Radix enumeration of rational languages is almost co-sequential.

Pierre-Yves Angrand, Jacques Sakarovitch

RADIX ENUMERATION OF RATIONAL LANGUAGES IS ALMOST CO-SEQUENTIAL

Pierre-Yves Angrand[1] and Jacques Sakarovitch[2]

Abstract. We define and study here the class of rational functions that are finite union of sequential functions. These functions can be realized by cascades of sequential transducers. After showing that cascades of any height are equivalent to cascades of height at most two and that this class strictly contains sequential functions and is strictly contained in the class of rational functions, we prove the result whose statement gives the paper its title.

AMS Subject Classification. — Give AMS classification codes —.

Introduction

We define and study here the class of rational functions that are finite union of sequential functions (of pairwise disjoint domains).

This class appeared rather naturally in the study of the concrete complexity of the successor function in some non standard numeration systems (cf. [2]). Without going into details of that work, one of the problem which is tackled there is the definition of a computation model that is powerful enough to describe successor functions and that allows the definition of complexity with a sufficient degree of abstraction.

For that purpose we consider *cascades of sequential transducers*, that is, functions that compute the value of a word w by the following procedure : w is read by a first sequential transducer τ which outputs a word u; then u is read by another sequential transducer σ_s that depends on the state q reached by τ at the end of the reading of w, and so on, for a fixed number of steps h.

Keywords and phrases: finite automata, rational functions of words, sequential transducers

[1] LTCI (UMR 5141), CNRS / ENST, 46 rue Barrault, 75634 Paris Cedex 13, France
e-mail: angrand@enst.fr

[2] LTCI (UMR 5141), CNRS / ENST, 46 rue Barrault, 75634 Paris Cedex 13, France
e-mail: sakarovitch@enst.fr

In this paper we first prove that on one hand these functions are all of the kind we just said : finite union of sequential functions with pairwise disjoint domains, independently of the parameter h and then that this class is strictly larger than the class of sequential function.

We then prove the main result of this paper, namely :

Theorem 1. *The radix enumeration of a rational language is a finite union of co-sequential functions.*

1. Preliminaries

In this section we will recall basic definitions and properties that will help us to define and manipulate the cascade of sequential transducers. For further details one may see [7].

1.1. Transducers basic definitions

In the sequel, A and B are two finite alphabets. A^* and B^* are the free monoids respectively generated by A and B. Let u be a word, $|u|$ is the length of u. Let $<$ be an order on the letters of the alphabet A. We define the radix order $\prec$ on words by $u \prec v$ if either $|u| < |v|$ or $|u| = |v|$ and there exist w, u', v' words and $a < b$ letters such that $u = wau'$ and $v = wbv'$. Let us also denote $\mathrm{Rat}\,(A^*)$ the set of rational languages over the alphabet A.

Definition 1.1 (finite real-time transducers)**.** Let us call *finite real-time transducer* from A^* into B^* an automaton $\tau = \langle Q, A, B^*, E, I, T\rangle$ where :

- Q is a finite set of states,
- E is a set of transitions $(p, (a, X), q)$, where p and q are states of Q, a is a letter of A and X is in $\mathrm{Rat}\,(B^*)$,
- I is a set of elements (p, X), where p is a state, which is said initial, and $X \in \mathrm{Rat}\,(B^*)$,
- T is a set of elements (p, X), where p is a state, which is said final, and $X \in \mathrm{Rat}\,(B^*)$.

In all that follows we call transducer any finite real-time transducer. The transducer τ describes a function φ that maps $u \in A^*$ with $\{v \mid v \in B^*\}$ such that there exist $v_1v_2v_3 = v$, $(p, X_1) \in I$ and $(q, X_3) \in T$ with $v_1 \in X_1$ and $v_3 \in X_3$ and there exists a path from p to q in τ labeled by (u, v_2). φ is a *rational relation.*

A rational relation φ is said to be functional if for every word $u \in A^*$ there exists at most one $v \in B^*$ such that $u\varphi = v$. $\mathrm{RatF}(A^* \times B^*)$ is the set of functions from A^* to B^*. If $(p, (u, v), q)$ is a transition of a transducer then we call u the *input* of this transition and v the *output.* A rational function is realized by a finite real-time transducer which outputs of initial states, final states and transitions are singletons on B^*.

Remark 1.2. The initial set I and the final set T can also be seen as a partial functions of $Q \to B^*$ such that for $p \in Q : pI = u \Leftrightarrow (p, u) \in I$ and $p \in Q : pT = u \Leftrightarrow (p, u) \in T$.

We also define the underlaying *input automaton* of a transducer $\tau = \langle Q, A, B^*, E, I, T\rangle$ as the finite automaton $input(\tau) = \langle Q, A, E', I', T'\rangle$ where :

- $(p, a, q) \in E'$ if there exists X such that $(p, (a, X), q) \in E$,
- $p \in I'$ if there exists X such that $(p, X) \in I$
- $q \in T'$ if there exists X such that $(p, X) \in T$

We call *domain* of the transducer τ the language recognized by the input automaton of τ.

1.2. Sequential Functions

In the sequel, in order to simplify definitions and notation, all word functions or relations map words of A^* into A^*. Since we are never interested in the functions or relations being total or surjective, this is not a restrictive hypothesis.

Definition 1.3 (sequential and co-sequential)**.** A transducer is said to be a sequential (resp. *co-sequential*) transducer if the underlaying input automaton is deterministic (resp. *co-deterministic*) and if the outputs are singletons.

Remark 1.4. A co-sequential transducer is also often seen as a transducer with deterministic input automaton but which reads and write words from the right to the left, which is then a *right sequential transducer*.
It will be no surprise that we follow the terminology of [7] and we reserve the qualifiers ‘ right ’ and ‘ left ’ *for automata* (and hence for transducers) which model physical machines, and according to as they read words *from right to left* or *from left to right* respectively. Functions are neither left or right but they can be realized by transducers which can be right or left.

A sequential or co-sequential transducer realizes a rational function. A rational function $\varphi : A^* \to A^*$ is sequential (resp. co-sequential) if it is realized by some sequential (resp. co-sequential) transducer. We denote by Seq the and coSeq the sets of sequential and co-sequential functions. It is known that these sets are closed under composition.

Sequential functions are characterized within rational functions by a topological criterion in the following way : the *prefix distance* d of two words u and v is defined as $\mathsf{d}(u, v) = |u| + |v| - |u \wedge v|$, where $|u|$ is the length of $|u|$ and $u \wedge v$ is the longest common prefix of u and v.

Definition 1.5 (Lipschitz function)**.** Let φ be a rational function. φ is said k-Lipschitz for the prefix distance if :

$$\forall f, g \in \mathsf{Dom}(\varphi) \ , \quad \mathsf{d}(f\varphi, g\varphi) \leq k \ \mathsf{d}(f, g)$$

If there exists k such that φ is k-Lipschitz then φ is Lipschitz.

Theorem 1.6. *Let φ be a rational function then φ is sequential if and only if φ is Lipschitz.*

Example 1.7. Let $U_\tau = \{1, 3, 8, 21, ...\}$ be the numeration system defined by the reccurence $u_{n+2} = 3u_{n+1} - u_n$. It is known that the integers are all represented by the rational language $L_1 = A^* \setminus \{A^*(21^*2)A^* \cup 0A^*\}$ where $A = \{0, 1, 2\}$ (cf. [4]). Let us consider the successor function such that, for $u \in L$, $\mathsf{Succ}_L(u) = v$ where v is the successor of u in L for the radix order (if $u \prec w$ then either $v = w$ or $v \prec w$).

Let $u = 2(1^k)$ and $v = 2(1^k)0$. It holds :

$$\mathsf{Succ}_{L_1}(u) = 1(0^{k+1}) \qquad \mathsf{Succ}_{L_1}(v) = 2(1^k)$$

It holds then $\mathsf{d}(u, v) = 1$ and $\mathsf{d}(\mathsf{Succ}_{L_1}(u), \mathsf{Succ}_{L_1}(v)) = 2k + 3$. For any k the function Succ_{L_1} is not k-Lipschitz and hence is not Lipschitz and not sequential.

Remark 1.8. By a left-right duality we define the suffix distance. A rational function is co-sequential if and only if it is Lipschitz for this new distance.

Theorem 1.9.

$$\mathrm{Seq} \subsetneq \mathrm{RatF} \quad , \quad \mathrm{coSeq} \subsetneq \mathrm{RatF} \quad .$$

2. Cascade of Sequential Transducers

In this section we introduce a new machine using transducers in order to compute a class of rational functions strictly including the sequential functions (and especially in order to include the Succ_L function of the previous example).

2.1. Definition and basic results

We abusively denote by the same letter τ the function and the transducer that realizes it, the context making clear which object we are referring to. If the set of states of τ is Q, we associate with τ (either way the transducer and the function) a function from A^* to Q denoted $\underset{\sim}{\tau}$ such that $f\underset{\sim}{\tau}$ is the state reached at the end of the computation reading f by τ.

We define a cascade of sequential transducers, of height 2, in the following way.

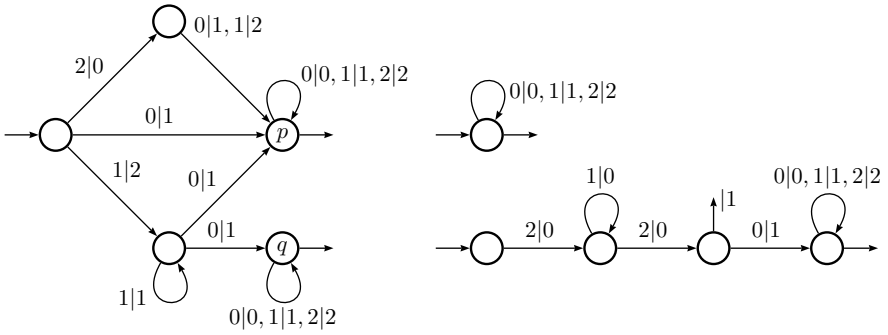
Definition 2.1 (Cascade of sequential transducers of height 2)**.** Let τ be a sequential transducer with set of states Q and let $\{\sigma_q\}_{q\in Q}$ be a family of $card(Q)$ sequential functions or transducers.

The cascade $\theta = (\tau, \{\sigma_q\})$ is the function θ from A^* into itself defined by :

$$\forall f \in A^* \qquad f\theta = (f\tau)\sigma_j \quad \text{if} \quad j = f\underset{\sim}{\tau}$$

that is $f\theta$ is the result of composition of two sequential functions τ and σ_q where the second one, σ_q, depends upon the state q reached by τ at the end of its computation on f.

Example 2.2 (*Ex. 1.7 cont.*)**.** Succ_{L_1} is recognized by a cascade of height 2 reading from right to left. The cascade $(\tau, \{\sigma_p, \sigma_q\})$ where τ and σ_q are described in Figure 1 and σ_p is the identity (sequential) transducer.



(a) τ (b) σ_p and σ_q

FIGURE 1

From this example it holds that cascades of sequential transducers allow to describe more functions than sequential transducers.

In view of generalisation to cascades of larger height, let us say that Q is the set of *active states* of τ and if R_q is the set of states of (the transducer) σ_q then $R = \bigcup_{q \in Q} R_q$ is the set of active states of θ. With θ is associated the function $\underset{\sim}{\theta} : A^* \to R$ by :

$$\forall f \in A^* \qquad f\underset{\sim}{\theta} = (f\tau)\underset{\sim}{\sigma_q} \quad \text{if} \quad f\underset{\sim}{\tau} = q$$

Definition 2.3 (Cascade of sequential transducers)**.** A cascade of height 1 is a sequential function (or transducer) τ.

Let θ be a cascade of height h with set of active states Q and for every q in Q let σ_q be a sequential transducer with set of states R_q. The function ω denoted $\omega = (\theta, \{\sigma_q\})$ and defined by :

$$\forall f \in A^* \qquad f\omega = (f\theta)\sigma_q \quad \text{if} \quad q = f\underset{\sim}{\theta}$$

is a cascade of height $h + 1$, whose set of active states is $R = \bigcup_{q \in Q} R_q$ and whose associated state map $\underset{\sim}{\omega}$ is defined by :

$$\forall f \in A^* \qquad f\underset{\sim}{\omega} = (f\theta)\underset{\sim}{\sigma_q} \quad \text{if} \quad q = f\underset{\sim}{\theta}$$

Let us recall the common following result :

We shall not elaborate more on cascade of arbitrary height because of the following theorem.

Theorem 2.4. *Let $\tau : A^* \to A^*$ be a function. The following are equivalent :*

(1) *τ is realized by a cascade of height 1 or 2,*
(2) *τ is realized by a cascade of height h, $h \geq 1$,*
(3) *τ is a finite union of sequential functions whose domain pairwise disjoint.*

Proof. (2) $\Rightarrow$ (3) : Let τ be a cascade of height h, with set of active states Q and for each $q \in Q$, let $K_q = (q)[\underset{\sim}{\tau}]^{-1}$ the set of words that are mapped by the cascade τ on the state q.

Let $\{\sigma_q\}_{q\in Q}$ be a family of sequential functions with active states R_q and $\omega = (\tau, \{\sigma_q\})$ the cascade of height $h+1$ it allows to define.

Let $r \in R = \bigcup_{q\in Q} R_q$ then there exists a unique q such that $r \in R_q$. We shall prove recursively the more precise property :

(a) : The K_q are pairwise disjoint rational sets.
(b) : The restriction of τ to K_q is a sequential function denoted τ_q .

The property obviously holds for $h = 1$.

$$\forall f \in K_q \qquad f\omega = (f\tau)\sigma_q = (f)[\tau_q\sigma_q]$$

that is the restriction of ω on K_q is a sequential function and $L_r = r[\underset{\sim}{\omega}]^{-1} = \left((r)[\underset{\sim}{\sigma}_q]^{-1}\right)\tau_q^{-1}$ is a rational set (contained in K_q). Since $\underset{\sim}{\omega}$ is a function the L_r are pairwise disjoints. From (a) and (b), now proven, we have a finite union of sequential transducers ω_r of pairwise disjoint language for ω cascade of height $h+1$.

(3) $\Rightarrow$ (1) : Let φ be recognized by a finite union of sequential transducers of domain pairwise disjoint $\{\sigma_1, \ldots, \sigma_k\}$. The domains of these transducers form a partition of A^* (we can consider the empty transducer for every word that is in no domain).

Lemma 2.5. *Let $\bigcup_i L_i$ be a finite rational partition of A^* (L_i are rational languages and pairwise disjoint). There exists a deterministic automaton such that there exists a partition of the final states for which the languages recognized by these final states are the languages L_i.* □

From Lemma 2.5 we can build an automaton $\mathcal{A}$ whose final states recognize the different rational languages. Let τ be the identity transducer with input automaton $\mathcal{A}$. τ is a sequential transducer whose final states recognize different rational languages of the partition.

The cascade, of height 2, $(\tau, \{\sigma'_{q_1}, \ldots, \sigma'_{q_l}\}$ where $\sigma'_{q_i} = \sigma_j$ if language recognized by final state q_i in $\mathcal{A}$ is included in domain of σ_j, obviously realizes φ.

□

By a left-right duality we define cascade of co-sequential transducers and it follows :

Corollary 2.6. *Let $\tau : A^* \to A^*$ be a function. The following are equivalent :*

- *τ is realized by a cascade of right sequential transducers of height 1 or 2,*
- *τ is realized by a cascade of right sequential transducers of height h, $h \geq 1$,*
- *φ is a finite union of co-sequential functions whose domain pairwise*

Corollary 2.7. *Cascades of sequential (resp. co-sequential) functions are rational functions.*

2.2. Cascades and rational functions

Let us now denote CSeq and CcoSeq the sets of cascades of respectively sequential and co-sequential transducers. From the previous subsection we know CSeq $\subset$ RatF, CcoSeq $\subset$ RatF and Seq $\subsetneq$ CSeq and coSeq $\subsetneq$ CSeq. We now prove that cascades form a strict subfamily of rational functions.

Let $A = \{a, b\}$ and let $\varphi : A^* \to A^*$ be the *Fibonacci reduction*, that is the function that maps every word w to the unique word $w\varphi$ obtained from w by the rewriting rule $abb \to baa$ and that contains no factor abb anymore.

It is well known that φ is a rational function (see for instance [1, Exer. III 5.5] , [5]) that is neither sequential or co-sequential (cf. [3]).

Let for instance $u_k = (ab)^k a$ and $v_k = (ab)^k b$. $d(u_k, v_k) = 1$ and $u_k\varphi = u_k$ and $v_k\varphi = (aa)^k b$ give $\mathsf{d}(u_k\varphi, v_k\varphi) = 2k$ and hence φ is not Lipschitz.

The following property holds :

Lemma 2.8. *Let u and v be two non empty words of A^*. It holds :*

$$\varphi(uaav) = \varphi(u)a\varphi(av)$$

Proposition 2.9. *The Fibonacci reduction is not realized by a cascade of either sequential or co-sequential transducers.*

Proof. Consider now the family of $n+1$ words X_i, $1 \leq i \leq n+1$ defined in the following way :

$$X_i = w_{i,n}\ aa\ w_{i,n-1}\ aa\ \ldots\ aa\ w_{i,1}$$

where $w_{i,k} = v_{l_i}$ if $k = i$ and $w_{i,k} = u_{l_i}$ if $k \neq i$ and where the $u_{l_i} = (ab)^{l_i} a$ and $v_{l_i} = (ab)^{l_i} b$ as above and where the sequence of the n integers l_i will be define inductively below. From Lemma 2.8 follows that :

$$X_i\varphi = u_{l_n}\ aa\ u_{l_{n-1}}\ aa\ \ldots\ aa\ u_{l_{I+1}}\ aa\ b(aa)^{l_i}\ aa\ u_{l_{i-1}}\ aa\ \ldots\ aa\ u_{l_1}$$

and :

$$X_{n+1}\varphi = X_{n+1} = u_{l_n}\ aa\ u_{l_{n-1}}\ aa\ \ldots\ aa\ u_{l_1}$$

from which we deduce that for all i, j, $1 \leq i < j \leq n+1$:

$$\mathsf{d}(X_i, X_j) = 2\left(\sum_{1\leq m\leq i-1}(2l_m+3)\right) \quad ,$$
$$\mathsf{d}(X_i\varphi, X_j\varphi) = 2\left(\sum_{1\leq m\leq i-1}(2l_m+3)+2l_i+1\right) \quad .$$

If we chose $l_1, \ldots, l_n$ to recursively verify :

$$l_1 > \frac{1}{2}(K-1)$$
$$l_i > \frac{1}{2}(K-1)\left(\sum_{1\leq m\leq i-1}(2l_m+3)\right)$$

then for all i, j, $1 \leq i < j \leq n+1$:

$$\mathsf{d}(X_i\varphi, X_j\varphi) > K\mathsf{d}(X_i\varphi, X_j\varphi)$$

and φ cannot be the union of only n K-Lipschitz functions and then cannot be union of sequential functions with pairwise disjoint domains.

By choosing $u_k = a(bb)^k$ and $v_k = ab(bb)^k$ we prove in the same way that φ cannot be the union of n co-sequential functions.

□

An similar proof gives the following proposition :

Proposition 2.10. *The successor function on the whole set A^* itself is a co-sequential function that is not finite union of sequential functions.*

Remark 2.11. Let $A = \{a, b, c\}$ be an alphabet. Let us consider the following functions :

$$\mathsf{Dom}(\chi_1) = \{(aa)^k | k \in \mathbb{N}\} \qquad \chi_1((aa)^k) = (aa)^k \ ,$$
$$\mathsf{Dom}(\chi_2) = \{a(aa)^k | k \in \mathbb{N}\} \qquad \chi_2(a(aa)^k) = b(bb)^k \ ,$$
$$\mathsf{Dom}(\chi_3) = \{ca(aa)^k | k \in \mathbb{N}\} \qquad \chi_3(ca(aa)^k) = (bb)^{k+1} \ .$$

$$\psi_1 = \chi_1 \cup \chi_2 \qquad \psi_2 = \chi_1 \cup \chi_3$$

The rational functions χ_1 , χ_2 and χ_3 are obviously sequential and co-sequential and have pairwise disjoint domains. ψ_1 and ψ_2 are then both in CcoSeq and in CSeq. ψ_1 is neither sequential or co-sequential and ψ_2 is sequential but not co-sequential. Symmetricaly to ψ_2 a function ψ_3 can be build, that is in both CcoSeq and CSeq and that is co-sequential but not sequential. The geography of Figure 2 for rational functions holds then.

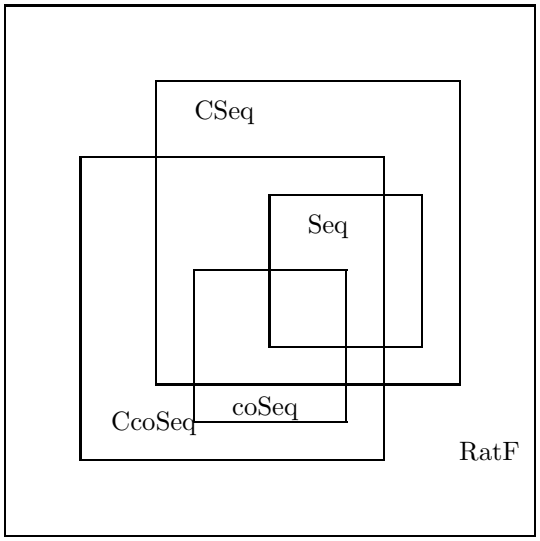


FIGURE 2. The geography of rational functions.

3. Radix enumeration of rational languages

Radix enumeration of a language L is realized by the iteration of the function $\mathsf{Succ}_L : A^* \to A^*$ that maps every word of L onto its successor in L ordered by radix order. Standard properties of (synchronized) rational relations allow to prove the following result (cf. [2], [7]) :

Proposition 3.1. *The successor function of a rational language is realized by a letter-to-letter finite transducer.*

The example 1.6 already shown that the successor function is not necessary a sequential function, even if L is rational but the continuation of this example have shown that it can be a co-sequential transducer. We now establish ou main result Theorem 1. For that purpose lut us fixe some notations and recall some further definitions.

Definition 3.2. Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ and $\mathcal{B} = \langle R, A, F, J, U\rangle$ be two finite automata. The *synchronized product* of $\mathcal{A}$ and $\mathcal{B}$ is the transducer :

$$\mathcal{A} \bowtie \mathcal{B} = \langle Q\times R, A, A, G, I\times J, T\times U\rangle$$

where the set of transitions G is defined by :

$$G = \{((p,r),(a,b),(q,s)) \mid (p,a,q) \in E, (r,b,s) \in F\}$$

Let $\mathcal{A}$ and $\mathcal{B}$ be two finite automata that recognize respectively the rational languages L and K. The synchronized product[1] $\mathcal{A} \bowtie \mathcal{B}$ realizes the relation $\theta : A^* \to A^*$ such that $\theta = \{(u, v) \mid u \in L, v \in K, |u| = |v|\}$. The transducer $\mathcal{A} \bowtie \mathcal{B}$ realizes a function if and only if K has at most one word for every length but even if $\mathcal{A}$ and $\mathcal{B}$ are deterministic, it is not likely to be sequential as soon as there is a state in $\mathcal{B}$ which is the source of more than one transition.

Example 3.3. The Figure 3 shows the synchronized product $\mathcal{A}_1 \bowtie \mathcal{B}_1$ where $\mathcal{A}_1$ and $\mathcal{B}_1$ are both deterministic and $\mathcal{A}_1$ recognizes $L_1 = \{u \in A^* \mid \exists k,\ u = (ab)^k\}$ and $\mathcal{B}_1$ recognizes $K_1 = \{v \in B^* \mid \exists k,\ u = ba(ba)^k aa\}$.
It can be seen that the state (p, q) is the source of two transitions with same input (and hence the synchronized product is not sequential) due to the fact that state q of $\mathcal{B}_1$ is source of two transitions.

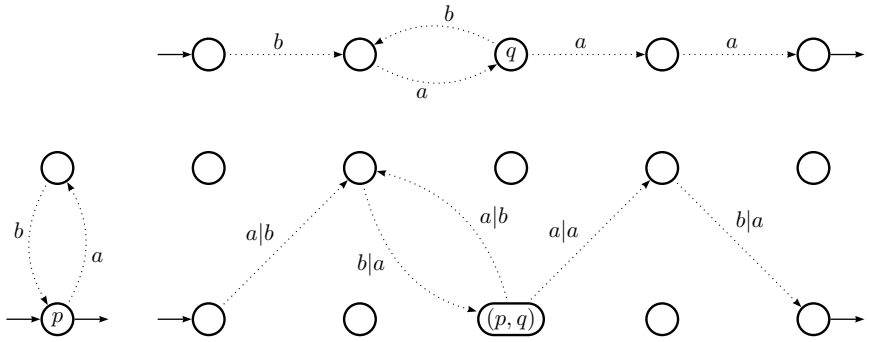


FIGURE 3. Synchronized product of two deterministic automata $\mathcal{A}_1$ and $\mathcal{B}_1$

The synchronized product is additive and $\left[\bigcup_{x\in X} \mathcal{A}_x\right] \bowtie \left[\bigcup_{y\in Y} \mathcal{B}_y\right] = \bigcup_{x,y}(\mathcal{A}_x \bowtie \mathcal{B}_y)$

For the purpose of the construction underlying our proof, we slightly enlarge the family of automata we are considering :

- the transitions of automata are labeled either by a letter a of A or by the empty word 1_A^* (transitions labeled by 1_A^* are *spontaneous transitions*),
- the subsets I and T of Q for initial and final states are replaced by (partial) functions from Q into A^*, still denoted by I and T, and their value $I(p)$ and $T(q)$ are rather denoted I_p and T_q.

The *label* $|c|$ of a computation $c : i \xrightarrow{f}_{\mathcal{A}} t$ is then $|c| = I_i.f.T_t$. We denote by $\ell(c)$ the number of transitions of the computation c.

[1]this synchronized product is slightly different from the one considered in [7, Exerc. IV.6.17] where it recognizes the relation $L \times K$.

When it is needed we call *classical automaton* an automaton without spontaneous transitions and with final and initial states. It is well known that the family of languages recognized by such automata is not larger than $RatA^*$ and that any classical automaton can be seen as automata choosing $T_t = 1_A^*$ when t is final, $T_t = \emptyset$ otherwise, $I_i = 1_A^*$ if i is initial and $I_i = \emptyset$ otherwise.

The definition of this larger class of automata is taken in view of the following construction.
A language L is a *ray language* if it is of the form $L = uv^*w$ and an automaton is a *ray automaton* if its structure shows that it recognizes a ray language : it is trim, it has a unique initial state, a unique final state and consists in a unique circuit together with a unique path coming into the circuit and a unique path going out of the circuit (*e.g.* $\mathcal{A}_1$ and $\mathcal{B}_1$ of Figure 3 are ray automata).

Definition 3.4. Every ray automaton $\mathcal{B}$ is transformed into an equivalent ray automaton $\breve{\mathcal{B}}$ with the following properties :

(i) Every state of $\breve{\mathcal{B}}$ is the source of at most one transition.
(ii) If the computations c in $\mathcal{B}$ and c' in $\breve{\mathcal{B}}$ have same label then $\ell(c) = \ell(c')$.

Property (i) is obtained by replacing the unique path going out of the circuit by final function on the state which is source of this path; Property (ii) by adding adequate number of spontaneous transitions at the beginning of the path coming in the circuit (as it can be seen on Figure 4 for $\mathcal{B}_1$).

The definition of synchronized product goes over this larger class of automata and it now holds:

$$\mathcal{A} \bowtie \mathcal{B} = \langle Q \times R, A \cup \{1_{A^*}\}, A \cup \{1_{A^*}\}, G, I \times J, T \times U \rangle$$

where :

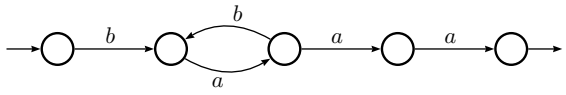
- $I \times J_{(p,q)} = (I_p, J_q)$
- $T \times U_{(p,q)} = (T_p, U_q)$
- $G = \{((p,r),(x,y),(q,s)) \mid (p,x,q) \in E, (r,y,s) \in F\}$

If $\mathcal{A}$ is a classical automaton, the synchronized product $\mathcal{A} \bowtie \mathcal{B}$ realizes relation which associate with u in L all words in K that are label of computations in $\mathcal{B}$ whose length is equal to $|u|$.

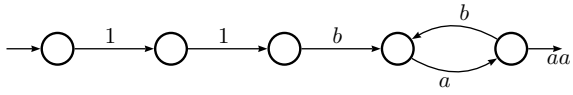
Example 3.5. The Figure 5 shows the example of synchronized product for $\mathcal{A}_1$ and $\breve{\mathcal{B}_1}$.

From Definition 3.4 it follows the two following properties :

Property 3.6. *Let $\mathcal{A}$ and $\mathcal{B}$ be two ray automata then $\mathcal{A} \bowtie \mathcal{B}$ and $\mathcal{A} \bowtie \breve{\mathcal{B}}$ are equivalent.*



(a) $\mathcal{B}_1$



(b) $\check{\mathcal{B}}_1$

FIGURE 4. $\mathcal{B}_1$ and $\check{\mathcal{B}}_1$

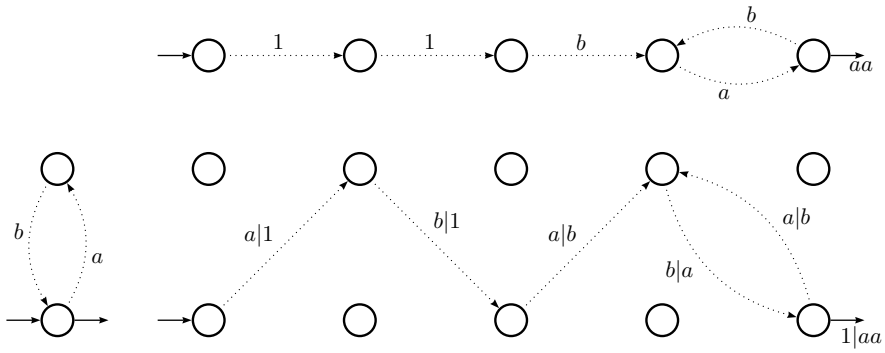


FIGURE 5. Synchronized product of $\mathcal{A}_1$ and $\check{\mathcal{B}}_1$

Property 3.7. *Let $\mathcal{A}$ and $\mathcal{B}$ be two deterministic ray automata. $\mathcal{A} \bowtie \check{\mathcal{B}}$ is a sequential transducer with a single initial and a single final state.*

Proposition 3.8. *Let L and K be two rational languages with at most one word for each length. Let $\varphi : L \to K$ such that $u\varphi = \{v \in K \mid |u| = |v|\}$. The function φ is realized by a finite union of sequential transducers with a single final state ($\varphi \in$ CcoSeq).*

Proof. A rational language with at most one word for each length is a union of disjoint ray languages and then can be realized by a union of deterministic ray

automata. Let $\mathcal{A}_1, \ldots, \mathcal{A}_l$ be the deterministic ray automata whose union recognizes L and $\mathcal{B}_1, \ldots, \mathcal{B}_k$ be the deterministic ray automata whose union recognizes K. K is also recognized by the union of $\breve{\mathcal{B}_1}, \ldots \breve{\mathcal{B}_k}$.

For all i and j, $\mathcal{A}_i \bowtie \breve{Bc}_j$ is a sequential transducer and from Property **??** realizes the same function that $\mathcal{A}_i \bowtie \breve{\mathcal{B}_j}$. The function φ is realized by the union of $\mathcal{A}_i \bowtie \mathcal{B}_j$, thus by the union of $\mathcal{A}_i \bowtie \breve{\mathcal{B}_j}$, which proves the statement. □

The previous propositions and their proofs are obviously symmetrical and the following result holds :

Proposition 3.9. *Let L and K be two rational languages with at most one word for each length. Let $\varphi : L \to K$ such that $u\varphi = \{v \in K \mid |u| = |v|\}$. The function φ is realized by a finite union of co-sequential transducers with a single initial state ($\varphi \in$ CcoSeq).* □

Let L be a language of A^*. Let us denote the sets of *minimal words* and of *maximal words* of L by :

$$\mathsf{Min}(L) = \{u \in L \mid \forall v \in L, |u| = |v| \Rightarrow u \prec v\}$$
$$\mathsf{Max}(L) = \{u \in L \mid \forall v \in L, |u| = |v| \Rightarrow v \prec u\}$$

It is also well known that if L is rational, so are $\mathsf{Min}(L)$ and $\mathsf{Max}(L)$ (cf. [8] [6] for instance).

The main building block of our construction is synchronized product, which yields transducers that realizes length-preserving functions. As Succ_L is not a length-preserving function (it holds $u \in \mathsf{Max}(L) \Leftrightarrow |u| < |\mathsf{Succ}_L(u)|$). We slightly change both the function and the language that we study.
If L is a language of A^*, let ULSucc_L be the *uniform length successor function*, that is the restriction of Succ_L to $(A \times A)^*$:

$$\forall u \in L,\ \mathsf{ULSucc}_L(u) = \mathsf{Succ}_L(u) \Leftrightarrow |u| = |\mathsf{Succ}_L(u)| \Leftrightarrow u \notin \mathsf{Max}(L)$$

And if u is in $\mathsf{Max}(L)$ then $\mathsf{ULSucc}_L(u)$ is undefined.
For an ordered alphabet A, let $A_\$ = A \cup \{\$\}$ where \$ is a letter that does not belongs to A and by assumption $\$ < a$ for every a in A. We associate with every language L of A^* the language $K = \$^*L$ of $A_\$^*$ and it holds :

$$\mathsf{Succ}_L(u) = v \Leftrightarrow \exists! k,\ \mathsf{ULSucc}_K(\$^k u) = v \tag{1}$$

Proposition 3.10. *If ULSucc_K is realized by finite union of co-sequential transducers, then so is Succ_L.*

Proof. Let $\pi : A_\$^* \to A^*$ be the projection that erases the \$ symbol. With a slight abuse of notation we write :

$$\pi\left[\mathsf{ULSucc}_K \cap (\$^* A^* \times A^*)\right] = \mathsf{Succ}_L$$

Let $\tau_1, \ldots, \tau_k$ be the co-sequential transducers whose union realizes ULSucc_K and for every τ_i let :

$$\tau_i' = \tau_i \cap \$^* A^* \times A^*$$

The transducer τ_i' is co-sequential as ($\$^* A^* \times A^*$) is a recognizable relation (cf. [7]). By (1) it holds that for any word $u \in A^*$, if $\$^k u$ is in $\mathsf{Dom}(\tau_i)$ then for no $l \neq k$, $\$^l u \in \mathsf{Dom}(\tau_i)$. For every state q of τ_i' from which can be read a word $u \in A^*$ to a final state, there exists at most one (finite) path with a word of $\$^*$ as input. Since $\mathsf{Dom}(\tau_i') \subset \$^* A^*$ this path begins with an initial state. Let us note v' the output on this path.
For every τ_i, let τ_i'' be the transducer built from τ_i' by replacing the finite paths labeled by $\$|v'$ by initial function with output v' on the state they arrive. τ_i'' is equivalent to $\pi(\tau_i')$ and is also co-sequential.
Since $\mathsf{Dom}(\tau_i') \subset \mathsf{Dom}(\tau_i)$ and since for any $u \in A^*$ there is at most a unique k such that $\$^k u \in \mathsf{Dom}(\tau')$, it holds that the domains of the τ_i'' are pairwise disjoints. □

An automaton is said to be *standard* (resp. *co-standard*) if it has a single initial (resp. final) state without any incoming (resp. outgoing) transition. Same definitions go for transducers.
For every rational relations φ and φ', the *concatenation relation* of these relations is denoted[2] by $\varphi||\varphi'$ and such that if $(u, v) \in \varphi$ and $(u', v') \in \varphi'$ then $(uu', vv') \in \varphi||\varphi'$.
A construction define the *concatenation* on the transducers τ and τ' by adding a spontaneous transition from final states of τ to initial states of τ' and the output of this transition is the concatenation of the final and initial function. The concatenation on transducers realizes the concatenation of the relation realized by the transducers.

Let τ be a co-sequential and co-standard transducer and let τ' be a co-sequential transducer with a single initial state then $\tau||\tau''$ realizes a co-sequential function.

Proposition 3.11. *If L is a rational language of A^* then ULSucc_L is a union of co-sequential functions.*

Proof. Let $\mathcal{A} = \langle Q, A, \delta, i, T\rangle$ be a deterministic automaton that recognizes L and that is fixed for the remaining of the proof. We also note $q = p.w$ if $q = \delta(p, w)$, $L_p = \{w \in A^* \mid p.w \in T\}$ and $L_p' = \{w \in A^* \mid i.w = p\}$. The L_p and L_p' are rational and the L_p' are pairwise disjoint as $\mathcal{A}$ is deterministic.

Analysis : Let u in L and $v = \mathsf{ULSucc}_L(u)$. Let $w = u \wedge v$ the longest prefix common to u and v : $u = wau'$ and $v = wbv'$ with $a < b$ (this holds since $|u| = |v|$). Both au' and bv' belong to L_p and w to L_p'.

[2]The concatenation on words is usually denoted as multiplication but in order to avoid confusion with the composition on functions that is also usually denoted that way we prefer to have a different notation.

Let $K_{p,a}$ be the set of maximal words of L_p that begin with an a :

$$K_{p,a} = \mathsf{Max}(a(a^{-1}L_p))$$

and let $H_{p,a}$ be the set of minimal words of L_p that begin with a letter c greater than a :

$$H_{p,a} = \mathsf{Min}(\bigcup_{c>a} c(c^{-1}L_p))$$

The sets $K_{p,a}$ and $H_{p,a}$ are both rational sets.

Claim 1 : If $w \in L'_p$ then $au' \in K_{p,a}$ and $bv' \in H_{p,a}$.

Proof of Claim 1 : If $w \in L'_p$ and wau' is recognized by $\mathcal{A}$ then $au' \in a(a^{-1}L_p)$. Let us suppose that $au' \notin K_{p,a}$ then there exists $au'' \in a(a^{-1}L_p)$ such that $au'' > au'$ and then $Succ(wau') \in wa(a^{-1}L_p)$. That is non sense with v being successor of u so $au' \in K_{p,a}$. We prove $bv' \in H_{p,a}$ the same way. □

$K_{p,a}$ and $H_{p,a}$ have both at most one word for every length. By Proposition 3.8 and Proposition 3.9 the synchronized product of automata that recognizes them is a finite union of sequential or co-sequential transducers $\tau_1, ..., \tau_k$ with pairwise disjoint domains. Let us choose the co-sequential option. There exists exactly one i such that au' belongs to $\mathsf{Dom}(\tau_i)$ and then $bv' = (au')\tau_i$.
Let $\mathcal{K}'_p$ be a co-deterministic and co-standard automaton recognizing L'_p and κ_p be the co-sequential and co-standard transducer that realizing the identity on L'_p. The concatenation $\theta_{p,i} = \kappa_p||\tau_i$ is a co-sequential transducer and it holds $v = u\theta_{p,i}$.

Synthesis : Let $\theta_{p,i} = \kappa_p||\tau_i$ be a co-sequential transducer built as above and let r and s be two words of A^* such that :

$$r = s\theta_{p,i}$$

First, s belongs to $\mathsf{Dom}(\theta_{p,i}) = \mathsf{Dom}(\kappa_p).\mathsf{Dom}(\tau_i)$ and necessarily $s = ww'$ with $w \in L'_p$ and $w' \in K_{p,a} \subset L_p$. s thus belongs to L. It follows then that $r = ww''$ with w'' in $H_{p,a}$ and $|w''| = |w'|$.

Claim 2 : If $w' \in K_{p,a}$, $w'' \in H_{p,a}$, $|w'| = |w''|$ then for all w in L'_p :

$$ww'' = \mathsf{ULSucc}_L(ww')$$

Proof of Claim 2 : If $w' \in K_{p,a}$, $w'' \in H_{p,a}$, $|w'| = |w''|$ then from definition of $K_{p,a}$ and $H_{p,a}$ it holds that ww' and ww'' are in L, $|ww'| = |ww''|$ and $ww' < ww''$ and hence the longest common prefix of ww' and its successor has w as prefix. Since w' is in $K_{p,a}$ (the set of maximal words of L_p beginning with a) the longest common prefix of ww' and its successor is prefix of w. Hence the longest common prefix of ww' and its successor is w. From Claim 1 and since $K_{p,a}$ and $H_{p,a}$ have

at most one word for every length $ww'' = \mathsf{ULSucc}_L(ww')$. □

As ULSucc_L is a function, the domain of the $\theta_{p,i}$ are pairwise disjoint which completes the proof of Proposition 3.11 and thus of Theorem 1.

□

Remark 3.12. Every construction involved in the proof of Proposition 3.11 is symmetrical in the sense that the automata can be chosen to be deterministic or co-deterministic and the transducers sequential or co-sequential (for instance the 'τ_i') but for one point : the automata $\mathcal{K}'_p$ can be chosen to be co-deterministic and co-standard but it is not possible to assume, in general, that it can be chosen deterministic and co-standard (which would be necessary to complete the symmetrical construction). And indeed (cf. Proposition 2.10) the successor function for A^* is not CSeq.

References

[1] Berstel, J. *Transductions and Context-Free Languages*. Teubner, 1979.

[2] Berthé, V., Frougny, C., Rigo, M., and Sakarovitch, J. On the cost and complexity of the successor function. In *WORDS2007* (2007), pp. 43–56.

[3] Frougny, C. Fibonacci representations and finite automata. *IEEE Trans. on Infor. Theory. 37* (1991), 393–399.

[4] Frougny, C. On the sequentiality of the successor function. *Inform. and Computation 139* (1997), 17–38.

[5] Perrin, D. Finite automata, *in: Handbook of Theoretical Computer Science*, (1990) Vol B, pp. 1–53.

[6] Sakarovitch, J. Deux remarques sur un théorème de S. Eilenberg. *RAIRO Theor. Informatics and Appl. 17* (1983), 23–48.

[7] Sakarovitch, J. *Eléments de théorie des automates*. Vuibert, 2003. English translation: *Elements of Automata Theory*, Cambridge University Press, to appear.

[8] Shallit, J. Numeration systems;linear recurrences, and regular sets. *Inform. and Comput. 113* (1994), 331–347.

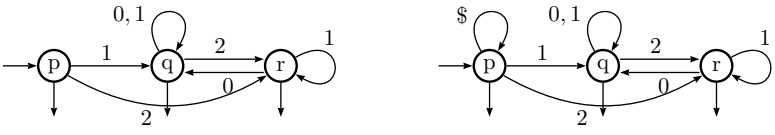
Appendix A. Example

For L a rational language, the proof of the theorem gives us an algorithm to build a union of co-sequential transducers realizing Succ_L :

(1) Choose $\mathcal{A}$ a deterministic automaton recognizing L.
(2) Add a loop labeled by \$ on the initial state of $\mathcal{A}$ to have $\mathcal{B}$ recognizing language is $K = \$^*L$.
(3) For every state p of $\mathcal{B}$ and every letter a, build automata for $K_{p,a}$ and $H_{p,a}$ and split them into union of ray automata.
(4) Co-determinize automata of $K_{p,a}$ and transform any automaton $\mathcal{H}$ of $H_{p,a}$ into $\breve{\mathcal{H}}$.
(5) $H_{p,a} \bowtie K_{p,a}$ is the union $\tau_{p,a,1}, \ldots, \tau_{p,a,k}$ of the synchronized products.
(6) Compute $\mathcal{K}'_p$ (and κ_p) for all state p of $\mathcal{B}$ and concatenate κ_p with $\tau_{p,a,i}$ for each letter a.
(7) Delete the transition with \$ as input and make state p initial with u as output whenever p could be reached from initial state by a path labeled $\$^k|u$.

We use this algorithm to build a union of co-sequential transducers to realize the successor function in the set of representation of integers in the base of the square of Fibonacci's numbers.

Let $A = \{0, 1, 2\}$ and let L_1 be the language denoted by $A^* - A^*(21^*2)A^* - 0A^*$. Let $K_1 = \$^*L_1$, the Figure 6 shows automata to recognize L and K (steps **(1)** and **(2)**).



(a) Deterministic automaton for L_1

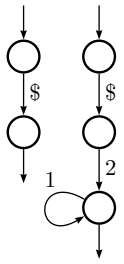
(b) Deterministic automaton for K_1

Figure 6. Automata for L_1 and K_1.

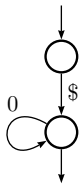
The Figure 7 gives the result for step **(3)**.

It can be seen that all components of $K_{s,a}$ for s state and a a letter of $\{0, 1, 2, \$\}$ are already co-deterministic. In this particular example (and in order to simplify the final result), even if some components of $H_{s,a}$ have input degrees greater than 1, we skip step **(4)**. This can be done in this special case because the synchronized products are nonetheless co-sequential (as it can be seen on Figure 8).

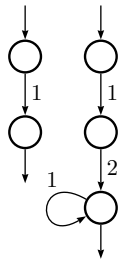
Figure 9 shows the κ_s for every states s. The Figure 10 gives the final result after steps **(6)** and **(7)**. It can be seen that transducers of Figure 10 are not



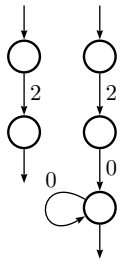
(a) $K_{p,\$}$



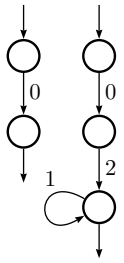
(b) $H_{p,\$}$



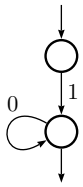
(c) $K_{p,1}$



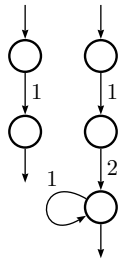
(d) $H_{p,1}$



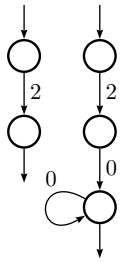
(e) $K_{q,0}$



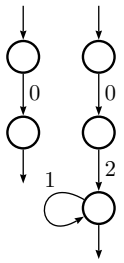
(f) $H_{q,0}$



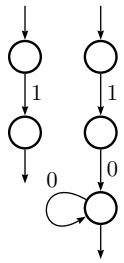
(g) $K_{q,1}$



(h) $H_{q,1}$



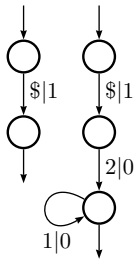
(i) $K_{r,0}$



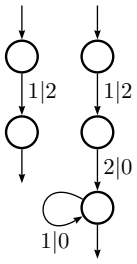
(j) $H_{r,0}$

FIGURE 7

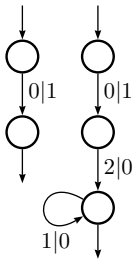
really co-sequential since there are several final states. We indeed merged the



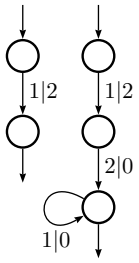
(a) $K_{p,\$} \bowtie H_{p,\$}$



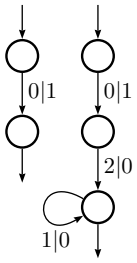
(b) $K_{p,1} \bowtie H_{p,1}$



(c) $K_{q,0} \bowtie H_{q,0}$



(d) $K_{q,1} \bowtie H_{q,1}$

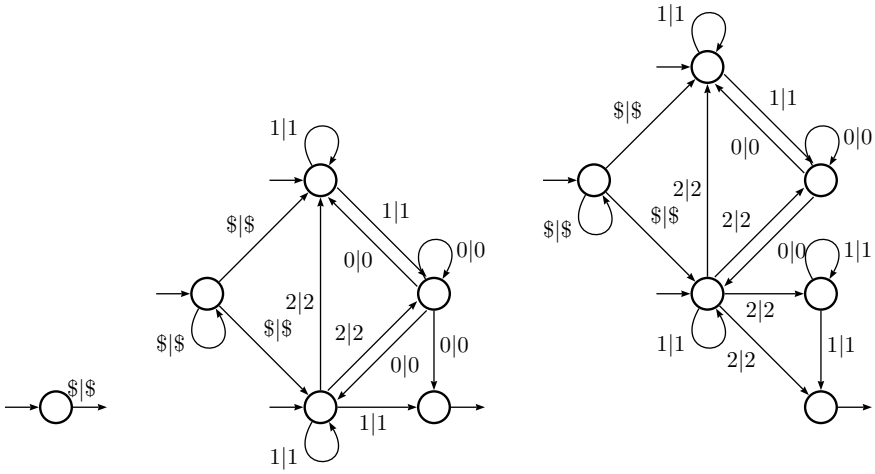


(e) $K_{r,0} \bowtie H_{r,0}$

FIGURE 8

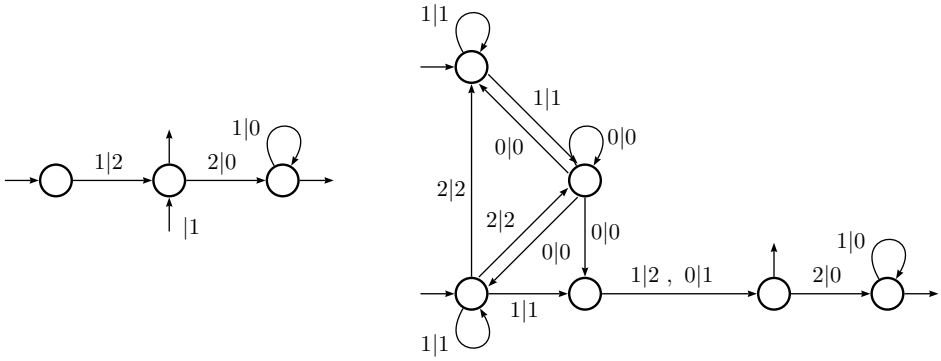
concatenations of κ_s and $K_{q,a} \bowtie H_{q,a}$, for every letters a, into the same transducer to simplify the result.

.

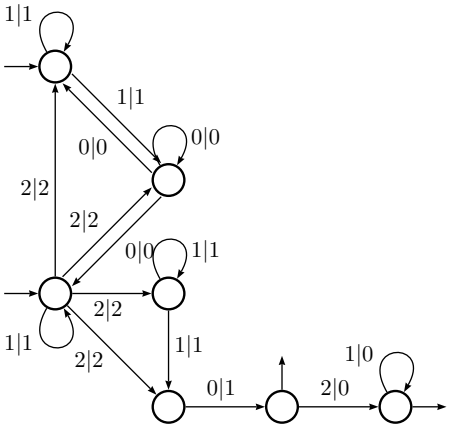


(a) $\mathcal{K}'_p$ (b) $\mathcal{K}'_q$ (c) $\mathcal{K}'_r$

FIGURE 9



(a) (b)



(c)

FIGURE 10