RAIRO. THEORETICAL INFORMATICS AND APPLICATIONS

D. SIMPLOT
A. TERLUTTE

Closure under union and composition of iterated rational transductions

RAIRO. Theoretical Informatics and Applications, tome 34, n° 3 (2000), p. 183-212

<http://www.numdam.org/item?id=ITA_2000__34_3_183_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **34** (2000) 183–212

CLOSURE UNDER UNION AND COMPOSITION OF ITERATED RATIONAL TRANSDUCTIONS

D. SIMPLOT[1] AND A. TERLUTTE[1]

Abstract. We proceed our work on iterated transductions by studying the closure under union and composition of some classes of iterated functions. We analyze this closure for the classes of length-preserving rational functions, length-preserving subsequential functions and length-preserving sequential functions with terminal states. All the classes we obtain are equal. We also study the connection with deterministic context-sensitive languages.

Résumé. Nous poursuivons notre étude des transductions itérées. Dans cet article, nous étudions la cloture par union et composition de quelques classes de fonctions itérées. Nous considérons les fonctions rationnelles, les fonctions sous-séquentielles et les fonctions séquentielles avec états terminaux, et plus particulièrement, celles préservant les longueurs. Toutes les classes de transductions obtenues sont égales et sont en relation étroite avec les langages contextuels déterministes.

AMS Subject Classification. 68Q45, 68Q42, 68Q70.

1. INTRODUCTION

The class of rational transductions, introduced by Elgot and Mezei [4] and mainly studied by Nivat, Eilenberg and Schüzenberger [3,7,8], certainly constitutes the best known class of transformations of languages. This class has good closure properties (composition, union, inversion,...) and the main families of languages are closed under rational transductions.

In language theory, some tools are defined in terms of iterations of processes (Chomsky grammars, Lindenmayer systems,...). We can inquire about iterations

Keywords and phrases: Rational transductions, rational functions, iteration of transductions, context-sensitive languages.

[1] URA 369 du CNRS, LIFL, Université de Lille I, bâtiment M3, Cité Scientifique, 59655 Villeneuve-d'Ascq Cedex, France; e-mail: {simplot,terlutte}@lifl.fr

of rational transductions. Iterating any kind of rational transductions is very powerful. We restrict our study to length-preserving (l.p.) rational transductions.

In [9], we have studied the smallest class of transductions containing l.p. rational transductions and closed under union, composition and iteration (named UCI-closure). We have found some equivalent representations of this family. For instance, the following classes are equal: UCI-closure of l.p. rational transductions ($UCI(\mathcal{T})$), UCI-closure of l.p. rational functions ($UCI(\mathcal{F})$), compositions using one iterated l.p. rational transduction ($\mathcal{T}\mathcal{T}_+\mathcal{T}$) and compositions using iteration of two l.p. rational functions ($\mathcal{F}(\mathcal{F}+\mathcal{F})_+\mathcal{F}$).

In this paper, we study the class $UC(\mathcal{F}_+)$ where $\mathcal{F}_+$ denotes the class of iterated length-preserving rational functions. We prove that $\mathcal{F}\mathcal{F}_+\mathcal{F}$ is a representation for this class (Sect. 4). Then we study $UC(\mathcal{C}_+)$ for some subclasses $\mathcal{C}$ of rational functions: subsequential functions (Sect. 5) and sequential functions with terminal states (Sect. 6). We can prove that these three classes are equal: $UC(\mathcal{F}_+) = UC(_s\mathcal{S}_+) = UC(_{ts}\mathcal{S}_+)$.

In [9], we have seen the connection between context-sensitive transductions and the class $UCI(\mathcal{T})$. This is due to the generation of context-sensitive languages by iterations of l.p. rational transductions. We shall prove the same kind of properties for deterministic context-sensitive languages: the class of deterministic context-sensitive languages is equal to $a^*UC(\mathcal{F}_+)$ where a is a letter (Sect. 7) and the class $UC(\mathcal{F}_+)$ is equal to the class of deterministic context-sensitive transductions (Sect. 8).

A part of this work has been already published as extended abstract in [5].

2. Preliminaries

We assume the reader to be familiar with basic formal language theory (see [2,3] for more precisions). The goal of this section is to fix notations and terminology.

2.1. Words and languages

For a finite alphabet Σ, we denote by Σ^* the free monoid generated by Σ. The neutral element of this monoid is the empty word, which is denoted by ε. The size of the alphabet Σ is denoted by $||\Sigma||$ and is equal to its number of letters. The length of a word u is denoted by $|u|$. For a set S of words, alph(S) denotes the alphabet of S, that is the set of letters which occur in S.

A language over Σ is a subset of Σ^*. The classes of regular, deterministic context-sensitive and context-sensitive languages over Σ are denoted respectively by $\mathcal{R}ec(\Sigma^*)$, $\mathcal{CS}_{\det}(\Sigma^*)$, $\mathcal{CS}(\Sigma^*)$.

Regular languages are recognized by finite states automata and context-sensitive languages are recognized by linear-bounded automata (definition of LBA is given in Sect. 7).

Let Σ be an alphabet which is the Cartesian product of n alphabets, $\Sigma = X_1 \times X_2 \times \ldots X_n$ (with $n \geq 1$). A word u belonging to $\Sigma^* = (X_1 \times X_2 \times ... \times X_n)^*$ will be denoted by $u = [u_1, u_2, ..., u_n]$ where u_i is the i-th component and belongs to X_i^*.

2.2. Transductions

Now we give some basic definitions about transductions. A transduction is a subset of $X^* \times Y^*$ where X and Y are two finite alphabets. For a word u, the set of images of u by a transduction τ is denoted by $u\tau$ and is defined by:

$$u\tau = \{v \mid (u, v) \in \tau\} \cdot$$

This definition is extended in a canonical way to languages: $L\tau = \{v \mid \exists u \in L$ such that $(u, v) \in \tau\}$.

For a transduction τ, the *domain* of τ, denoted by $\mathrm{Dom}(\tau)$, is the set of all words which have an image by τ. The *set of images* of τ, denoted by $\mathrm{Im}(\tau)$, is the language of words which have an antecedent by τ.

$$\begin{aligned} \mathrm{Dom}(\tau) &= \{u \mid \exists v \text{ such that } (u, v) \in \tau\}, \\ \mathrm{Im}(\tau) &= \{v \mid \exists u \text{ such that } (u, v) \in \tau\}. \end{aligned}$$

A transduction τ is called complete on X^* if $\mathrm{Dom}(\tau) = X^*$.

The inverse of a transduction τ is the transduction whose couples are the permutation of first and second components of the couples belonging to τ. It is denoted by $\tau^{-1} = \{(v, u) \mid (u, v) \in \tau\}$.

The set of transductions has a structure of semigroup according to the composition operation:

Definition 2.1. Let τ and σ be two transductions. The composition of τ and σ is the transduction defined by:

$$\tau\sigma = \{(u, w) \mid \exists v \text{ such that } (u, v) \in \tau \wedge (v, w) \in \sigma\} \cdot$$

A transduction τ from X^* into Y^* is rational if and only if it is a rational part of $X^* \times Y^*$ - that is a part built of finite sets of $X^* \times Y^*$ and using union, usual concatenation and star operator. It is also the class of transductions which can be realized by a finite transducer – that is a finite automaton where edges are labeled by an input and an output word.

Formally, a finite transducer T is a 6-uple (X, Y, Q, δ, I, F) where X is a finite alphabet called the input alphabet, Y is a finite alphabet called the output alphabet, Q is a finite set of states, δ is the transition function from $Q \times \{X \cup \varepsilon\}$ into the finite parts of $Q \times Y^*$, I is the set of initial states included in Q and F is the set of final states included in Q.

The transition function is extended to δ_* defined on $Q \times X^*$ in its entirety. First δ_* contains δ. For each state q, we force $(q, \varepsilon)\delta_*$ to contain (q, ε). If $(q, x)\delta_* \ni (q', y)$

and $(q', x')\delta \ni (q'', y')$ then $(q, xx')\delta_*$ contains (q'', yy'). For a given word $u \in X^*$, we say that u is transformed in $v \in Y^*$ if there exist an initial state $q_i \in I$ and a final state $q_f \in F$ such that $(q_f, v) \in (q_i, u)\delta_*$. The transduction τ associated with T is defined by:

$$\tau = \bigcup_{q_i \in I,\ q_f \in F} \{(u, v) \in X^* \times Y^* \mid (q_f, v) \in (q_i, u)\delta_*\} \cdot$$

The next theorem presents well-known closure properties of the class of rational transductions (see [2] for detailed proofs of these properties).

Theorem 2.2. *The class of rational transductions is closed under union, composition and inverse.*

A transduction τ is length-preserving if and only if for each couple $(u, v) \in \tau$ we have $|u| = |v|$. In the remainder of the paper, we consider only length-preserving transductions. The class of all length-preserving rational transductions is denoted by $\mathcal{T}$.

A length-preserving transduction is letter-to-letter if and only if the transition function δ is defined from $Q \times X$ into the finite parts of $Q \times Y$. A class of letter-to-letter transductions is denoted by $\mathcal{C}_{ll}$.

Let $\mathcal{C}$ and $\mathcal{C}'$ be two classes of transductions. We denote by $\mathcal{C}\mathcal{C}'$ the class of transductions obtained by composition of a transduction of $\mathcal{C}$ with a transduction of $\mathcal{C}'$:

$$\mathcal{C}\mathcal{C}' = \{\tau\tau' \mid \tau \in \mathcal{C} \wedge \tau' \in \mathcal{C}'\} \cdot$$

Let $\mathcal{C}$ be a class of transductions. The class of all transductions whose inverses belong to $\mathcal{C}$ is denoted by $\mathcal{C}^{-1}$:

$$\mathcal{C}^{-1} = \{f^{-1} \mid f \in \mathcal{C}\} \cdot$$

2.3. Functions

A transduction τ is functional if for each word u in $\mathrm{Dom}(\tau)$, $u\tau$ contains exactly one word. When we deal with a function τ we will write $u\tau = v$ instead of $u\tau = \{v\}$.

The class of length-preserving rational functions is denoted by $\mathcal{F}$.

The intersection with a language could be considered as a transduction which is the identity on this language. Let L be a language over X^*.

$$(\cap L) = \{(u, u) \mid u \in L\} \cdot$$

This transduction is obviously a length-preserving function. It is a rational function if and only if the language L is rational.

Let $\mathcal{L}$ be a class of languages. The class of functions which consist to an intersection with a language of $\mathcal{L}$, denoted by $(\cap\mathcal{L})$, is defined by:

$$(\cap\mathcal{L}) = \{\tau \mid \tau = (\cap A) \text{ for some } A \in \mathcal{L}\}\cdot$$

There are no class of transducers associated to the class of rational functions. Some restrictions can be made in order to introduce determinism in the transitions. Subclasses of rational functions are thus defined.

In a left sequential transducer, the transition function δ is defined from $Q \times X$ into $Q \times Y^*$. The usual presentation of left sequential transducers replaces the transition function by a next state function μ and an output function λ. Formally, a left sequential transducer is a 6-uple $(X, Y, Q, q_0, \mu, \lambda)$ where X is the input alphabet, Y is the output alphabet, Q is the set of states, including a unique initial state q_0, μ is the next state function included in $(Q \times X) \times Q$ and λ is the output function included in $(Q \times X) \times Y^*$.

The next state function and the output function can be extended to $Q \times X^*$ by setting, for $q \in Q$, $u \in X^*$ and $x \in X$,

$$\begin{array}{ll} (q,\varepsilon)\mu_* = q, & (q,ux)\mu_* = ((q,u)\mu_*),x)\mu, \\ (q,\varepsilon)\lambda_* = \varepsilon, & (q,ux)\lambda_* = (q,u)\lambda_*((q,u)\mu_*),x)\lambda. \end{array}$$

For a given word $u \in X^*$, we say that u is transformed into $v \in Y^*$ if $(q_0,u)\mu_* \neq \emptyset$ and $(q_0,u)\lambda_* = v$.

Let us remark that all states are terminal.

A transduction is left sequential if there exists a left sequential transducer which realizes it. The class of length-preserving sequential functions is denoted by $\mathcal{S}$.

By the same way, right sequential transducers are defined. The input word is then read from right to left (for more details, see [2, 3]).

The *ts*-sequential transducers (sequential with terminal states) are defined by adding a set of terminal states to the definition of left sequential transducers.

A *ts*-sequential transducer is a 7-uple $(X, Y, Q, q_0, Q_F, \mu, \lambda)$ where Q_F is the set of final states. A word $u \in X^*$ is transformed into $v \in Y^*$ if $(q_0,u)\mu_* \in Q_F$ and $(q_0,u)\lambda_* = v$.

A transduction is *ts*-sequential if there exists a *ts*-sequential transducer which realizes it. The class of length-preserving *ts*-sequential functions is denoted by ${}_{ts}\mathcal{S}$.

Subsequential transducers are defined by adding a final state function ρ included in $Q \times Y^*$. We don't need Q_F anymore; if a state is not final, its output will be empty.

A subsequential transducer is a 7-uple $(X, Y, Q, q_0, \mu, \lambda, \rho)$ where ρ is the final state function. A word $u \in X^*$ is transformed into $vw \in Y^*$ if $v \in Y^*$ if $(q_0,u)\mu_* \neq \emptyset$, $(q_0,u)\lambda_* = v$ and $((q_0,u)\mu_*)\rho = w$.

A transduction is subsequential if there exists a subsequential transducer which realizes it. The class of length-preserving subsequential functions is denoted by ${}_{s}\mathcal{S}$.

2.4. Morphisms

A morphism φ is a rational function that satisfies the conditions $\varepsilon\varphi = \varepsilon$ and $(uv)\varphi = (u\varphi)(v\varphi)$ for every words u, v. Hence, the morphism φ is completely defined by the values $a\varphi$ of the letters $a \in \text{Alph}(\text{Dom}(\varphi))$. The morphism φ is called letter-to-letter, if for each $a \in \text{Alph}(\text{Dom}(\varphi))$ we have $|a\varphi| = 1$. The class of letter-to-letter morphisms is denoted by $\mathcal{H}_{sa}$.

In our proofs, we shall use several particular kinds of morphisms. For an arbitrary alphabet A, the identity over A^* is denoted by I_A — notice that $I_A = \{(u, u) \mid u \in A^*\}$ is equivalent to the intersection with A^* which is denoted by $(\cap A^*)$. When we consider an alphabet Σ which is the Cartesian product of n alphabets, $\Sigma = X_1 \times X_2 \times \ldots X_n$ (with $n \geq 1$), the morphism Π_i, with $1 \leq i \leq n$ is the projection onto the ith component.

In our proofs, we shall also use:

- morphisms ρ_n which transform any letter x in a n-fold $(x, x, ..., x)$;
- morphisms $\delta_{(i_1,i_2,...,i_n)}$ which transform any n-fold $(x_1, x_2, ..., x_n)$ in a n-fold $(x_{i_1}, x_{i_2}, ..., x_{i_n})$ with $1 \leq i_j \leq n$;
- morphisms κ_x which transform every letter a in a letter x.

We shall fit the domains of these morphisms for the contexts.

2.5. Iterations of rational transductions

First, we give a formal definition of iterated transductions.

Definition 2.3. Let τ be a transduction. The iteration of τ, denoted by τ^+, is the transduction defined by:

$$\tau^+ = \bigcup_{i>0} \tau^i,$$

where τ^i is defined inductively by $\tau^1 = \tau$, $\tau^{n+1} = \tau^n\tau$ for $n > 0$.

For a given class of transductions $\mathcal{C}$, we denote by $\mathcal{C}_+$ the class of iterations of transductions of $\mathcal{C}$:

$$\mathcal{C}_+ = \{\tau^+ \mid \tau \in \mathcal{C}\}.$$

Thus, $\mathcal{T}_+$ denotes the class of iterated length-preserving rational transductions and $\mathcal{F}_+$ denotes the class of iterated length-preserving rational functions.

2.6. UCI-closure of length-preserving rational transductions

Definition 2.4. For a given class of transductions $\mathcal{C}$, the class $UCI(\mathcal{C})$ is the smallest class of transductions which contains $\mathcal{C}$ and is closed under union, composition and iteration.

Definition 2.5. The class $UC(\mathcal{C})$ is the smallest class of transductions which contains $\mathcal{C}$ and is closed under union and composition.

It is well known that the class of (length-preserving) rational transductions is closed under union and composition. Thus $UC(\mathcal{T}) = \mathcal{T}$. It is obvious that the iteration of a length-preserving rational transduction is not necessarily a rational transduction. Thus $UC(\mathcal{T}_+) \neq \mathcal{T}$.

The UCI-closure of length-preserving rational transductions was studied in [9].

We obtain the following characterization:

Theorem 2.6 (Representation theorem). *Let τ be a transduction. The following properties are equivalent:*

1. *the transduction τ can be defined by using union, composition and iteration of length-preserving rational transductions* $(\tau \in UCI(\mathcal{T}))$,
2. *there exist three length-preserving rational transductions σ_1, σ_2 and σ_3 such that $\tau = \sigma_1\sigma_2^+\sigma_3$* $(\tau \in \mathcal{T}\mathcal{T}_+\mathcal{T})$,
3. *there exist two length-preserving rational transductions σ_1 and σ_3 and a one-step transduction σ_2 such that $\tau = \sigma_1\sigma_2^+\sigma_3$* $(\tau \in \mathcal{T}\mathcal{O}_+\mathcal{T})$,
4. *there exist two letter-to-letter morphisms φ and ψ and a context-sensitive language A such that $\tau = \varphi^{-1}(\cap A)\psi$* $(\tau \in \mathcal{T}_{\mathcal{CS}})$,
5. *there exists a recognizable picture language L such that $\tau \setminus \{(\varepsilon, \varepsilon)\} = \tau_L$* $(\tau \in \mathcal{T}_{\mathcal{R}ec(LF)})$,
6. *the transduction τ can be defined by using union, composition and iteration of letter-to-letter ts-sequential functions* $(\tau \in UCI({}_{ts}\mathcal{S}_{ll+}))$,
7. *there exist four letter-to-letter ts-sequential functions σ_1, σ_2, σ_3 and σ_4 such that $\tau = \sigma_1(\sigma_2 + \sigma_3)^+\sigma_4$* $(\tau \in {}_{ts}\mathcal{S}_{ll}({}_{ts}\mathcal{S}_{ll} + {}_{ts}\mathcal{S}_{ll})_{+ts}\mathcal{S}_{ll})$.

In this paper, we shall study $UC(\mathcal{C}_+)$ for different classes of length-preserving rational functions.

3. Some remarks

Remark 3.1. The classes of length-preserving transductions are included as follows $\mathcal{H}_{sa} \subsetneq \mathcal{S} \subsetneq {}_{ts}\mathcal{S}_{ll} \subsetneq {}_{ts}\mathcal{S} \subsetneq {}_{s}\mathcal{S} \subsetneq \mathcal{F} \subsetneq \mathcal{T}$.

Length-preserving sequential functions are letter-to-letter. Letter-to-letter subsequential functions are *ts*-sequential.

In this work, we shall try to obtain the results for the smallest classes of transductions.

Remark 3.2. The class $\mathcal{T}$ is included in the class $UC(\mathcal{T}_+)$ but not in $\mathcal{T}_+$.

Indeed, when the image of a transduction uses the same alphabet as the domain, we cannot forbid the iteration of the transduction. But, using disjoint alphabets, we can use composition of two iterations in order to simulate a transduction. Thus the class of length-preserving rational transductions is included in $\mathcal{T}_+\mathcal{T}_+$. For instance, let us take the transduction which transforms a^n into b^n and transforms b^n into a^n. A single application changes a^n into b^n and reversely. But a second

application allows a^n to be the image of a^n. We can define a first transduction which transforms a^n into $\overline{b}^n$, b^n into $\overline{a}^n$ and is undefined over $\{\overline{a}, \overline{b}\}$ and a second one which transforms $\overline{b}^n$ into b^n, $\overline{a}^n$ into a^n and is undefined over $\{a, b\}$. The iteration of the first transduction followed by the iteration of the second is equal to the initial transduction.

This remark also holds for the class of rational (subsequential, *ts*-sequential or sequential) functions.

The same remark makes also the equality $UCI(\mathcal{T}) = UC(\mathcal{T}_+)$ obvious.

Remark 3.3. If two length-preserving rational functions have disjoint domains, their union is still a rational function.

The union of two length-preserving subsequential (*ts*-sequential, sequential) functions whose domains alphabets are disjoint, is a subsequential (*ts*-sequential, sequential) function.

For instance, let f_1 (resp. f_2) be the function which transforms a^{2n} into b^{2n} (resp. a^{2n-1} into c^{2n-1}) for all $n \geq 1$. These two functions are *ts*-sequential. The union $f_1 + f_2$ is a rational function but not a subsequential one.

Definition 3.4. A rational function f is called complete on X^* if $\mathrm{Dom}(f) = X^*$.

Remark 3.5. Let f be a length-preserving rational function included in $X^* \times Y^*$ and containing $(\varepsilon, \varepsilon)$.

Let $Z = X \cup Y \cup \{\diamond\}$, where $\diamond$ is a new special letter.

There exists a length-preserving rational function g, included in $Z^* \times Z^*$, complete on Z^* such that $f = g(\cap Y^*)$ and $f^+ = g^+(\cap Y^*)$.

We just have to define g on Z^* by $ug = uf$ when $u \in \mathrm{Dom}(f)$ and $ug = \diamond^{|u|}$ when $u \notin \mathrm{Dom}(f)$.

Let us note that, for any $u \in Z^*$ and all $n \in \mathbb{N}$, $u \in \mathrm{Dom}(g^n)$.

Let us note also that, if $u \in Z^* \diamond Z^*$ then $ug = \diamond^{|u|}$ and $ug^+ = \diamond^{|u|}$.

In order to have the same property ($f = g(\cap Y^*)$ and $f^+ = g^+(\cap Y^*)$), we could associate a complete length-preserving subsequential (resp. sequential) function g to a length-preserving subsequential (resp. sequential) function f, but not a complete length-preserving *ts*-sequential function to a length-preserving *ts*-sequential function. When dealing with length-preserving transductions, complete *ts*-sequential functions are sequential functions and length-preserving sequential functions are letter-to-letter.

For instance, let f be the function which transforms a^{2n} into b^{2n} for all $n \geq 0$. This function is *ts*-sequential. We could define a complete subsequential g which transforms a^{2n} into b^{2n} and a^{2n+1} into $b^{2n}\diamond$ for all $n \geq 0$. Thus we have the property $f = g(\cap b^*)$. But we cannot define a complete length-preserving *ts*-sequential function having the same property.

Let Σ be an alphabet which is the Cartesian product of n alphabets. We shall define a function which will be equivalent to n functions working on each component of n-folds. Let f_i, for $1 \geq i \geq n$, be functions included in $X_i^* \times Y_i^*$. Then $\langle f_1, f_2, ..., f_n \rangle$ will denote the function f with $\mathrm{Dom}(f) = \mathrm{Dom}(f_1) \times \mathrm{Dom}(f_2)$

$\times ... \times \mathrm{Dom}(f_n) \cap (X_1 \times X_2 \times ... \times X_n)^*$ defined by $uf = [u_1 f_1, u_2 f_2, ..., u_n f_n]$, for all $u = [u_1, u_2, ..., u_n] \in \mathrm{Dom}(f)$.

When the n functions are rational, the function thus defined is still rational.

Remark 3.6. Let τ be a length-preserving rational transduction included in $X^* \times Y^*$. Let $k = \|X\|$. For all $u \in X^*$, we have $uf^+ = \{uf^i \mid 0 < i \leq k^{|u|}\}$.

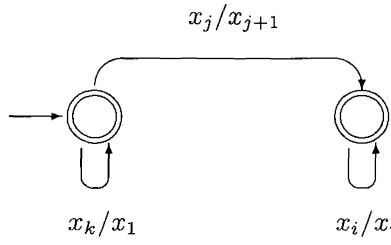
The set uf^+ is included into $X^{|u|}$ which is a finite set having $k^{|u|}$ elements.

We often need to count the number of iterations in order to limit them to $k^{|u|}$. We also need to enumerate all the words having a certain length. We can count from 1 to $k^{|u|}$ by enumerating all the words of length $|u|$ built from an alphabet of k letters.

Let X be the alphabet $\{x_1, x_2, ..., x_k\}$.

We define the function Succ_X which transforms a word into its successor of same length in the "mirror" lexicographical order. By "mirror" lexicographical order, we mean that the words are ordered reading from right to left. For instance, using the alphabet $\{x_1, x_2, x_3\}$, the words are ordered as follows $x_1x_1x_1 <_{lex} x_2x_1x_1 <_{lex} x_3x_1x_1 <_{lex} x_1x_2x_1 <_{lex} x_2x_2x_1 <_{lex} x_3x_2x_1 <_{lex} x_1x_3x_1 <_{lex} \cdots <_{lex} x_3x_3x_3$

The function Succ_X is defined by the following transducer.



for all $1 \leq j < k$

for all $1 \leq i \leq k$.

The function Succ_X thus defined is a length-preserving sequential function. It is complete on X^*; in particular $x_k^n \,\mathrm{Succ}_X = x_1^n$.

We can easily prove that a classical class of rational transductions which are not functions are obtained by iterations of functions.

Lemma 3.7 (Simplot and Terlutte [9]). *The class $\mathcal{H}_{sa}^{-1}$ of inverses of letter-to-letter morphisms is included in $UC(\mathcal{F}_+)$. Moreover, it is included in the class $\mathcal{S}\,\mathcal{S}_+\,\mathcal{S}$.*

Proof. Let h be a letter-to-letter morphism included in $X^* \times Y^*$. Let X_h be the alphabet defined by $X_h = \{(xh, x) \mid x \in X\}$. Let $k = \|X\|$.

Let us consider a word u for which we want the images through h^{-1}. We shall put u in the first component and enumerate all the words of $X^{|u|}$ into the second component in order to verify whether or not the couples belong to X_h^*.

Let $f_1 = (\cap Y^*)\rho_2\langle I_Y, \kappa_{x_k}\rangle$.

Let $f_2 = \langle I_Y, \mathrm{Succ}_X\rangle$.

Let $f_3 = (\cap X_h^*)\Pi_2$.

We verify that $h^{-1} = f_1 f_2^+ f_3$.

For all u belonging to Y^*, we have

$$\begin{aligned} uf_1f_2^+f_3 &= [u, x_k^{|u|}]f_2^+f_3 \\ &= \{[u,v] \mid v \in X^* \text{ and } |u| = |v|\}f_3 \\ &= \{[u,v] \mid v \in X^* \text{ and } u = vh\}\Pi_2 \\ &= uh^{-1}. \end{aligned}$$

□

After iterations of functions, we shall study iterations of subsequential and iterations of ts-sequential functions. The "decomposition theorem" [1, 3, 4, 8], characterizes rational functions by using sequential functions.

Theorem 3.8 (Decomposition Theorem). *Let φ be a function including $(\varepsilon, \varepsilon)$. The function φ is equal to $s_r s_l$ where s_r is a right sequential transduction and s_l is a left sequential transduction.*

Let us remark that the right sequential function can always be chosen length-preserving (then letter-to-letter). When the function φ is length-preserving, the left sequential function is then length-preserving for the words belonging to the image of the right sequential function, but not necessarily for the whole domain.

It is known [3,6] that a length-preserving transduction is the composition of an inverse of letter-to-letter morphism, an intersection with a rational language and a letter-to-letter morphism. Using this property in the proof of [1], it is immediate that the left sequential function s_l can also be chosen letter-to-letter.

Theorem 3.9 (Decomposition Theorem for length-preserving functions). *Let φ be a length-preserving function including $(\varepsilon, \varepsilon)$. The function φ is equal to $s_r s_l$ where s_r is a letter-to-letter right sequential function and s_l is a letter-to-letter left sequential function.*

In the decomposition theorem, the first sequential function read the word from right to left. We shall see that we can simulate a right sequential function with iterations of ts-sequential functions.

4. Use of one iteration of a rational function

We first study some properties of the class $\mathcal{F}\,\mathcal{F}_+\,\mathcal{F}$ in order to prove that this class coincides with the class $UC(\mathcal{F}_+)$.

Lemma 4.1. *Let τ be a length-preserving transduction belonging to $\mathcal{F}\,\mathcal{F}_+\,\mathcal{F}$, included in $A^* \times B^*$ and containing $(\varepsilon, \varepsilon)$.*

There exist an alphabet Z including A, a function g included in $Z^* \times Z^*$ and complete on Z^* and a morphism h included in $Z^* \times B^*$ such that $\tau = (\cap A^*)g^+h$.

Proof. Let $\tau = f_1f_2^+f_3$ with $f_i \subseteq A_i^* \times B_i^*$. We can suppose that $A \subseteq A_1$ and $B \subseteq B_3$. Let $Z_0 = (\cup_{i=1}^3 A_i) \cup (\cup_{i=1}^3 B_i)$.

The functions f_i are included in $Z_0^* \times Z_0^*$. Let $Z = Z_0 \cup \diamond$. Using Remark 3.5, we can define functions g_i included in $Z^* \times Z^*$, complete on Z^*, such that $f_i = g_i(\cap Z_0^*)$ and $f_i^+ = g_i^+(\cap Z_0^*)$.

Thus τ is equal to $f_1 f_2^+ f_3 = g_1(\cap Z_0^*) g_2^+(\cap Z_0^*) g_3(\cap Z_0^*)$.

In the Remark 3.5, the functions are built in such manner that $ug_i = uf_i \in Z_0^* \subset Z^*$ if $u \in \mathrm{Dom}(f_i)$, $ug_i = \diamond^{|u|}$ if $u \notin \mathrm{Dom}(f_i)$ and $\diamond^{|u|} g_i = \diamond^{|u|}$. Thus we have $\tau = g_1 g_2^+ g_3(\cap Z_0^*)$. The transduction $g_1 g_2^+ g_3$ is included in $Z^* \times Z^*$ and is complete on Z^*. Since $\mathrm{Dom}(\tau)$ is included in A^*, the transduction τ is also equal to $(\cap A^*) g_1 g_2^+ g_3(\cap Z_0^*)$.

We define

$\varphi_1 = (\cap A^*)\rho_2\langle g_1 g_2, g_1 g_2 g_3\rangle$

$\varphi_2 = (\cap(Z \times Z)^*)\delta_{(1,1)}\langle g_2, g_2 g_3\rangle$

$\psi_1 = \Pi_2(\cap Z_0^*)$.

We shall verify that $\tau = (\cap A^*) g_1 g_2^+ g_3(\cap Z_0^*)$ is equal to $(\cap A^*)(\varphi_1 + \varphi_2)^+ \psi_1$.

Let us first remark that the function φ_1 is included in $A^* \times (Z \times Z)^*$ and is complete on A^* and the function φ_2 is included in $(Z \times Z)^* \times (Z \times Z)^*$ and is complete on $(Z \times Z)^*$.

For all $u \in A^*$, for all $n \geq 1$, we have

$$\begin{aligned} u(\varphi_1 + \varphi_2)^n \psi_1 &= u\varphi_1 \varphi_2^{n-1} \psi_1 \\ &= [ug_1 g_2, ug_1 g_2 g_3](\delta_{(1,1)}\langle g_2, g_2 g_3\rangle)^{n-1} \Pi_2(\cap Z_0^*) \\ &= [ug_1 g_2^n, ug_1 g_2^n g_3]\Pi_2(\cap Z_0^*) \\ &= ug_1 g_2^n g_3(\cap Z_0^*). \end{aligned}$$

The functions φ_1 and φ_2 have disjoint domains (excepted on ε which is transformed in ε by both functions), thus the union $\varphi_1 + \varphi_2$ is a function.

The function $\varphi_1 + \varphi_2$ is included in $(A \cup (Z \times Z))^* \times (Z \times Z)^*$. Using Remark 3.5, we can define φ, included in $Z_1^* \times Z_1^*$, complete on $Z_1^* \supseteq (A \cup (Z \times Z))^*$, such that $(\varphi_1 + \varphi_2)^+ = \varphi^+(\cap(Z \times Z)^*)$.

Thus $\tau = (\cap A^*)\varphi^+\psi$ where $\psi = (\cap(Z \times Z)^*)\Pi_2(\cap Z_0^*) = (\cap(Z \times Z_0)^*)\Pi_2$ is a morphism on the alphabet $Z \times Z_0$. □

Lemma 4.2. *The class $\mathcal{F}\,\mathcal{F}_+\,\mathcal{F}$ is closed under union.*

Proof. Let τ_1 and τ_2 belonging to $\mathcal{F}\,\mathcal{F}_+\,\mathcal{F}$. We shall first suppose that $(\varepsilon, \varepsilon)$ belongs to τ_1 and τ_2. Let $A_1^* \times B_1^*$ and $A_2^* \times B_2^*$ be monoids in which the transductions τ_1 and τ_2 are respectively included. Then they are respectively included in $A^* \times B_1^*$ and $A^* \times B_2^*$ where $A = A_1 \cup A_2$.

By Lemma 4.1, there exist an alphabet $Z_1 \supseteq A$, a function g_1 included in $Z_1^* \times Z_1^*$ and complete on Z_1^* and a morphism h_1 included in $Z_1^* \times B_1^*$, such that $\tau_1 = (\cap A^*) g_1^+ h_1$. Idem for τ_2.

Let us show the construction on a diagram. The Figure 1 shows $(\cap A^*) g_1^+ h_1 + (\cap A^*) g_2^+ h_2$.

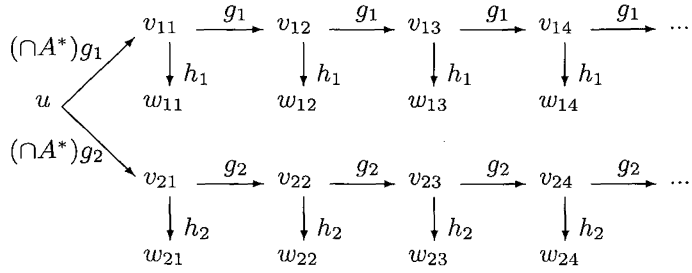


FIGURE 1. Generation of images through $(\cap A^* g_1^+ h_1) + (\cap A^* g_2^+ h_2)$.

In order to simulate $\tau_1 + \tau_2 = (\cap A^*)(g_1^+ h_1 + g_2^+ h_2)$, we shall use triplets. The transduction τ_1 will act on the first component and τ_2 on the second. The transductions τ_1 and τ_2 will act alternatively, according to the third component as shown in Figure 2.

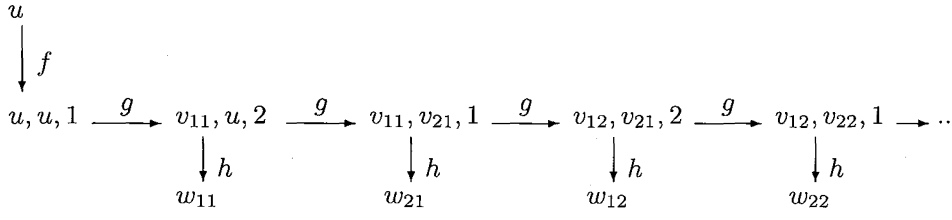


FIGURE 2. Generation of the same images through fg^+h.

Let $f = (\cap A^*)\rho_3\langle I_A, I_A, \kappa_1\rangle$.

This function replicates the initial word and transforms the third component in 1 (in fact $1^{|u|}$).

Let $g = (\cap(Z_1 \times Z_2 \times 1)^*)\langle g_1, I_{Z_2}, \kappa_2\rangle + (\cap(Z_1 \times Z_2 \times 2)^*)\langle I_{Z_1}, g_2, \kappa_1\rangle$.

This function applies g_1 to the first component when the third is 1; it applies g_2 to the second component when the third is 2. It also transforms 1 in 2, and conversely 2 in 1 in the third component. The function g_1 being complete on Z_1^* and the function g_2 being complete on Z_2^*, the function g is included in $((Z_1 \times Z_2 \times 1)^* \times (Z_1 \times Z_2 \times 2)^*) \cup ((Z_1 \times Z_2 \times 2)^* \times (Z_1 \times Z_2 \times 1)^*)$ and is always defined on $(Z_1 \times Z_2 \times 1)^* \cup (Z_1 \times Z_2 \times 2)^*$.

Let $h = (\cap(Z_1 \times Z_2 \times 2)^*)(\Pi_1 h_1) + (\cap(Z_1 \times Z_2 \times 1)^*)(\Pi_2 h_2)$.

This function selects the first component and applies h_1 when the third is 2; it selects the second and applies h_2 when the third is 1.

We shall prove that $\tau_1 + \tau_2 = fg^+h$.

For all $u \in A^*$, for all $n \geq 0$, we have

$$\begin{aligned} ufg^{2n+1}h &= [u, u, 1^{|u|}]\ g^{2n+1}h \\ &= [ug_1^n, ug_2^n, 1^{|u|}]gh \\ &= [ug_1^{n+1}, ug_2^n, 2^{|u|}]h \\ &= ug_1^{n+1}h_1. \end{aligned}$$

For all $u \in A^*$, for all $n \geq 1$, we have

$$\begin{aligned} ufg^{2n}h &= [u, u, 1^{|u|}]\ g^{2n}h \\ &= [ug_1^n, ug_2^{n-1}, 2^{|u|}]gh \\ &= [ug_1^n, ug_2^n, 1^{|u|}]h \\ &= ug_2^n h_2. \end{aligned}$$

If $(\varepsilon, \varepsilon)$ belongs neither to τ_1 nor τ_2, it would be removed from the functions f, g or h. □

Lemma 4.3. *The class $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$ is closed under composition.*

Proof. Let τ_1 and τ_2 belonging to $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$. We shall first suppose that $(\varepsilon, \varepsilon)$ belongs to the transductions τ_1 and τ_2. Let $A_1^* \times B_1^*$ and $A_2^* \times B_2^*$ be monoids in which τ_1 and τ_2 are respectively included.

By Lemma 4.1, there exist an alphabet $Z_1 \supseteq A_1$, a function g_1 included in $Z_1^* \times Z_1^*$ and complete on Z_1^* and a morphism h_1 included in $Z_1^* \times B_1^*$, such that $\tau_1 = (\cap A_1^*)g_1^+h_1$. Idem for τ_2.

Then $\tau_1\tau_2$ is equal to $(\cap A_1^*)g_1^+h_1(\cap A_2^*)g_2^+h_2$.

The transduction $h_1(\cap A_2^*)g_2^+h_2$ belongs to $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$, then we can apply Lemma 4.1 again. The transduction $h_1(\cap A_2^*)g_2^+h_2$ is included in $Z_1^* \times B_2^*$. There exist an alphabet $Z_3 \supseteq Z_1$, a function g_3 included in $Z_3^* \times Z_3^*$ and complete on Z_3^* and a morphism h_3 included in $Z_3^* \times B_2^*$, such that $h_1(\cap A_2^*)g_2^+h_2 = (\cap Z_1^*)g_3^+h_3$.

Then $\tau_1\tau_2$ is equal to $(\cap A_1^*)g_1^+(\cap Z_1^*)g_3^+h_3 = (\cap A_1^*)g_1^+g_3^+h_3$.

We only have to prove that $g_1^+g_3^+$ belongs to $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$.

Let us show the construction on diagrams. The Figure 3 shows how a word is transformed through $g_1^+g_3^+$.

As noticed in the Remark 3.6, $ug^+ = \{ug^i \mid 0 < i \leq k^{|u|}\}$ where k is the size of the alphabet used in g. Hence, the length of an horizontal iteration can be bounded by $k^{|u|}$. We count from 1 to $k^{|u|}$ by enumerating all the words of length $|u|$ built from an alphabet of k letters.

We shall use triplets. The function g_1 will act on the first component. For each word in the first component, we shall generate its images by the iteration of the function g_3 on the second component. The third component will count the number of actions of g_3 in order to know when we can generate a new image in the first component. The Figure 4 shows how will act fg^+h on these triplets.

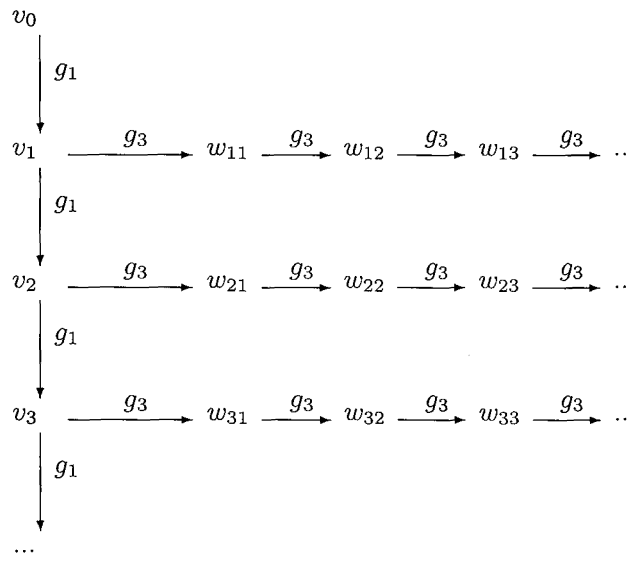


FIGURE 3. Generation of images through $g_1^+ g_3^+$.

Let $k = \|Z_3\|$.

Let $f = (\cap Z_1^*)\rho_3\langle I_{Z_1}, I_{Z_1}, \kappa_{z_k}\rangle$.

This function triples the starting word if it belongs to Z_1^*. Then it put z_k in the third component.

Let $g = (\cap\Gamma_1^*)\delta_{(1,1,3)}\langle g_1, g_1 g_3, \kappa_{z_1}\rangle + (\cap\Gamma_2^* \setminus \Gamma_1^+)\langle I_{Z_1}, g_3, \mathrm{Succ}_{Z_3}\rangle$ where $\Gamma_1 = Z_1 \times Z_3 \times \{z_k\}$ and $\Gamma_2 = Z_1 \times Z_3 \times Z_3$.

When the third component belongs to z_k^*, the function makes a step in the iteration of g_1 (in the first component) and start that of g_3 (in the second component). The iteration of g_3 will be repeated $k^{|u|} - 1$ times. In order to count the iterations of g_3, we start with a word in z_1^* and we shall use the function Succ_{Z_3}.

When the third component does not belong to z_k^+, the function makes a step in the iteration of g_3 in the second component and increases the third component.

Let $h = \Pi_2$.

We shall prove that $g_1^+ g_3^+ = f g^+ h$.

For all $u \in Z_1^*$, for all $n \geq 1$, we have

$$uf g^n h = u g_1^i \, g_3^j \text{ with } i = ((n-1) div \, k^{|u|}) + 1 \text{ and } j = ((n-1) mod \, k^{|u|}) + 1.$$

If $(\varepsilon, \varepsilon)$ did not belong to τ_1 or τ_2, it would be removed from the functions f, g or h. □

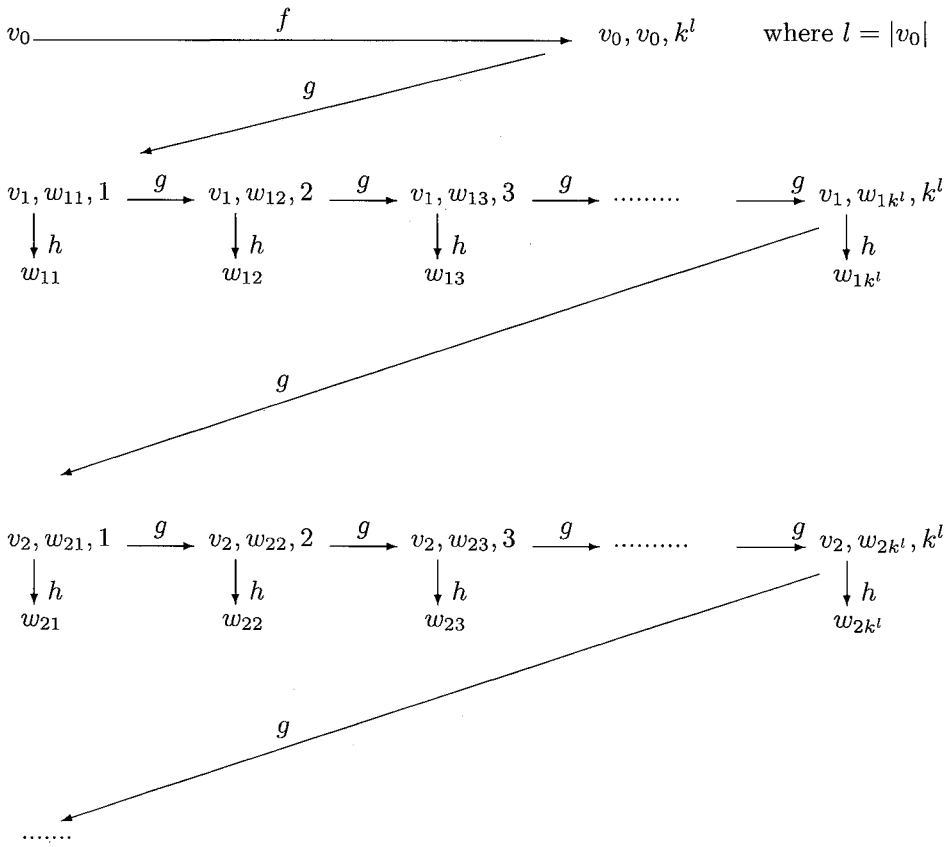


FIGURE 4. Generation of the same images through fg^+h.

Using these two lemmas, we can give a characterization of the class $UC(\mathcal{F}_+)$.

Proposition 4.4. *The class $UC(\mathcal{F}_+)$ is equal to the class $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$. That means, a transduction τ belongs to $UC(\mathcal{F}_+)$ if and only if $\tau = f_1 f_2^+ f_3$ for some f_1, f_2 and f_3 in $\mathcal{F}$.*

Proof. The class of length-preserving rational functions is included in $\mathcal{F}_+\ \mathcal{F}_+$ (same arguments than in Rem. 3.2). Then we have $\mathcal{F} \subset UC(\mathcal{F}_+)$ and $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F} \subseteq UC(\mathcal{F}_+)$.

On the other hand, the class $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$ includes $\mathcal{F}_+$ and is closed by union and composition (Lem. 4.2 and Lem. 4.3) then the class $\mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$ contains $UC(\mathcal{F}_+)$. □

5. Subsequential functions

We shall improve the characterization of the class $UC(\mathcal{F}_+)$.

Proposition 5.1. *The class $UC(\mathcal{F}_+)$ is equal to the class $_s\mathcal{S}\ _s\mathcal{S}_+\ \mathcal{H}_{sa} = UC(_s\mathcal{S}_+)$. That means, a transduction τ belongs to $UC(\mathcal{F}_+)$ if and only if $\tau = f_1 f_2^+ f_3$ for some f_1 and f_2 in $_s\mathcal{S}$ and f_3 in $\mathcal{H}_{sa}$.*

More precisely, the proof shows that $UC(\mathcal{F}_+) = {}_s\mathcal{S}\ {}_{ts}\mathcal{S}_+\ \mathcal{H}_{sa}$.

Proof. Obviously $_s\mathcal{S}\ _s\mathcal{S}_+\ \mathcal{H}_{sa}$ is included in $UC(_s\mathcal{S}_+)$ which is included in $UC(\mathcal{F}_+)$.

In order to prove the reverse inclusion, let us assume that a transduction $\tau \in UC(\mathcal{F}_+)$ contains $(\varepsilon, \varepsilon)$. All subsequential functions we will define, will contain $(\varepsilon, \varepsilon)$.

We use the characterization of the Lemma 4.1: there exist an alphabet Z including A, a function g included in $Z^* \times Z^*$ and complete on Z^* and a morphism h included in $Z^* \times B^*$ such that $\tau = (\cap A^*)g^+h$. We shall prove that $(\cap A^*)g^+h$ is equal to $f_1 f_2^+ f_3$ for some $f_1 \in {}_s\mathcal{S}$, $f_2 \in {}_{ts}\mathcal{S}$ and $f_3 \in \mathcal{H}_{sa}$.

The Figure 5 shows how a word is transformed through $(\cap A^*)g^+h$.

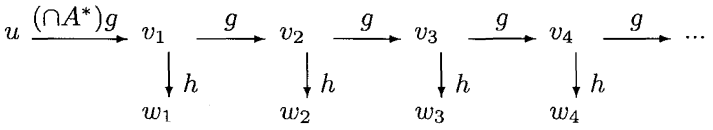


FIGURE 5. Generation of images through $(\cap A^*)g^+h$.

By the "Decomposition Theorem", we can find a letter-to-letter right sequential function s_r and a letter-to-letter left sequential function s_l such that the function g is equal to $s_r s_l$. Let $(Z, Y, Q, \mu_r, \lambda_r)$ be a right transducer which realizes s_r. We can suppose that the initial state q_0 of that right transducer cannot be reached by the other states. Thus, in a path, the state q_0 can only appear once, associated with the rightmost letter.

A first subsequential function f_1 will mark the last letter of the words and will associate to them the initial state of the right transducer. For all $u \in A^*$ and all $y \in A$, $uy\, f_1 = u(\overline{y}, q_0)$.

The second subsequential function is defined in such manner that its iteration realizes $(s_r s_l)^+$. The function f_2 simulates one transition of the right sequential transducer, bringing the state on the left. When the state is on the first letter the function f_2 applies the last transition of the right transducer and the left sequential function. It is defined as follow:

for all $(\overline{x}, q_0) \in \overline{Z} \times q_0$, $\quad (\overline{x}, q_0)\ f_2 = (\overline{x s_r s_l}, q_0) = (\overline{xg}, q_0)$,

for all $u_1 z(\overline{x}, q_0) \in Z^+(\overline{Z} \times q_0)$, $\quad u_1 z(\overline{x}, q_0)\ f_2 = u_1(z, q')\overline{x}'$

where $x' = (q_0, x)\lambda_r$ and $q' = (q_0, x)\mu_r$,

for all $u_1 z(x,q)u_2\overline{y} \in Z^+(Z\times Q)Y^*\overline{Y}$, $u_1 z(x,q)u_2\overline{y}\ f_2 = u_1(z,q')x'u_2\overline{y}$ where $x' = (q,x)\lambda_r$ and $q' = (q,x)\mu_r$,

for all $(x,q)u_2\overline{y} \in (Z\times Q)Y^*\overline{Y}$, $(x,q)u_2\overline{y}\ f_2 = v_1'(\overline{y}',q_0)$ where $v_1'y' = x'u_2y\ s_l$ and $x' = (q,x)\lambda_r$.

The function f_2 is ts-sequential; it is sufficient to delay the output of one letter and to overtake the delay on the letter which brings the state. Let us note also that a ts-sequential function can associate the state q_0 to the rightmost letter because this letter is marked.

Let us remark that $vy = ux\,g$ implies $v(\overline{y},q_0) \in u(\overline{x},q_0)\,f_2^+$ and conversely $v(\overline{y},q_0) \in u(\overline{x},q_0)f_2^+$ implies $vy \in ux\,g^+$.

The last morphism f_3 is defined on $Z\cup(\overline{Z}\times q_0)$ by $xf_3 = xh$ and $(\overline{x},q_0)f_3 = xh$ for all $x\in Z$.

The function f_1 (resp. f_2 and f_3) belongs to ${}_s\mathcal{S}$ (resp. to ${}_{ts}\mathcal{S}$ and to $\mathcal{H}_{sa}$).

The Figure 6 shows how a word is transformed through $f_1f_2^+f_3$.

We can verify that $\tau = (\cap A^*)g^+h = f_1f_2^+f_3$.

The following properties are equivalent:

$wz \in ux\,(\cap A^*)g^+h$;

$ux \in A^+$ and $\exists vy \in Z^+$ such that $vy \in ux\,g^+ = ux\,(s_rs_l)^+$ and $vy\,h = wz$;

$ux \in A^+$ and $\exists vy \in Z^+$ such that $v(\overline{y},q_0) \in u(\overline{x},q_0)\,f_2^+$ and $v(\overline{y},q_0)\,f_3 = wz$;

$wz \in ux\,f_1f_2^+f_3$.

If $(\varepsilon,\varepsilon)$ does not belong to the transduction τ, we can remove $(\varepsilon,\varepsilon)$ from the function f_1. □

In the previous proof, only the first function f_1 which marks the last letter, is really subsequential. When the last letters of words are marked, we can use ts-sequential functions rather than subsequential functions. We could easily prove that $(\cap X^*\overline{X})\ {}_s\mathcal{S} = (\cap X^*\overline{X})\ {}_{ts}\mathcal{S}$.

In order to obtain a marked word, we can iterate ts-sequential functions. The function which transforms $ux \in X^+$ into $u\overline{x} \in X^*\overline{X}$ belongs to ${}_{ts}\mathcal{S}\ {}_{ts}\mathcal{S}_+\ {}_{ts}\mathcal{S}$. Then, we could easily verify that $UC(\mathcal{F}_+) = {}_{ts}\mathcal{S}\ {}_{ts}\mathcal{S}_+\ {}_{ts}\mathcal{S}\ {}_{ts}\mathcal{S}_+\ \mathcal{H}_{sa}$. But we cannot apply the techniques used in the proof of Lemma 4.3 to show that ${}_{ts}\mathcal{S}_+\ {}_{ts}\mathcal{S}\ {}_{ts}\mathcal{S}_+ \subseteq {}_{ts}\mathcal{S}\ {}_{ts}\mathcal{S}_+\ {}_{ts}\mathcal{S}$. Nevertheless, we can prove the equality of the families $UC(\mathcal{F}_+)$ and $UC({}_{ts}\mathcal{S}_+)$ as done in the next section.

6. TS-SEQUENTIAL FUNCTIONS

The Decomposition Theorem characterizes a function φ as the composition of a right sequential function s_r followed by a left sequential one s_l. Moreover, if φ is included in $Z^*\times Z^*$ and complete on Z^*, then each word (and thus *any prefix of any word*) on Z^* has an image through $(s_rs_l)^n$, for any $n\geq 1$.

We shall use this property in the proof of the following result. Instead of marking the last letter of a word, we shall work on all his prefixes.

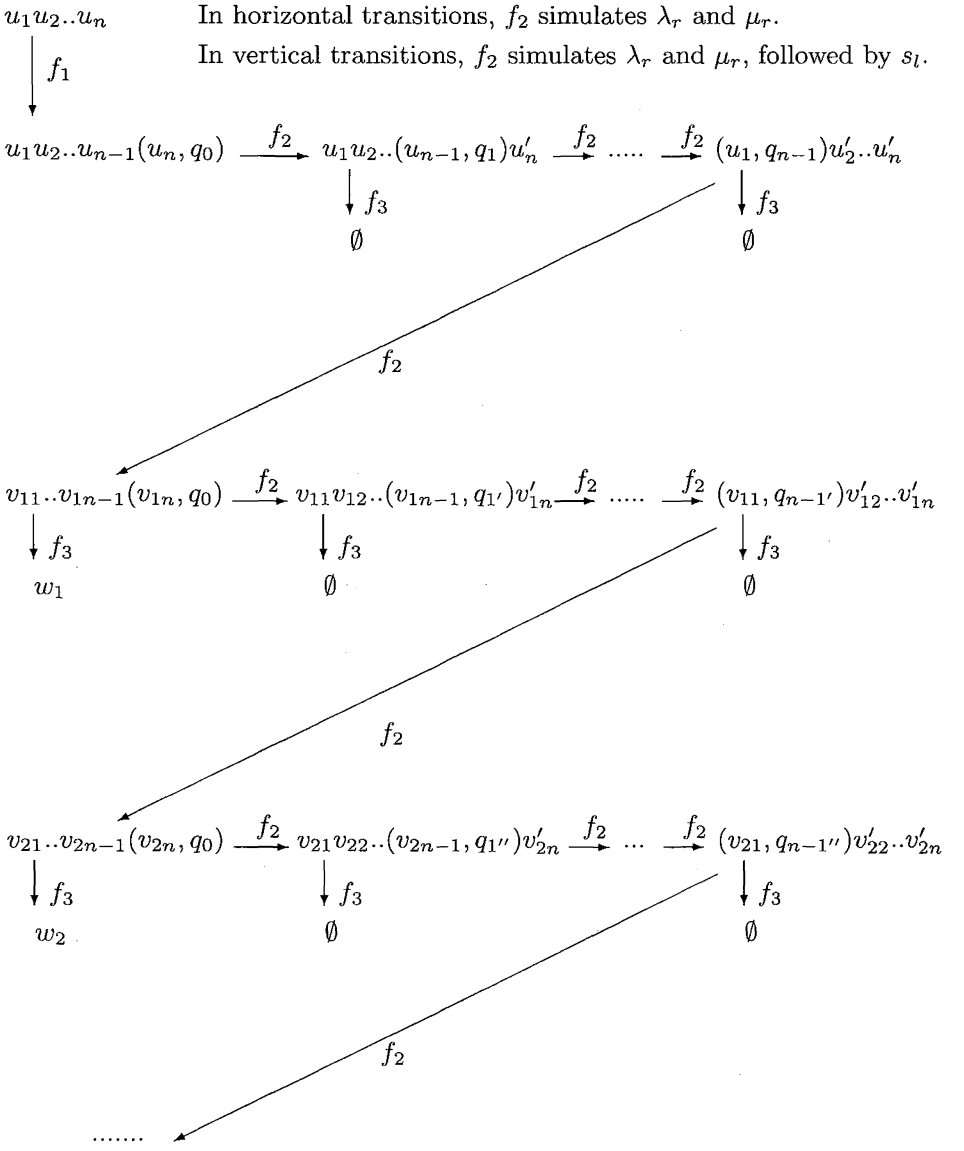


FIGURE 6. Generation of the same images through $f_1 f_2^+ f_3$.

Proposition 6.1. *The class $UC(\mathcal{F}_+)$ is equal to the class ${}_{ts}\mathcal{S}\ {}_{ts}\mathcal{S}_+\ \mathcal{H}_{sa}$. That means, a transduction τ belongs to $UC(\mathcal{F}_+)$ if and only if $\tau = f_1 f_2^+ f_3$ for some f_1 and f_2 in ${}_{ts}\mathcal{S}$ and f_3 in $\mathcal{H}_{sa}$.*

Proof. Obviously $_{ts}\mathcal{S}$ $_{ts}\mathcal{S}_+$ $\mathcal{H}_{sa}$ is included in $UC(_{ts}\mathcal{S}_+)$ which is included in $UC(\mathcal{F}_+)$.

In order to prove the reverse inclusion, let us assume that a transduction $\tau \in UC(\mathcal{F}_+)$ contains $(\varepsilon, \varepsilon)$. All deterministic functions we will define, will contain $(\varepsilon, \varepsilon)$.

We use the characterization of the Lemma 4.1: there exist an alphabet $Z = \{z_1, z_2, ..., z_k\}$ including A, a function g included in $Z^* \times Z^*$ and complete on Z^* and a morphism h included in $Z^* \times B^*$ such that $\tau = (\cap A^*)g^+h$. We shall prove that $(\cap A^*)g^+h$ is equal to $f_1 f_2^+ f_3$ for some $f_1 \in {}_{ts}\mathcal{S}$, $f_2 \in {}_{ts}\mathcal{S}$ and $f_3 \in \mathcal{H}_{sa}$.

By the "Decomposition Theorem", we can find a letter-to-letter right sequential function s_r and a letter-to-letter left sequential function s_l such that the function g is equal to $s_r s_l$. Let $(Z, Y, Q, \mu_r, \lambda_r)$ be a right transducer which realizes s_r.

In order to realize g^+, we shall work on 7-folds

$$[u, v_{p+1}v_{r-1}, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, cpt, w, \alpha].$$

The first component will memorize the initial word and will never be modified. In the second one, we will make g^+ as the iteration $(s_r s_l)^+$. But this iteration will act on the prefixes of u of length $p+1$ where p will grow from 0 to $|u|-1$. In the third component, a word 1^p0^r will store the length p which is completely treated. The fourth component stores the same information shifted to $\underline{1}^{p+1}\underline{0}^{r-1}$; it will be used to define the domain of the final morphism (not defined on $\underline{0}$). The fifth component contains the number of iterations of g which are done for the actual length $p+1$. The sixth component will be used to the transmission of informations such that the state reached in the right sequential transducer or the result of the control of the counter... The seventh one will store the action in process. It will be useful to determine from the first letter what kind of action is actually made: simulation of the left sequential transduction or control of the counter...

We shall prove that, for all u in A^*, ug^+ is equal to $uf_1(f_{21}+f_{22}+f_{23}+f_{24})^+f_3$.

The function f_1 is defined by $uf_1 = [u, u, 0^{|u|}, \underline{1}\,\underline{0}^{|u|-1}, z_k{}^{|u|}, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}]$ for all $u \in A^+$. It is a length-preserving ts-sequential function.

The morphism f_3 is defined by $(\cap(Z \times Z \times \{1,0\} \times \{\underline{1}\} \times Z \times \{\overleftarrow{\$_n}, \overleftarrow{\$_r}, \sharp\} \times \{\$_?\})^*)\Pi_2 h$.

The functions f_{2i} are defined below.

The transformation realized by the function f_{21}:

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, \overrightarrow{q_0}\sharp^{p+r-1}, \overrightarrow{q_0}^{p+r}]f_{21} = [u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, \sharp^p q_0 \sharp^{r-1}, s_r^{p+r}].$$

The function f_{21} associates the state q_0 to the first letter which contains a 0 in the third component. In parallel, it put the symbol s_r in every seventh component of the letters. The function f_{21} is ts-sequential.

The transformation realized by the function f_{22} (which simulates the transitions of the right sequential transducer s_r):

$$[u, v_{p_1} z v_{p_2+r-1}, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, \sharp^{p_1} q \sharp^{p_2+r-1}, s_r^{p+r}] f_{22}$$
$$= [u, v_{p_1}((q,z)\mu_r) v_{p_2+r-1}, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, \sharp^{p_1-1}((q,z)\lambda_r)\sharp^{p_2+r}, s_r^{p+r}]$$
with $p_1 > 0$, $p_2 \geq 0$ and $p = p_1 + p_2$,
$v_{p_1} \in Z^{p_1}$ and $v_{p_2+r-1} \in Y^{p_2} Z^{r-1}$

$$[u, z v_{p+r-1}, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, q\sharp^{p+r-1}, s_r^{p+r}] f_{22}$$
$$= [u, ((q,z)\mu_r) v_{p+r-1}, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, \sharp^{p+r}, s_l^{p+r}].$$
with $p \geq 0$ and $v_{p+r-1} \in Y^p Z^{r-1}$.

The first case concerns the transitions (except the last) of the right sequential transducer. The function f_{22} simulates one transition: image of a letter, moving of the state on the left.

The second case concerns the last transition of the right sequential transducer (image of the first letter); since the right sequential transducer has finished, it also put the symbol s_l in every seventh component of the letters. The function f_{22} is ts-sequential; it is sufficient to delay the output of one letter and to overtake the delay on the letter which brings the state.

The transformation realized by the function f_{23} (which simulates the transitions of the left sequential transducer s_l):

$$[u, v_{p+1} v_{r-1}, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w, \sharp^{p+r}, s_l^{p+r}] f_{23}$$
$$= [u, (v_{p+1} s_l) v_{r-1}, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w\mathrm{Succ}_Z, \overrightarrow{\$_?}\sharp^{p+r-1}, \$_?{}^{p+r}].$$

In the second component, the function f_{23} stores the image of v_{p+1} by the left sequential transducer. In the fifth one, the counter is increased by one. In the sixth one of the first letter, it put the $\overrightarrow{\$_?}$; this symbol will carry the information of the counter check. In parallel, it put the symbol $\$_?$ in every seventh component of the letters.

The transformation realized by the function f_{24} (which controls whether the counter has reached the bound k^{p+1} and prepares the word for a new iteration of g):

$$[u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^{p_1} z_k w_{p_2+r-1}, \sharp^{p_1}\overrightarrow{\$_?}\sharp^{p_2+r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^{p_1} z_k w_{p_2+r-1}, \sharp^{p_1+1}\overrightarrow{\$_?}\sharp^{p_2+r-2}, \$_?{}^{p+r}]$$
with $p_1 \geq 0$, $p_2 > 0$ and $p = p_1 + p_2$ and $w_l \in Z^l$

$$[u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^{p_1} z_i w_{p_2+r-1}, \sharp^{p_1}\overrightarrow{\$_?}\sharp^{p_2+r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^{p_1} z_i w_{p_2+r-1}, \sharp^{p_1+1}\overrightarrow{\$_n}\sharp^{p_2+r-2}, \$_?{}^{p+r}]$$
with $z_i \neq z_k$

$$[u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p_1} z w_{p_2+r-1}, \sharp^{p_1}\overrightarrow{\$_n}\sharp^{p_2+r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p_1} z w_{p_2+r-1}, \sharp^{p_1+1}\overrightarrow{\$_n}\sharp^{p_2+r-2}, \$_?{}^{p+r}]$$

$$[u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^p z_k w_{r-1}, \sharp^p\overrightarrow{\$_?}\sharp^{r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p 0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^p z_k w_{r-1}, \sharp^p\overleftarrow{\$_r}\sharp^{r-1}, \$_?{}^{p+r}]$$

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^p z_i w_{r-1}, \sharp^p \overrightarrow{\$_?} \sharp^{r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, z_k^p z_i w_{r-1}, \sharp^p \overleftarrow{\$_n} \sharp^{r-1}, \$_?{}^{p+r}] \text{ with } z_i \neq z_k$$

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_p z w_{r-1}, \sharp^p \overrightarrow{\$_n} \sharp^{r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_p z w_{r-1}, \sharp^p \overleftarrow{\$_n} \sharp^{r-1}, \$_?{}^{p+r}]$$

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \sharp^i \overleftarrow{\$_n} \sharp^j, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \sharp^{i-1} \overleftarrow{\$_n} \sharp^{j+1}, \$_?{}^{p+r}] \text{ with } 1 < i \leq p$$

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \sharp^i \overleftarrow{\$_r} \sharp^j, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \sharp^{i-1} \overleftarrow{\$_r} \sharp^{j+1}, \$_?{}^{p+r}]$$

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \overleftarrow{\$_n} \sharp^{p+r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \overrightarrow{q_0} \sharp^{p+r-1}, \overrightarrow{q_0}{}^{p+r}]$$

$$[u, v, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, w_{p+r}, \overleftarrow{\$_r} \sharp^{p+r-1}, \$_?{}^{p+r}] f_{24}$$
$$= [u, u, 1^{p+1}0^{r-1}, \underline{1}^{p+2}\underline{0}^{r-2}, w_{p+r}, \overrightarrow{q_0} \sharp^{p+r-1}, \overrightarrow{q_0}{}^{p+r}].$$

The first three transformations concern a symbol $\overrightarrow{\$}$ in the sixth component associated to a symbol 1 in the third. In these case, the symbol $\overrightarrow{\$}$ is shifted on the right. It keeps the mark $_?$ while symbols z_k are in the fifth component; the mark becomes $_n$ when a symbol is lower than z_k. The mark $_n$ means that the counter has not reached the bound k^{p+1}.

The next three transformations concern a symbol $\overrightarrow{\$}$ associated to the first 0 in the third component. If the symbol $\overrightarrow{\$}$ is still marked by $_?$ and if the symbol z_k is in the fifth component, the counter has reached the bound k^{p+1}. Then this length of prefixes is completely treated and it must start the length $p+2$. If the counter has not reached the bound k^{p+1}, it must go on with the iteration of g on the second component. In the two cases the symbol \$ must be brought back to the first letter with the information "reset" or "next".

The next two transformations bring the symbol $\overleftarrow{\$_r}$ or the symbol $\overleftarrow{\$_n}$ back to the first letter.

The last two transformations put the symbol $\overrightarrow{q_0}$ for a new application of s_r and s_l. If the counter had reached the bound, the word u must be duplicate from the first to the second component and the first 0 in the third and fourth component must be changed into 1 to indicate the new length of prefixes to treat (it is not necessary to put the counter to 0, remember that $0 \equiv z_k^n$).

The function f_{24} is ts-sequential; it is sufficient to delay the output of one letter and to overtake the delay on the letter which brings the symbol \$ in sixth component.

Now let us analyze the transduction $f_1(f_{21}+f_{22}+f_{23}+f_{24})^+f_3$.
Using the definition of f_3, it is equal to

$$f_1(f_{21}+f_{22}+f_{23}+f_{24})^+(\cap(Z\times Z\times\{1,0\}\times\{\underline{1}\}\times Z\times\{\overleftarrow{\$_n},\overleftarrow{\$_r},\sharp\}\times\{\$_?\})^*)\Pi_2 h.$$

By studying the evolution of the third and fourth component which have the form 1^p0^r and $\underline{1}^{p+1}\underline{0}^{r-1}$, we verify that it is equal to

$$f_1(f_{21}+f_{22}+f_{23}+f_{24})^+(\cap\ Z^*\times Z^*\times 1^*0\times\underline{1}^*\times Z^*\times\{\overleftarrow{\$_n},\overleftarrow{\$_r},\sharp\}^*\times\$_?^*)\Pi_2 h.$$

In the sixth component, when the symbol $\overleftarrow{\$}$ has appeared, it is only brought back to the first letter. Thus it is equal to

$$f_1(f_{21}+f_{22}+f_{23}+f_{24})^+(\cap\ Z^*\times Z^*\times 1^*0\times\underline{1}^*\times Z^*\times\{\overleftarrow{\$_n},\overleftarrow{\$_r}\}\sharp^*\times\$_?^*)\Pi_2 h.$$

The seventh component is different for each function. Its value is $\overrightarrow{q_0}^{|u|}$ after f_1. Then we can only apply f_{21} and its value will be $s_r^{|u|}$,... The sixth component needs the application of f_{24} to belong to $\{\overleftarrow{\$_n},\overleftarrow{\$_r}\}\sharp^*$. Then the transduction is equal to

$$f_1(f_{21}f_{22}^+f_{23}f_{24}^+)^+(\cap\ Z^*\times Z^*\times 1^*0\times\underline{1}^*\times Z^*\times\{\overleftarrow{\$_n},\overleftarrow{\$_r}\}\sharp^*\times\$_?^*)\Pi_2 h.$$

Now let us see some properties of the iteration $(f_{21}f_{22}^+f_{23}f_{24}^+)^+$. We can verify that, for any $p\geq 0$ and for any value of the counter cpt,

$$[u,v_{p+1}v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt,\overrightarrow{q_0}\sharp^{|u|-1},\overrightarrow{q_0}^{|u|}]f_{21}$$
$$=[u,v_{p+1}v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt,\sharp^p q_0\sharp^{r-1},s_r^{|u|}]$$

$$[u,v_{p+1}v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt,\overrightarrow{q_0}\sharp^{|u|-1},\overrightarrow{q_0}^{|u|}]f_{21}f_{22}^{p+1}$$
$$=[u,(v_{p+1}s_r)v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt,\sharp^{|u|},s_l^{|u|}].$$

(Let us remark that $f_{22}^{i<p+1}$ lets the seventh component unchanged at s_r and $f_{22}^{i>p+1}$ is impossible since the seventh component has become s_l. Remember also that $v_{p+1}s_r$ is always defined)

$$[u,v_{p+1}v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt,\overrightarrow{q_0}\sharp^{|u|-1},\overrightarrow{q_0}^{|u|}]f_{21}f_{22}^{p+1}f_{23}$$
$$=[u,(v_{p+1}s_rs_l)v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt+1,\overrightarrow{\$_?}\sharp^{|u|-1},\$_?^{|u|}]$$

$$[u,v_{p+1}v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt,\overrightarrow{q_0}\sharp^{|u|-1},\overrightarrow{q_0}^{|u|}]f_{21}f_{22}^{p+1}f_{23}f_{24}^{2p+1}$$
$$=[u,(v_{p+1}s_rs_l)v_{r-1},1^p0^r,\underline{1}^{p+1}\underline{0}^{r-1},cpt+1,\overleftarrow{\$_\alpha}\sharp^{|u|-1},\$_?^{|u|}]$$
where $\$_\alpha\in\{\overleftarrow{\$_n},\overleftarrow{\$_r}\}$.

(Let us remark that $f_{24}^{i<2p+1}$ does not give a sixth component belonging to $\{\overleftarrow{\$_n},\overleftarrow{\$_r}\}\sharp^*$ and that $v_{p+1}s_rs_l$ is always defined.)

And the cycle can start again with the last application of f_{24}.
If the counter (cpt+1) is lower than $||Z||^{p+1}$

$$[u, v_{p+1}v_{r-1}, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, cpt, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}]f_{21}f_{22}^{p+1}f_{23}f_{24}^{2p+2}$$
$$= [u, (v_{p+1}s_rs_l)v_{r-1}, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, cpt+1, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}].$$

If the counter (cpt+1) is equal to $||Z||^{p+1}$

$$[u, v_{p+1}v_{r-1}, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, cpt, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}]f_{21}f_{22}^{p+1}f_{23}f_{24}^{2p+2}$$
$$= [u, u, 1^{p+1}0^{r-1}, \underline{1}^{p+2}\underline{0}^{r-2}, 0, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}].$$

So we have for all $0 \leq p < |u|$ and for all $0 \leq i < ||Z||^{p+1}$

$$[u, u, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, 0, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}](f_{21}\, f_{22}^{p+1}\, f_{23}\, f_{24}^{2p+2})^i\, f_{21}\, f_{22}^{p+1}\, f_{23}\, f_{24}^{2p+1}$$
$$= [u, u_{p+1}(s_rs_l)^{i+1}u_{r-1}, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, i+1, \overleftarrow{\$_\alpha}\sharp^{|u|-1}, \$_?^{|u|}]$$
$$\text{where } \$_\alpha \in \{\overleftarrow{\$_n}, \overleftarrow{\$_r}\}$$

and, for all $0 \leq p < |u| - 1$

$$[u, u, 1^p0^r, \underline{1}^{p+1}\underline{0}^{r-1}, 0, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}](f_{21}\, f_{22}^{p+1}\, f_{23}\, f_{24}^{2p+2})^{||Z||^{p+1}}$$
$$= [u, u, 1^{p+1}0^{r-1}, \underline{1}^{p+2}\underline{0}^{r-2}, 0, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}].$$

To sum up these properties and conclude, we have the following equivalences:

- v belongs to $uf_1(f_{21} + f_{22} + f_{23} + f_{24})^+f_3$
- v belongs to $[u, u, 0^{|u|}, \underline{1}\underline{0}^{|u|-1}, 0, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}](f_{21}f_{22}^+f_{23}f_{24}^+)^+(\cap\ R)\Pi_2 h$

  $$\text{where } R = Z^* \times Z^* \times 1^*0 \times \underline{1}^* \times Z^* \times \{\overleftarrow{\$_n}, \overleftarrow{\$_r}\}\sharp^* \times \$_?^*$$

- v belongs to
  $[u, u, 1^{|u|-1}0, \underline{1}^{|u|}, 0, \overrightarrow{q_0}\sharp^{|u|-1}, \overrightarrow{q_0}^{|u|}](f_{21}\, f_{22}^{|u|}\, f_{23}\, f_{24}^{2|u|})^i\, f_{21}\, f_{22}^{|u|}\, f_{23}\, f_{24}^{2|u|-1}\Pi_2 h$

  $$\text{with } 0 \leq i < ||Z||^{|u|}$$

- v belongs to $u(s_rs_l)^+h = ug^+h$.

We have proved that $(\cap A^*)g^+h$ is equal to $f_1(f_{21} + f_{22} + f_{23} + f_{24})^+f_3$. Since the alphabets of the domains are differents (due to the seventh component) and since $(\varepsilon, \varepsilon)$ belongs to each function, $(f_{21} + f_{22} + f_{23} + f_{24})$ is a ts-sequential function.

If $(\varepsilon, \varepsilon)$ does not belong to the transduction τ, we can remove $(\varepsilon, \varepsilon)$ from the functions f_1, f_{2i} and f_3. □

Using marked alphabet, we obtain:

Proposition 6.2. *The class $UC(\mathcal{F}_+)$ is equal to the class $(\cap X^*)\ {}_{ts}\mathcal{S}_+\ \mathcal{H}_{sa}$. That means a transduction τ belongs to $UC(\mathcal{F}_+)$ if and only if $\tau = (\cap X^*)f^+h$ for some f in ${}_{ts}\mathcal{S}$ and h in $\mathcal{H}_{sa}$.*

Remark 6.3. The class $(\cap X^*)\ {}_{ts}\mathcal{S}_+\ \mathcal{H}_{sa}$ strictly contains the class $(\cap X^*)\ {}_{ts}\mathcal{S}_+$ $(\cap Y^*)$ even if $X \cap Y = \emptyset$.

7. Deterministic context-sensitive languages

The context-sensitive languages are recognized by *linear-bounded automata.* A linear-bounded automaton is sextuple $M = (Q, X, Y, \delta, q_0, Q_F)$ where Q is the set of states, X is the input alphabet, $Y \supseteq X$ is the working alphabet, δ is the transition map included in $(Q \times Y) \times (Q \times Y \times \{-1, 0, 1\})$, q_0 is the initial state and Q_F is the set of final states.

A step in the recognition of a word is given by δ:

- $ubqav \mapsto uq'ba'v$ if $(q', a', -1) \in (q, a)\delta$, $u, v \in Y^*$, $a, b \in Y$;
- $uqav \mapsto uq'a'v$ if $(q', a', 0) \in (q, a)\delta$, $u, v \in Y^*$, $a \in Y$;
- $uqav \mapsto ua'q'v$ if $(q', a', 1) \in (q, a)\delta$, $u, v \in Y^*$, $a \in Y$.

The language recognized by M is defined by iteration of δ

$$L(M) = \{u \mid q_0 u \overset{*}{\longmapsto} wq_f \text{ for some } w \in Y^* \text{ and } q_f \in Q_F\} \cdot$$

A LBA is said to be deterministic if $||(q, a)\delta|| \leq 1$ for all q in Q and a in Y.

We shall now establish a new characterization of the class of deterministic context-sensitive languages in terms of iterations of functions.

Proposition 7.1. *The class of deterministic context-sensitive languages is equal to $a^*UC(\mathcal{F}_+)$ where a is a letter.*

Proof. We prove the equality by two inclusions: Proposition 7.3 and Lemma 7.5. □

Lemma 7.2. *Let g be a length-preserving rational function included in $X^* \times Y^*$. The language $\{[u, v] \mid u \in X^*, v \in ug^+\}$ belongs to $\mathcal{CS}_{\text{det}}$.*

Proof. Let $\dot{X}$ and $\ddot{X}$ be marked alphabets built from alphabet X. These alphabets will be used to distinguish the first and the last letters. In a first time, we omit the words having less than two letters.

We transform g into g_m in such way that g_m acts only on marked words and preserves the marks.

We shall verify that $L = \{[u, v] \mid u \in \dot{X}X^*\ddot{X}, v \in ug_m^+\}$ belongs to $\mathcal{CS}_{\text{det}}$.

We shall define the transitions of a LBA which recognizes L. Starting from a word $[u, v]$, the LBA will iterate the function g on the first component. After each application of g, the LBA will check up the contingent equality of the two components; when they are equal, the LBA goes into a final state and the word $[u, v]$ is recognized.

Using the "decomposition theorem", a length-preserving rational function including $(\varepsilon, \varepsilon)$ is the composition of a letter-to-letter right sequential transduction followed by a letter-to-letter left sequential transduction.

Since L contains only words having more than two letters, we may suppose that $(\varepsilon, \varepsilon)$ belongs to the function g_m.

Sequential transductions being ts-sequential, a letter-to-letter transition is easily changed into a transition of deterministic linear bounded acceptor.

We build the set of transitions of a LBA which recognizes L.

Transitions to go to the last letter: $\forall \alpha, \beta \in X$

$(q_{00}, (\dot{\alpha}, \dot{\beta})) \to (q_{00}, (\dot{\alpha}, \dot{\beta}), 1)$
$(q_{00}, (\alpha, \beta)) \to (q_{00}, (\alpha, \beta), 1)$
$(q_{00}, (\ddot{\alpha}, \ddot{\beta})) \to (q_{10}, (\ddot{\alpha}, \ddot{\beta}), 0).$

Transitions for the right sequential transducer: $\forall \alpha \in X$

$(q_{10}, (\ddot{x}, \ddot{\alpha})) \to (q_{1i}, (\ddot{y}, \ddot{\alpha}), -1)$ if $(q_0, x)\lambda_r = y$ and $(q_0, x)\mu_r = q_i$
$(q_{1i}, (x, \alpha)) \to (q_{1j}, (y, \alpha), -1)$ if $(q_i, x)\lambda_r = y$ and $(q_i, x)\mu_r = q_j$
$(q_{1i}, (\dot{x}, \dot{\alpha})) \to (q_{20}, (\dot{y}, \dot{\alpha}), 0)$ if $(q_i, x)\lambda_r = y$.

Transitions for the left sequential transducer: $\forall \alpha \in X$

$(q_{20}, (\dot{x}, \dot{\alpha})) \to (q_{2i}, (\dot{y}, \dot{\alpha}), 1)$ if $(q_0, x)\lambda_l = y$ and $(q_0, x)\mu_l = q_i$
$(q_{2i}, (x, \alpha)) \to (q_{2j}, (y, \alpha), 1)$ if $(q_i, x)\lambda_l = y$ and $(q_i, x)\mu_l = q_j$
$(q_{2i}, (\ddot{x}, \ddot{\alpha})) \to (q_{30}, (\ddot{y}, \ddot{\alpha}), 0)$ if $(q_i, x)\lambda_l = y$.

Transitions to return on the first letter: $\forall \alpha, \beta \in X$

$(q_{30}, (\ddot{\alpha}, \ddot{\beta})) \to (q_{30}, (\ddot{\alpha}, \ddot{\beta}), -1);$
$(q_{30}, (\alpha, \beta)) \to (q_{30}, (\alpha, \beta), -1);$
$(q_{30}, (\dot{\alpha}, \dot{\beta})) \to (q_{40}, (\dot{\alpha}, \dot{\beta}), 0).$

Test of contingent equality, transition to the state q_{10} in case of inequality: $\forall \alpha, \beta \in X$

$(q_{40}, (\dot{x}, \dot{x})) \to (q_{40}, (\dot{x}, \dot{x}), 1)$
$(q_{40}, (x, x)) \to (q_{40}, (x, x), 1)$
$(q_{40}, (\ddot{x}, \ddot{x})) \to (q_F, (\ddot{x}, \ddot{x}), 1)$
$(q_{40}, (\dot{x}, \dot{y})) \to (q_{50}, (\dot{x}, \dot{y}), 1)$ if $x \neq y$
$(q_{40}, (x, y)) \to (q_{50}, (x, y), 1)$ if $x \neq y$
$(q_{40}, (\ddot{x}, \ddot{y})) \to (q_{10}, (\ddot{x}, \ddot{y}), 0)$ if $x \neq y$
$(q_{50}, (\alpha, \beta)) \to (q_{50}, (\alpha, \beta), 1)$
$(q_{50}, (\ddot{\alpha}, \ddot{\beta})) \to (q_{10}, (\ddot{\alpha}, \ddot{\beta}), 0).$

It is easy to verify that

$q_{00}[u, v] \xrightarrow{*} [u', v']q_{10}(\ddot{x}', \ddot{y}')$ iff $u = u'\ddot{x}'$ and $v = v'\ddot{y}'$.
$[u', v']q_{10}(\ddot{x}', \ddot{y}') \xrightarrow{*} [u'', v']q_{30}(\ddot{x}'', \ddot{y}')$ iff $u'\ddot{x}'g_m = u''\ddot{x}''$ and $u'\ddot{x}' \in \dot{X}X^*\ddot{X}$.
$[u'', v'']q_{30}(\ddot{x}'', \ddot{y}'') \xrightarrow{*} [u''\ddot{x}'', v''\ddot{y}'']q_F$ iff $u''\ddot{x}'' = v''\ddot{y}''$.
$[u'', v'']q_{30}(\ddot{x}'', \ddot{y}'') \xrightarrow{*} [u'', v'']q_{10}(\ddot{x}'', \ddot{y}'')$ iff $u''\ddot{x}'' \neq v''\ddot{y}''$.

Thus the couple $[u, v]$ is recognized if and only if there exists $i \geq 1$ such that $v = ug_m^i$. The language L belongs to $\mathcal{CS}_{\text{det}}$.

The language $\{[u, v] \mid v \in ug^+\}$ also belongs to $\mathcal{CS}_{\text{det}}$; we obtain it from the language L by erasing the marks and adding the finite language $\{[u, v] \mid u \in X^{\leq 1}$ *and* $v \in ug^+\}$. □

Proposition 7.3. *The class $\mathcal{CS}_{\text{det}}$ is closed under $UC(\mathcal{F}_+)$.*

Proof. The class $\mathcal{CS}_{\text{det}}$ is closed under length-preserving rational function; we prove that it is also closed under iterated length-preserving rational function.

Let L be a language belonging to $\mathcal{CS}_{\text{det}}$. Let g be length-preserving rational function.

We have $Lg^+ = L\Pi_1^{-1}(\cap\{[u,v] \mid v \in ug^+\})\Pi_2$. The language $\{[u,v] \mid v \in ug^+\}$ belongs to $\mathcal{CS}_{\text{det}}$ (Lem. 7.2). The class $\mathcal{CS}_{\text{det}}$ is closed under inverse of letter-to-letter morphism, intersection and letter-to-letter morphism. Then the language Lg^+ belongs to $\mathcal{CS}_{\text{det}}$. □

When we want to produce context-sensitive languages, we can iterate a *ts*-sequential function to simulate the deterministic LBA.

Proposition 7.4. *For each deterministic context-sensitive language $L \subseteq X^*$, there exists a length-preserving ts-sequential function φ such that $L = \underline{X}^*\varphi^+(\cap X^*)$.*

Proof. Let $L \subseteq X^+$ be an ε-free context-sensitive language. There exists a deterministic $LBA = (Q, X, Y, \delta, q_0, Q_f)$ which recognizes L.

Our purpose is to check whether a word u is in the language L. We shall start from a word $\underline{u}$ and the iteration of the *ts*-sequential functions shall check his prefixes. For each prefix u_1x of u there will exist some $r \in \mathbb{N}$ such that

$$\begin{aligned} u_1\underline{xu_2}\varphi^r = \overline{u}_1\underline{xu_2}\varphi^r &= u_1x\underline{u_2} \quad \text{if } u_1x \in L \\ &= \overline{u_1x}\underline{u_2} \quad \text{if } u_1x \notin L. \end{aligned}$$

Let us remark that the recognition of a word (or a prefix of a word) by a LBA is realized with a finite number of steps depending on the length of the word (the prefix) which is treated, on the number of letters in the working alphabet and on the number of states. Like in other proofs, we shall count by enumerating all words written with an alphabet having enough letters to ensure us that all reachable configurations in the LBA have been obtained.

The algorithm of recognition of the word u will be the following

```
start with xu2
while it remains underlined letters do
*    prepare u1xu2 or u1xu2 in order to treat the prefix u1x
*    while not enough iterations do
*    *    apply one transition of the LBA and increase the counter
*    *    control the counter of iterations
*    check if the prefix is recognized
*    output the result of the recognition u1xu2 or u1xu2.
```

We shall use 5-folds $[u_1\underline{xu_2}, v_1\underline{yv_2}, counter, transmission, action]$.

The first component is the word u we want to test the recognition by testing the recognition of the prefixes u_1x. The second component will be the working memory of the LBA using a k-letters alphabet $Z = Y \cup (Q \times Y) \cup \underline{(Q \times Y)} \cup (Y \times \underline{Q})$. The words u_1x and $v_1\underline{y}$ have the same length. Thus the first underlined letter in the second component points out the last letter of the prefix which is actually treated and v_2 will always be equal to u_2. This underlined letter serves for end-marking and allows the transducers to take one letter delay.

The third component is the counter of iterations using the alphabet Z, the fourth one carries informations like control of the counter or control of recognition and the fifth one stores the action to do.

A first function g_1 prepares the 5-fold. The value 0 of the counter is the word $z_k^{|u|}$.

$$\underline{xu_2}g_1 = [x\underline{u_2}, \underline{(q_0, x)u_2}, 0, \sharp^{|u|}, L^{|u|}]$$

$$zu_1\underline{xu_2}g_1 = [zu_1x\underline{u_2}, (q_0, z)u_1\underline{xu_2}, 0, \sharp^{|u|}, L^{|u|}]$$

$$\overline{zu}_1\underline{xu_2}g_1 = [zu_1x\underline{u_2}, (q_0, z)u_1\underline{xu_2}, 0, \sharp^{|u|}, L^{|u|}].$$

A second function g_2 applies one transition of the LBA to the actually treated prefix.

We use a slight modification in our notation. The transition function δ is applied in a context. The configuration $ubqacv\underline{w}$ is denoted by $ub(q, a)cv\underline{w}$ and can give through δ one of the following images: $u(q', b)a'cv\underline{w}$, $ub(q', a')cv\underline{w}$ or $uba'(q', c)v\underline{w}$. The letter c could be underlined, v being ε. The rightmost application of δ on a word $u\underline{(q, a)w}$ can also give a word $u\underline{(q', a')w}$ or $u\underline{(a', q')w}$ if the transition indicates a right move.

The LBA being deterministic, there is at most one transition to apply at any time. The function δ is completed in order to be the identity in all cases not defined in the LBA.

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, L^{|u|}]g_2 = [u_1x\underline{u_2}, \underline{(v_1yv_2)\delta}, cpt + 1, \sharp^{|u|}, C^{|u|}].$$

The third function g_3 controls the counter to know whether the LBA must go on (counter $< ||Z||^{|u_1x|}$) or whether the recognition of the word has to be tested (counter $= ||Z||^{|u_1x|}$).

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, C^{|u|}]g_3 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u_1|}\overleftarrow{\$_r}\sharp^{|u_2|}, B^{|u|}]$$
$$\text{if } cpt \in z_k^{|u_1x|}Z^{|u_2|}$$

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, C^{|u|}]g_3 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u_1|}\overleftarrow{\$_n}\sharp^{|u_2|}, B^{|u|}]$$
$$\text{if } cpt \in Z^{<|u_1x|}z_iZ^* \text{ with } z_i \neq z_k.$$

The function g_4 brings the result of this control back to the first letter.

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^i\overleftarrow{\$_\alpha}\sharp^j, B^{|u|}]g_4 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{i-1}\overleftarrow{\$_\alpha}\sharp^{j+1}, B^{|u|}]$$
$$\text{with } i > 1 \text{ and } \overleftarrow{\$_\alpha} \in \{\overleftarrow{\$_n}, \overleftarrow{\$_r}\}$$

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \overleftarrow{\$_n}\sharp^{|u|-1}, B^{|u|}]g_4 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, L^{|u|}]$$

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \overleftarrow{\$_r}\sharp^{|u|-1}, B^{|u|}]g_4 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, R^{|u|}].$$

The function g_5 tests whether u_1x is recognized.

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, R^{|u|}]g_5 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u_1|}r\sharp^{|u_2|}, O^{|u|}]$$
$$\text{if } v_1\underline{yv_2} \in Y^{|u_1|}\underline{(Y \times Q_F)Y^{|u_2|}}$$

$$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u|}, R^{|u|}]g_5 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{|u_1|}\overline{r}\sharp^{|u_2|}, O^{|u|}] \quad \text{else.}$$

The function g_6 brings back the information. If the prefix is not recognized, it outputs the prefix with marked letters.

$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^i\alpha\sharp^j, O^{|u|}]g_6 = [u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \sharp^{i-1}\alpha\sharp^{j+1}, O^{|u|}]$
where $i > 0$ and $\alpha \in \{r, \overline{r}\}$.

$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, r\sharp^{|u|-1}, O^{|u|}]g_6 = u_1x\underline{u_2}$

$[u_1x\underline{u_2}, v_1\underline{yv_2}, cpt, \overline{r}\sharp^{|u|-1}, O^{|u|}]g_6 = \overline{u_1x}\underline{u_2}$.

Due to the sixth component which is different in each function, we could verify that

$$\underline{X}^+(g_1 + g_2 + g_3 + g_4 + g_5 + g_6)^+(\cap X^+) = \underline{X}^+(g_1(g_2g_3g_4^+)^+g_5g_6^+)^+(\cap X^+).$$

More precisely, for all $u = u_1\underline{xu_2} \in X^*\underline{X}^+$ or $u = \overline{u_1}\underline{xu_2} \in \overline{X}^*\underline{X}^+$, we have

$$\begin{aligned} ug_1(g_2g_3g_4^{|u_1x|})^{||Z||^{|u_1x|}}g_5g_6^{|u_1x|} &= u_1x\underline{u_2} \quad \text{if } u_1x \in L \\ &= \overline{u_1x}\underline{u_2} \quad \text{if } u_1x \notin L \\ ug_1(g_2g_3g_4^i)^jg_5g_6^k &\notin X^*\underline{X}^+ \cup \overline{X}^*\underline{X}^+ \\ &\quad \text{if } i \neq |u_1x| \text{ or } j \neq ||Z||^{|u_1x|} \text{ or } k \neq |u_1x|. \end{aligned}$$

Then, starting with $\underline{u}$, it suffices to use $|u|$ times this cycle and we obtain u if $u \in L$ and $\overline{u}$ if $u \notin L$. Thus L is equal to $\underline{X}^+(g_1 + g_2 + g_3 + g_4 + g_5 + g_6)^+(\cap X^+)$. If ε belongs to L, we can add the couple $(\varepsilon, \varepsilon)$ to the functions g_i.

All functions are *ts*-sequential and the alphabets of their domains are distinct. We can define the *ts*-sequential function $\varphi = g_1 + g_2 + g_3 + g_4 + g_5 + g_6$. □

Lemma 7.5. *For each deterministic context-sensitive language L, there exist a length-preserving rational function g and a morphism h such that $L = a^*\ g^+h$.*

Proof. Let L be a deterministic context-sensitive language included in X^*. The Proposition 7.4 shows that $L = \underline{X}^*\ g^+(\cap X^*)$.

It is obvious that $\underline{X}^* = a^*\kappa_{\underline{x_1}}\mathrm{Succ}^+_{\underline{X}} = a^*(\cap a^*)\kappa_{\underline{x_1}}\mathrm{Succ}^+_{\underline{X}}$. Then $L = a^*\tau_2$ where $\tau_2 = (\cap a^*)\kappa_{\underline{x_1}}\mathrm{Succ}^+_{\underline{X}}g^+(\cap X^*)$.

The transduction τ_2 belongs to $UC(\mathcal{F}_+) = \mathcal{F}\ \mathcal{F}_+\ \mathcal{F}$ and is included in $\{a\}^* \times X^*$. By Lemma 4.1, there exist a function g' and a morphism h', such that $\tau_2 = (\cap a^*)g'^+h'$. Thus $L = a^*g'^+h'$. □

8. Deterministic context-sensitive transductions

We define deterministic context-sensitive transductions as we have defined context-sensitive transductions [9].

Definition 8.1. Let τ be a transduction. The transduction τ is a deterministic context-sensitive transduction if and only if there exist two letter-to-letter morphisms φ, ψ and a deterministic context-sensitive language A such that $\tau = \varphi^{-1}(\cap A)\psi$.

In other words, the class of deterministic context-sensitive transductions is $\mathcal{H}_{sa}^{-1}(\cap\mathcal{CS}_{\text{det}})\,\mathcal{H}_{sa}$.

Since the class of deterministic context-sensitive languages is defined in terms of iterations of functions, we can easily prove the equality between the class of deterministic context-sensitive transductions and the class $UC(\mathcal{F}_+)$.

Proposition 8.2. *The class of deterministic context-sensitive transductions is equal to the class $UC(\mathcal{F}_+)$. That means $UC(\mathcal{F}_+) = \mathcal{H}_{sa}^{-1}(\cap\mathcal{CS}_{\text{det}})\,\mathcal{H}_{sa}$.*

Proof. The equality is proved by two inclusions: Lemmas 8.3 and 8.4. □

Lemma 8.3. *The class $UC(\mathcal{F}_+)$ is included in $\mathcal{H}_{sa}^{-1}(\cap\mathcal{CS}_{\text{det}})\,\mathcal{H}_{sa}$.*

Proof. Let $\sigma = fg^+h \in UC(\mathcal{F}_+)$. Let us suppose that $\sigma \subseteq X^* \times Y^*$.

The transduction $\tau = \rho_2\langle I_X, f\rangle\langle I_X, g\rangle^+\langle I_X, h\rangle$ belongs to $UC(\mathcal{F}_+)$.

The language $X^*\tau = \{[u, v] \mid v \in u\sigma\}$ is a deterministic context-sensitive language (Prop. 7.3).

The transduction σ is equal to $\Pi_1^{-1}(\cap\, X^*\tau)\,\Pi_2$ and belongs to $\mathcal{H}_{sa}^{-1}(\cap\mathcal{CS}_{\text{det}})\,\mathcal{H}_{sa}$. □

Lemma 8.4. *The class $\mathcal{H}_{sa}^{-1}(\cap\mathcal{CS}_{\text{det}})\,\mathcal{H}_{sa}$ is included in $UC(\mathcal{F}_+)$.*

Proof. We mainly have to prove that intersection with a deterministic context-sensitive language $L \subset X^*$ belongs to $UC(\mathcal{F}_+)$.

By Lemma 7.5, we know that there exist a letter a and two functions f_1 and f_2 such that $L = a^* f_1^+ f_2$. Then the intersection with the language L is realized by $\rho_2\langle I_X, \kappa_a\rangle\langle I_X, f_1\rangle^+\langle I_X, f_2\rangle(\cap(\{(x, x) \mid x \in X\})^*)\Pi_1$ which belongs to $UC(\mathcal{F}_+)$.

A letter-to-letter morphism is a length-preserving rational function and then belongs to $UC(\mathcal{F}_+)$; the inverse of a letter-to-letter morphism also belongs to $UC(\mathcal{F}_+)$ (Lem. 3.7). Since $UC(\mathcal{F}_+)$ is closed by composition, every deterministic context-sensitive transduction belongs to $UC(\mathcal{F}_+)$. □

We deduce from the Proposition 8.2 the following corollary:

Corollary 8.5. *The class $UC(\mathcal{F}_+)$ is closed by inverse.*

The closure properties of the class $\mathcal{CS}_{\text{det}}$ allows us to state the next corollary.

Corollary 8.6. *The class $UC(\mathcal{F}_+)$ is closed by intersection and difference.*

9. Conclusion

We have studied the UC-closure of some classes of iterated functions. This study states the kind of transductions we can obtain by iterating rational transductions. It also gives characterizations of the family of context-sensitive languages and of deterministic context-sensitive languages.

If a language L belongs to $\mathcal{CS}$, there exist two letter-to-letter *ts*-sequential functions g_1 and g_2 and an alphabet Y such that $L = a^*(g_1 + g_2)^+(\cap Y^*)$.

If a language L belongs to $\mathcal{CS}_{\text{det}}$, there exist a length-preserving ts-sequential function g and a letter-to-letter morphism h such that $L = a^*g^+h$.

But the well-known problem of the equality between context-sensitive languages and deterministic context-sensitive languages remains open.

We would like to thank Michel Latteux for supervising this work and his constant encouragements and help during the redaction. We also thank Jacques Sakarovitch for useful remarks on a previous version of this paper.

REFERENCES

[1] A. Arnold and M. Latteux, A new proof of two theorems about rational transductions. *Theoret. Comput. Sci.* **8** (1979) 261–263.

[2] J. Berstel, *Transductions and Context-Free Languages*. Teubner Studienbücher, Stuttgart (1979).

[3] S. Eilenberg, *Automata, Languages and Machines*, Vol. A. Academic Press, New York (1974).

[4] C.C. Elgot and J.E. Mezei, On relations defined by generalized finite automata. *IBM J. Res. Develop.* **9** (1965) 47–68.

[5] M. Latteux, D. Simplot and A. Terlutte, Iterated length-preserving rational transductions, in *Proc. 23rd Symposium on Mathematical Foundations of Computer Science (MFCS'98)* (Brno, Czech Republic, 1998), edited by L. Brim, J. Gruska and J. Zlatuška. Springer-Verlag, Berlin, *Lecture Notes in Comput. Sci.* **1450** (1998) 286–295.

[6] J. Leguy, Transductions rationnelles décroissantes. *Theoret. Informatics Appl.* **15** (1981) 141–148.

[7] M. Nivat, Transductions des langages de Chomsky. *Ann. Inst. Fourier (Grenoble)* **18** (1968) 339–456.

[8] M.P. Schützenberger, Sur les relations rationnelles entre monoïdes libres. *Theoret. Comput. Sci.* **3** (1976) 243–259.

[9] D. Simplot and A. Terlutte, Iterations of Transductions. *Theoret. Informatics Appl.* **34** (2000) 99–130.

[10] D. Wood, Iterated a-NGSM maps and Γ systems. *Inform. and Control* **32** (1976) 1–26.

Communicated by Ch. Choffrut.
Received April, 1999. Accepted June 26, 2000.