq \delta(\triangleright)$ and $\{(\bullet, \varepsilon, (r_0, -1)), ((r_0, 1), \varepsilon, (r_0, 1))\} \subseteq \delta(\triangleleft)$
- $((r,-1), \sigma(a), (r,1)) \in \delta(a)$ if $a \in \Sigma$ and $\sigma_a \in \Gamma^*$
- $((r,-1), w, (r',-1)) \in \delta(a)$ if $a \in \Sigma$ and $wr' \in \Gamma^*R$ is a prefix of $\sigma_a(r)$
- $((r,1),w,(r',-1)) \in \delta(a)$ if $a \in \Sigma$ and $rwr' \in R\Gamma^*R$ is an infix of $\sigma_a(r'')$ for some r''
- $((r,1),w,(r',1)) \in \delta(a)$ if $a \in \Sigma$ and $rw \in R\Gamma^*$ is a suffix of $\sigma_a(r')$
- $(\bullet, \varepsilon, \bullet) \in \delta(a)$ if $a \in \Sigma$

The basic idea is that we treat an entry in the state (r, -1) as a query of what is the value stored in the register r, which is going to be computed along the run that will eventually loop back to state (r, 1). Each case in the definition above corresponds to producing part of the output described by the register updates and querying previous register values accordingly. Then to initialize the run, we simply need to move to the rightmost position and get into state $(r_0, -1)$; this is achieved by starting from the state \bullet , which we take as the initial state. Then we set $(r_0, 1)$ to be the unique final state.

Due to the fact that the register updates are copyless, we obtain that the transitions are reversible and deterministic. The soundness of the construction is easily derived by the invariant explained above, which can be proven by induction.

As for planarity, the order on states will simply correspond to the lexicographic order over $R \times \{-1,1\}$, supplemented by stating that \bullet is below all the other states. The monotonicity constraint on register updates then clearly ensures planarity of all transitions $\delta(a)$ for $a \in \Sigma$, and the extremal transitions $\delta(\triangleright)$ (which only relates successive states in the order except for \bullet) and $\delta(\triangleleft)$ (that has two elements) are easily seen to be planar as well.

5 Some perspectives

We hope to have demonstrated that the notions of planar deterministic/reversible automaton/transducer is an interesting new way to capture star-free languages and first-order transductions.

As we explained in the "related work" section, this notion first arose (Hines, 2006) from the geometry of interaction semantics of non-commutative linear logic. Our interest in this semantics is part of a broader plan to link fragments of the (linear) λ -calculus and automata theory that is exposed in (Nguyễn, 2021). More specifically, we aim to improve the results of (Nguyễn and Pradic, 2020) – characterizing star-free languages by means of a non-commutative λ -calculus – to include first-order regular transductions using semantic methods.

Something worth investigating might be to extend the planarity condition to a suitable notion of *tree-walking automata* (see e.g. Bojańczyk and Colcombet, 2008) and tree-walking transducers outputting strings or trees. One natural question for instance would be to ask whether reversible tree-to-tree walking transducers are also closed under composition, and whether planarity would still be retained.

Acknowledgments We would like to thank Sam van Gool, Denis Kuperberg, Paul-André Melliès and Noam Zeilberger for stimulating feedback on this work.

References

Samson Abramsky. Temperley–Lieb Algebra: From Knot Theory to Logic and Computation via Quantum Mechanics. In Goong Chen, Louis Kauffman, and Samuel Lomonaco, editors, *Mathematics of Quantum Computation and Quantum Technology*, volume 20074453, pages 515–558. Chapman and Hall/CRC, September 2007. ISBN 978-1-58488-899-4 978-1-58488-900-7. doi: 10.1201/9781584889007.ch15.

Rajeev Alur and Pavol Černý. Expressiveness of streaming string transducers. In Kamal Lodaya and Meena Mahajan, editors, IARCS Annual Conference on Foundations of Software Technology

- and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India, volume 8 of LIPIcs, pages 1–12. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2010. doi: 10.4230/LIPIcs.FSTTCS.2010.1.
- Dimitry S. Ananichev and Mikhail V. Volkov. Synchronizing monotonic automata. *Theoretical Computer Science*, 327(3):225–239, 2004. doi: 10.1016/j.tcs.2004.03.068.
- Karl Auinger. Pseudovarieties generated by Brauer type monoids. *Forum Mathematicum*, 26(1):1–24, 2014. doi: doi:10.1515/form.2011.146.
- Nicolas Baudru, Louis-Marie Dando, Nathan Lhote, Benjamin Monmege, Pierre-Alain Reynier, and Jean-Marc Talbot. Weighted automata and expressions over pre-rational monoids. In Florin Manea and Alex Simpson, editors, 30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference), volume 216 of LIPIcs, pages 6:1–6:16. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.CSL.2022.6.
- Jean-Camille Birget. Concatenation of inputs in a two-way automaton. *Theoretical Computer Science*, 63 (2):141–156, 1989. doi: 10.1016/0304-3975(89)90075-3.
- Mikołaj Bojańczyk and Thomas Colcombet. Tree-walking automata do not recognize all regular languages. *SIAM Journal on Computing*, 38(2):658–701, 2008. doi: 10.1137/050645427.
- Mikołaj Bojańczyk, Laure Daviaud, and Shankara Narayanan Krishna. Regular and First-Order List Functions. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science LICS '18*, pages 125–134, Oxford, United Kingdom, 2018. ACM Press. ISBN 978-1-4503-5583-4. doi: 10.1145/3209108.3209163.
- Guillaume Bonfante and Florian L. Deloup. Decidability of regular language genus computation. *Mathematical Structures in Computer Science*, 29(9):1428–1443, 2019. doi: 10.1017/S0960129519000057.
- Ronald V. Book and Ashok K. Chandra. Inherently nonplanar automata. *Acta Informatica*, 6:89–94, 1976. doi: 10.1007/BF00263745.
- Olivier Carton and Luc Dartois. Aperiodic Two-way Transducers and FO-Transductions. In Stephan Kreutzer, editor, 24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany, volume 41 of LIPIcs, pages 160–174. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2015. doi: 10.4230/LIPIcs.CSL.2015.160.
- Pierre Clairambault and Andrzej S. Murawski. On the Expressivity of Linear Recursion Schemes. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. ISBN 978-3-95977-117-7. doi: 10.4230/LIPIcs.MFCS.2019.50.
- Luc Dartois, Paulin Fournier, Ismaël Jecker, and Nathan Lhote. On reversible transducers. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 113:1–113:12. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2017. doi: 10.4230/LIPIcs.ICALP.2017.113.

- Luc Dartois, Ismaël Jecker, and Pierre-Alain Reynier. Aperiodic String Transducers. *International Journal of Foundations of Computer Science*, 29(05):801–824, August 2018. ISSN 0129-0541, 1793-6373. doi: 10.1142/S0129054118420054.
- James East. Presentations for Temperley–Lieb Algebras. *The Quarterly Journal of Mathematics*, 72(4): 1253–1269, 02 2021. ISSN 0033-5606. doi: 10.1093/qmath/haab001.
- Calvin C. Elgot and Jorge E. Mezei. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68, 1965. doi: 10.1147/rd.91.0047.
- Henning Fernau, Carolina Haase, and Stefan Hoffmann. The synchronization game on subclasses of automata. In Pierre Fraigniaud and Yushi Uno, editors, 11th International Conference on Fun with Algorithms, FUN 2022, May 30 to June 3, 2022, Island of Favignana, Sicily, Italy, volume 226 of LIPIcs, pages 14:1–14:17. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.FUN.2022.14.
- Emmanuel Filiot, Shankara Narayanan Krishna, and Ashutosh Trivedi. First-order definable string transformations. In Venkatesh Raman and S. P. Suresh, editors, 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India, volume 29 of LIPIcs, pages 147–159. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2014. ISBN 978-3-939897-77-4. doi: 10.4230/LIPIcs.FSTTCS.2014.147.
- Jean Goubault-Larrecq. Musings around the geometry of interaction, and coherence. *Theoretical Computer Science*, 412(20):1998–2014, 2011. doi: 10.1016/j.tcs.2010.12.023.
- Peter M. Higgins. Divisors of semigroups of order-preserving mappings on a finite chain. *International Journal of Algebra and Computation*, 5(6):725, 1995. doi: 10.1142/S0218196795000306.
- Peter Hines. A categorical framework for finite state machines. *Mathematical Structures in Computer Science*, 13(3):451–480, 2003. doi: 10.1017/S0960129503003931.
- Peter Hines. Temperley-Lieb Algebras as two-way automata. http://www.dcs.gla.ac.uk/~simon/qnet/talks/Hines.pdf, 2006. Slides of a talk given at the QNET Workshop 2006.
- Shin-ya Katsumata. Attribute grammars and categorical semantics. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations, volume 5126 of Lecture Notes in Computer Science, pages 271–282. Springer, 2008. doi: 10.1007/978-3-540-70583-3 23.
- Louis Kauffman. An invariant of regular isotopy. *Transactions of the American Mathematical Society*, 318(2):417–471, April 1990. doi: 10.2307/2001315.
- Kenneth Krohn and John Rhodes. Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. *Transactions of the American Mathematical Society*, 116:450–464, 1965. ISSN 0002-9947, 1088-6850. doi: 10.1090/S0002-9947-1965-0188316-1.

- Kwok Wai Lau and D. G. FitzGerald. Ideal structure of the kauffman and related monoids. *Communications in Algebra*, 34(7):2617–2629, 2006. doi: 10.1080/00927870600651414.
- Olga Martynova and Alexander Okhotin. State complexity of transforming graph-walking automata to halting, returning and reversible. *Information and Computation*, 291:105011, 2023. doi: 10.1016/j.ic. 2023.105011.
- Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins series in the mathematical sciences. Johns Hopkins University Press, 2001. ISBN 978-0-8018-6689-0. doi: 10.56021/9780801866890.
- Anca Muscholl and Gabriele Puppis. The Many Facets of String Transducers. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. ISBN 978-3-95977-100-9. doi: 10. 4230/LIPIcs.STACS.2019.2.
- Lê Thành Dũng Nguyễn. *Implicit automata in linear logic and categorical transducer theory*. PhD thesis, Université Paris XIII (Sorbonne Paris Nord), December 2021. URL https://theses.hal.science/tel-04132636.
- Lê Thành Dũng Nguyễn and Cécilia Pradic. Implicit automata in typed λ-calculi I: aperiodicity in a non-commutative logic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, 47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference), volume 168 of LIPIcs, pages 135:1–135:20. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.ICALP.2020.135.
- Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959. doi: 10.1147/rd.32.0114.
- John C. Shepherdson. The reduction of two-way automata to one-way automata. *IBM Journal of Research and Development*, 3(2):198–200, 1959. doi: 10.1147/rd.32.0198.
- Howard Straubing. First-order logic and aperiodic languages: a revisionist history. *ACM SIGLOG News*, 5(3):4–20, 2018. doi: 10.1145/3242953.3242956.