INFORMATIQUE THÉORIQUE ET APPLICATIONS

ALAIN TERLUTTE
DAVID SIMPLOT

Iteration of rational transductions

NUMDAM

ITERATION OF RATIONAL TRANSDUCTIONS

ALAIN TERLUTTE[1] AND DAVID SIMPLOT[1]

Abstract. The purpose of this paper is to show connections between iterated length-preserving rational transductions and linear space computations. Hence, we study the smallest family of transductions containing length-preserving rational transductions and closed under union, composition and iteration. We give several characterizations of this class using restricted classes of length-preserving rational transductions, by showing the connections with "context-sensitive transductions" and transductions associated with recognizable picture languages.

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

1. INTRODUCTION

The family of rational languages turns out to be one of the most important classes within the Chomsky hierarchy. Finite automata that are the main object for studying rational languages are now used in most domains of computer science. Rational transductions introduced by Elgot and Mezei [5] are a natural extension of rational languages and were very useful to represent several kinds of computations. For instance, the addition of two integers in any base can be realized by a rational transduction and the study of these tools has led to efficient parallel algorithms (see [1,17] for instance).

The theory of rational transductions was mainly developed by Schützenberger, Eilenberg and Nivat (see [4,18,19]). This theory is now well established and its basic results can be found in [2,4]. More recently, some representation theorems were achieved in terms of compositions of morphisms and inverse morphisms [8,23].

At the contrary, there are only a few papers dealing with iteration of rational transductions (see [7,24]). Since several mechanisms of computation are actually iterations of rational transductions, it seems that this study deserves to be undertaken. For instance, finitely generated congruences, derivations in a grammar,

---

[1] UPRESA 8022 du CNRS, LIFL, Université de Lille I, bâtiment M3, Cité Scientifique, 59655 Villeneuve-d'Ascq Cedex, France; e-mail: {`simplot,terlutte`}@lifl.fr

partial commutation – as well as semi-commutation –, L-systems are examples of such mechanisms.

The set of transductions equipped with the operation of composition has a semigroup structure closed under iteration. The subset of rational transductions is closed under composition but not under iteration. We are mainly interested by the the smallest set of transductions closed under union, composition and iteration and containing the rational transductions. Indeed to obtain several interesting transductions, both rational transductions and iterated rational transductions are needed. For instance, the mirror operation is shown in the preliminaries to be such a transduction. It is neither a rational transduction nor an iterated rational transduction but it can be realized by composition of these two kinds of transductions. Note also that iterations introduce non-determinism. So it is established in Section 8 (Lem. 8.1) that the inverse of a letter-to-letter morphism can be obtained as the composition of an iterated rational function with a letter-to-letter morphism.

In this paper, we shall restrict ourselves to iterated length-preserving transductions, more precisely, we shall study the the smallest family of transductions containing length-preserving rational transductions and closed under union, composition and iteration. There are two main reasons for this choice. First, one easily verifies that iterations of arbitrary rational transductions can be obtained by composition of arbitrary rational transductions with iterated length-preserving rational transductions. At reverse, arbitrary rational transductions can be achieved by composition of length-preserving rational transductions and iterations of faithful rational transductions. In this way the projection from A^* onto B^* with $B \subseteq A$ is equal to the composition of the iteration of the rational function which erases only the first occurrence of a letter of $A \setminus B$ with the length-preserving rational function which corresponds to the intersection with B^*.

We close the present introduction with a description of the structure of the article. After recalling some definitions and notations in Section 2, we give in Section 3 three examples of non-rational transductions which can be expressed with iteration.

The next section deals with several characterizations of the closure under union, composition and iteration of the class of length-preserving rational transductions. A such closure of a class of transductions is denoted "UCI-closure" for Union, Composition and Iteration. In Section 4 we show that the iteration of only one length-preserving rational transduction is needed to obtain all transductions of this class (Th. 4.3). As a consequence we see in Section 5 a result which states that the iterated transduction can be chosen in a restricted class of length-preserving rational transductions called "one-step transductions" – which correspond to finite rewriting systems with length-preserving rules (Prop. 5.2).

In Section 6, we prove that this class of transductions coincides with the class of "context-sensitive transductions" which are the natural extension of rational transductions using intersection with context-sensitive languages instead of regular languages (Th. 6.4).

The connection with transductions associated with recognizable picture languages [6] is given in Section 7 (Prop. 7.3). In Section 8, we show that the UCI-closure of the class of length-preserving rational functions also coincides with this class.

All these characterizations are summarized in the last section (Th. 9.1) and we show how these results allow us to obtain several closure properties.

A part of this work has been already published as extended abstract in [13].

2. PRELIMINARIES

We assume the reader to be familiar with basic formal language theory (see [2,4] 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 length of a word u is denoted by $|u|$, while $|u|_a$ denotes the number of occurrences of the letter a in u. The mirror image of a word u is denoted by $\tilde{u}$.

For a word $u \in \Sigma^*$, we denote by $\mathrm{alph}(u)$ the alphabet of u that is defined by: $\mathrm{alph}(u) = \{a \in \Sigma \mid |u|_a \neq 0\}$. The set of all factors of u is denoted by $\mathrm{Fac}(u)$ and is defined by:

$$\mathrm{Fac}(u) = \{w \in \Sigma^* \mid \exists v, v' \in \Sigma^* \text{ such that } vwv' = u\}.$$

The set of all factors of length n, for $n \geq 0$, of a word u is denoted by $\mathrm{Fac}_n(u)$ and corresponds to $\mathrm{Fac}(u) \cap \Sigma^n$. The set of all left factors of u is denoted by $\mathrm{LFac}(u)$ and is defined by:

$$\mathrm{LFac}(u) = \{w \in \Sigma^* \mid \exists v \in \Sigma^* \text{ such that } wv = u\}.$$

A language over Σ is a subset of Σ^*. The classes of regular, deterministic context-sensitive, context-sensitive, ε-free context-sensitive and recursively enumerable languages over Σ are denoted respectively by $\mathrm{Rec}(\Sigma^*)$, $\mathrm{CS_d}(\Sigma^*)$, $\mathrm{CS}(\Sigma^*)$, $\mathrm{CS}_\varepsilon(\Sigma^*)$ and r.e.(Σ^*).

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:

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

This definition is extended in a canonical way to languages. When we describe applications of several transductions, it is convenient to use the notation $u \xmapsto{\tau} v$ when $v \in u\tau$.

For a transduction τ, the *domain* of τ, denoted by Dom τ, is the set of all words which have an image by τ. The *set of images* of τ, denoted by Im τ, is the language of words which have an antecedent by τ.

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

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 a 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:

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

A transduction τ from X^* into Y^* is rational if and only if it is a rational part of the monoid $X^* \times Y^*$ (with the classical concatenation product which is defined by $(x,y).(x',y') = (xx',yy')$). It is 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-tuple (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 finite transition function from $Q \times X^*$ into the 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 $Q \times X^*$ in its entirety. For each state q, we force $\delta(q,\varepsilon)$ to contain (q,ε). If $\delta(q,x) \ni (q',y)$ and $\delta(q',x') \ni (q'',y')$ then $\delta(q,xx')$ 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 $(q_f,v) \in \delta(q_i,u)$. The transduction τ associated with T is defined by:

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

The next theorem presents well-known closure properties of the family 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.*

We say that a transduction τ is functional if for each word u in Dom τ, $u\tau$ contains exactly one word. When we deal with a function τ we will write $u\tau = v$ instead of $u\tau = \{v\}$. More precisions and definitions about functions will be given in Section 8.

A transduction τ is length-preserving (l.p. for short) 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 $\mathcal{T}$ and the class of length-preserving rational functions is denoted by $\mathcal{F}$.

Despite the fact that the class of rational transductions is not closed under intersection, the intersection of two length-preserving rational transductions is a length-preserving rational transduction.

A morphism φ is a rational function that satisfies the conditions $\varepsilon h = \varepsilon$ and $(uv)h = (uh)(vh)$ for every words u, v. Hence, the morphism φ is completely defined by the values $a\varphi$ of the letters $a \in \mathrm{alph}(\mathrm{Dom}\ \varphi)$. The morphism φ is called strictly alphabetic, or letter-to-letter, if for each $a \in \mathrm{alph}(\mathrm{Dom}\ \varphi)$ we have $|a\varphi| = 1$. The family of letter-to-letter morphisms is denoted by $\mathcal{M}$.

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.

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 $\mathcal{C}'$:

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

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

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

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

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

For instance $\mathcal{M}^{-1}(\cap \mathrm{Rec})$ denotes the class of transductions obtained by composition of an inverse letter-to-letter morphism and the intersection with a regular language.

3. ITERATIONS OF RATIONAL TRANSDUCTIONS

First, we give a formal definition of iterated transductions. The iteration is the natural extension of the Kleene operator to the semigroup of length-preserving transductions.

Definition 3.1. 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 family of iterated length-preserving rational transductions and $\mathcal{F}_+$ denotes the family of iterated length-preserving rational functions.

Example 1. *Mirror image.* Let us start with a simple example to explain the use of iterated length-preserving rational transductions. Let X be an arbitrary alphabet. We consider the function f which associates the mirror image with each word of X^*: $\forall w \in X^*$, $wf = \widetilde{w}$. Although this function is not rational, *i.e.* cannot be realized by a finite transducer, we show that f can be obtained by composition of rational functions and iterated length-preserving rational functions.

Let Σ be the alphabet $X \cup \dot{X}$ containing non-marked and marked letters of X. We define a rational function τ from Σ^* into Σ^*: for every word $w = auv$ with $a \in X$, $u \in X^*$ and $v \in \dot{X}^*$, we have $w\tau = u\dot{a}v$; the image of the empty word is the empty word, $\varepsilon\tau = \varepsilon$, and the transduction is undefined in other cases.

For instance, the successive applications of τ on a word over X of length seven are:

$$
\begin{aligned}
w &= a_1 a_2 a_3 a_4 a_5 a_6 a_7, \\
w\tau &= a_2 a_3 a_4 a_5 a_6 a_7 \dot{a}_1, \\
w\tau^2 &= a_3 a_4 a_5 a_6 a_7 \dot{a}_2 \dot{a}_1, \\
&\vdots \\
w\tau^7 &= \dot{a}_7 \dot{a}_6 \dot{a}_5 \dot{a}_4 \dot{a}_3 \dot{a}_2 \dot{a}_1, \\
w\tau^8 &= \emptyset.
\end{aligned}
$$

It is clear that for any word $w \in X^*$, the set $w\tau^+ (\cap \dot{X}^*)$ contains a single word which is the marked mirror of w. If we denote by φ the morphism from $\dot{X}^*$ into X^* which gives the unmarked image, we have $f = (\cap X^*)\tau^+ \varphi$.

Example 2.   *Natural integer division.*   Let us now consider the transduction which realizes the division of two positive integers. We use the binary representation of integers and we consider words over the alphabet $A = \{0, 1\}$. For a given word $u \in A^*$, we denote by $\hat{u}$ the integer represented by u – for convenience, we state that the most significant digit is on the right of the word, hence 1011 represents the integer 13, but the other case can be treated similarly. We also consider that the empty word represents 0.

For now, we define the transduction τ we want to build with l.p. rational transductions. The input words are over the alphabet $A \times A$ coding the two integers we want to treat:

$$\tau = \left\{ (u, v) \in (A \times A)^* \times A^* \mid |u| = |v| \wedge \hat{v} = \widehat{u\Pi_1} / \widehat{u\Pi_2} \right\}.$$

We use the naive algorithm which subtracts the divisor to the dividend until the dividend is smaller than the divisor and to count the number of subtractions.

This method could be used in VLSI to implement the integer division (see Fig. 1) but there exist faster methods.

It is well known that the subtraction of two integers can be realized by a finite transduction. In order to iterate subtractions we need to keep the divisor and to obtain the result of the subtraction in place of the dividend. Moreover we need to increase a counter we code in the third component of the input word. We consider the transduction σ defined by:

$$\sigma = \left\{ (u, v) \in (A \times A \times A)^* \times (A \times A \times A)^* \mid \begin{array}{c} \widehat{v\Pi_1} = \widehat{u\Pi_1} - \widehat{u\Pi_2} \\ \widehat{v\Pi_2} = \widehat{u\Pi_2} \\ \widehat{v\Pi_3} = \widehat{u\Pi_3} + 1 \end{array} \right\}.$$

The Figure 2 gives a finite transducer which realizes a simple subtraction. It is easy to construct a finite transducer for σ. Let us note that in the construction, if in a given word $u \in (A \times A \times A)^*$, the dividend $(\widehat{u\Pi_1})$ is strictly smaller than the divisor $(\widehat{u\Pi_2})$ then u has no image by σ.

To initialize the iteration, we use the rational transduction ρ which simply puts a third component which is the counter initialized to 0:

$$\rho = \{ (u, v) \in (A \times A)^* \times (A \times A \times A)^* \mid v\Pi_1 = u\Pi_1 \wedge v\Pi_2 = u\Pi_2 \wedge v\Pi_3 \in 0^* \}.$$

Now, we know how to initialize the computation and to run the subtraction. To achieve the computation, we just need to extract the result. We use the rational transduction φ which compares the two first components and outputs the third component when the dividend is smaller than the divisor:

$$\varphi = \left\{ (u, v) \in (A \times A \times A)^* \times A^* \mid \begin{array}{c} \widehat{u\Pi_1} < \widehat{u\Pi_2} \\ v = u\Pi_3 \end{array} \right\}.$$
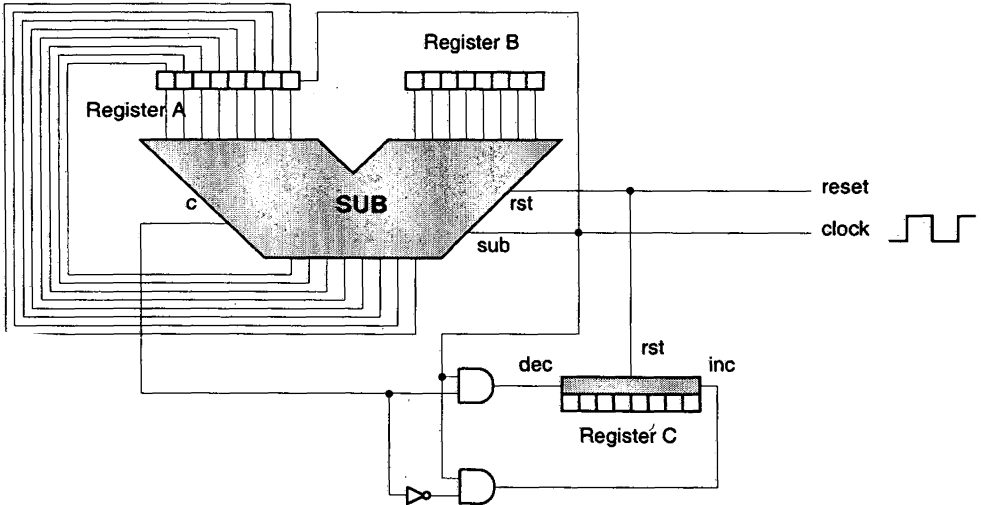
The dividend is stored in the register A while the divisor is stored in the register B. The sequencer initializes the computation by sending the **reset** signal which sets the counter, register C, to 0 and sets the carry, signal c of the component SUB, to zero.

Each clock cycle, the divisor B is subtracted to the dividend A, the result is stored in the register A and the counter is increased of one. These operations are repeated until the carry is set to 1. When the carry takes the value one, it means that the dividend was smaller than the divisor and that we have made an extra subtraction. Then it suffices to decrease the counter to obtain the quotient. Notice that it is "cheaper in time" to wait the carry than to compare A and B before each subtraction.

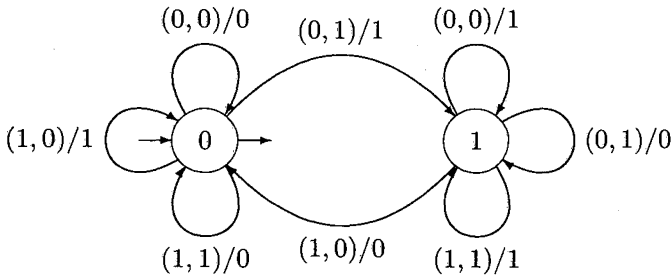FIGURE 1. A simple implementation of 8-bit integer division.

FIGURE 2. A transducer which realizes the subtraction.

The expected transduction τ can be expressed by using ρ, σ and φ:

$$\tau = \rho(\sigma + I_{A \times A \times A})^+ \varphi.$$

Since the iteration operator $^+$ means at least one application of the iterated transduction, we need to add the identity over $(A \times A \times A)^*$ to consider the case where the dividend is strictly smaller than the divisor.

Example 3. *Prime numbers.* For now, we give a way to generate the set of prime numbers by using iteration of rational transductions. In the previous example, we have seen how to realize the division. A similar method allows us to build a transduction for the modulo. We denote by τ_m the transduction corresponding to this operation:

$$\tau_m = \left\{ (u,v) \in (A \times A)^* \times (A \times A \times A)^* \mid \begin{array}{c} v\Pi_1 = u\Pi_1 \\ v\Pi_2 = u\Pi_2 \\ \widehat{v\Pi_3} = \widehat{u\Pi_1} \bmod \widehat{u\Pi_2} \end{array} \right\},$$

$$= \rho_m(\sigma_m + I_{A \times A \times A})^+ \varphi_m,$$

where ρ_m, σ_m and φ_m are the rational transductions defined by:

$$\rho_m = \left\{ (u,v) \in (A \times A)^* \times (A \times A \times A)^* \mid \begin{array}{c} v\Pi_1 = u\Pi_1 \\ v\Pi_2 = u\Pi_2 \\ v\Pi_3 = u\Pi_1 \end{array} \right\},$$

$$\sigma_m = \left\{ (u,v) \in ((A \times A \times A)^*)^2 \mid \begin{array}{c} v\Pi_1 = u\Pi_1 \\ v\Pi_2 = u\Pi_2 \\ \widehat{v\Pi_3} = \widehat{u\Pi_3} - \widehat{u\Pi_2} \end{array} \right\},$$

$$\varphi_m = \left\{ (u,v) \in (A \times A \times A)^* \times (A \times A \times A)^* \mid u = v \wedge \widehat{u\Pi_3} < \widehat{u\Pi_2} \right\}.$$

In order to test whether a given word u codes a prime number, we enumerate the words (of same length) which code the numbers belonging to $[2, \hat{u} - 1]$ and we check that $\hat{u}$ cannot be divided by these numbers.

We use τ_m and the following length-preserving rational transductions:

$$\rho = \{ (u,v) \in A^* \times (A \times A)^* \mid v\Pi_1 = u \wedge \widehat{v\Pi_2} = 2 \},$$

$$\sigma = \left\{ (u,v) \in (A \times A \times A)^* \times (A \times A)^* \mid \begin{array}{c} v\Pi_1 = u\Pi_1 \\ \widehat{v\Pi_2} = \widehat{u\Pi_2} + 1 \\ \widehat{u\Pi_3} \neq 0 \end{array} \right\},$$

$$\varphi = \{ (u,v) \in (A \times A)^* \times A^* \mid v = u\Pi_1 \wedge u\Pi_2 = u\Pi_1 \}.$$

The transduction τ is defined by:

$$
\begin{aligned}
\tau &= \{(u,u) \in A^* \times A^* \mid \hat{u} \text{ is prime}\}, \\
&= \rho(\tau_m\sigma)^+\varphi + \{(1,1)\}, \\
&= \rho(\rho_m(\sigma_m + I_{A \times A \times A})^+\varphi_m\sigma)^+\varphi + \{(1,1)\}.
\end{aligned}
$$

The couple $(1,1)$ has to be added since the word 1 has no image by ρ because we need at least two letters to code the integer 2. Notice that this restriction spares us to consider the case of the empty word.

The set of words over A which code prime numbers is then the image of A^* by the transduction τ.

4. THE UCI-CLOSURE OF LENGTH-PRESERVING RATIONAL TRANSDUCTIONS

As already mentioned above, it is well known that the class of rational transductions is closed under union and composition. It is obvious that the iteration of a length-preserving rational transduction is not necessarily a rational transduction. To be convinced, it suffices to consider the transitive closure of a rewriting system like semi-commutations – see also the examples of Section 3.

We use the notation UCI for Union, Composition and Iteration closure of a family of transduction.

Definition 4.1. The class UCI($\mathcal{T}$) is the smallest family of transductions which contains $\mathcal{T}$ and is closed under union, composition and iteration.

For the sequel, we give several representation theorems for this class of transductions. We first need a lemma which is useful in many proofs.

Lemma 4.2. *Let τ be a transduction of $\mathcal{T}\mathcal{T}_+\mathcal{T}$. We can find three length-preserving rational transductions σ_1, σ_2 and σ_3 such that $\tau = \sigma_1\sigma_2^+\sigma_3$ and* alph(Im σ_1) $\cap$alph(Dom σ_3) $= \emptyset$.

Proof. Let us consider three arbitrary transductions of $\mathcal{T}$ denoted by τ_1, τ_2 and τ_3 and the transduction $\tau = \tau_1\tau_2^+\tau_3$. We build σ_1, σ_2 and σ_3 which satisfy the conditions.

We rename the alphabets: let Σ be the alphabet alph(Dom τ_2) $\cup$alph(Im τ_2)$\cup$alph(Dom τ_3) and let $\overline{\Sigma}$ be the alphabet defined by $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$. Let h be the bijective morphism from Σ^* into $\overline{\Sigma}^*$ such that $ah = \overline{a}$ for every $a \in \Sigma$.

Let σ_2', σ_2'' be the two transductions defined by:

$$
\sigma_2' = \tau_2 h \text{ and } \sigma_2'' = h^{-1}\tau_2 h.
$$

Now, we take the next definitions:

$$\sigma_1 = \tau_1, \quad \sigma_2 = \sigma_2' + \sigma_2'', \text{ and } \quad \sigma_3 = h^{-1}\tau_3.$$

It is easy to see that $\tau = \sigma_1\sigma_2^+\sigma_3$. Since the alphabet alph(Dom σ_3) is a new alphabet (included in $\overline{\Sigma}$), the conditions are satisfied. $\square$

Theorem 4.3. *The family* UCI($\mathcal{T}$) *is equal to the family* $\mathcal{T}\mathcal{T}_+\mathcal{T}$. *This means that a transduction* τ *belongs to* UCI($\mathcal{T}$) *if and only if* $\tau = \sigma_1\sigma_2^+\sigma_3$ *for some* $\sigma_1, \sigma_2, \sigma_3 \in \mathcal{T}$.

Proof.   Since $\mathcal{T}\mathcal{T}_+\mathcal{T}$ is included in UCI($\mathcal{T}$), it is sufficient to show that $\mathcal{T}\mathcal{T}_+\mathcal{T}$ is closed under union, composition and iteration. Let σ_1, σ_2, σ_3, σ_1', σ_2' and σ_3' be transductions belonging to $\mathcal{T}$. We consider the transductions $\sigma_1\sigma_2^+\sigma_3$ and $\sigma_1'\sigma_2'^+\sigma_3'$.

Because we can rename the letters used "inside" the iterations, namely $\Sigma = \text{alph}(\text{Im }\sigma_1 \cup \text{Dom }\sigma_2 \cup \text{Im }\sigma_2 \cup \text{Dom }\sigma_3)$ and $\Sigma' = \text{alph}(\text{Im }\sigma_1' \cup \text{Dom }\sigma_2' \cup \text{Im }\sigma_2' \cup \text{Dom }\sigma_3')$, we can suppose that $\Sigma \cap \Sigma' = \emptyset$. Moreover, according to Lemma 4.2, we can suppose that alph(Im σ_1) and alph(Dom σ_3) are disjoint.

The case of the empty word may be problematic. Therefore, we suppose that $(\varepsilon, \varepsilon)$ belongs to a σ_i (respectively a σ_i') if and only if each σ_i (resp. each σ_i') contains this couple.

- The transduction $\sigma_1\sigma_2^+\sigma_3 + \sigma_1'\sigma_2'^+\sigma_3'$ belongs to $\mathcal{T}\mathcal{T}_+\mathcal{T}$:

$$\sigma_1\sigma_2^+\sigma_3 + \sigma_1'\sigma_2'^+\sigma_3' = (\sigma_1 + \sigma_1')(\sigma_2 + \sigma_2')^+(\sigma_3 + \sigma_3').$$

- The transduction $\sigma_1\sigma_2^+\sigma_3\sigma_1'\sigma_2'^+\sigma_3'$ belongs to $\mathcal{T}\mathcal{T}_+\mathcal{T}$:

$$\sigma_1\sigma_2^+\sigma_3\sigma_1'\sigma_2'^+\sigma_3' = \sigma_1(\sigma_2 + \sigma_3\sigma_1' + \sigma_2')^+\sigma_3'.$$

- The transduction $(\sigma_1\sigma_2^+\sigma_3)^+$ belongs to $\mathcal{T}\mathcal{T}_+\mathcal{T}$:

$$(\sigma_1\sigma_2^+\sigma_3)^+ = \sigma_1(\sigma_2 + \sigma_3\sigma_1)^+\sigma_3.$$

Hence UCI($\mathcal{T}$) is included in $\mathcal{T}\mathcal{T}_+\mathcal{T}$ and the result holds. $\square$

These result can be refined as follows:

Proposition 4.4. *Let* $\tau \subseteq X^* \times Y^*$ *be a transduction such that* $X \cap Y = \emptyset$. *The transduction* τ *belongs to* UCI($\mathcal{T}$) *if and only if* $\tau = (\cap X^*)\sigma^+(\cap Y^*)$ *for some* $\sigma \in \mathcal{T}$.

Proof. Let σ_1, σ_2 and σ_3 be three l.p. rational transductions such that $\tau = \sigma_1\sigma_2^+\sigma_3$. For the empty word, we suppose that $(\varepsilon, \varepsilon)$ belongs to a σ_i if and only if each σ_i

contains this couple. We consider the alphabets of the different domains and sets
of images:

$$X_1 = \text{alph}(\text{Dom } \sigma_1) \quad Y_1 = \text{alph}(\text{Im } \sigma_1)$$
$$X_2 = \text{alph}(\text{Dom } \sigma_2) \quad Y_2 = \text{alph}(\text{Im } \sigma_2)$$
$$X_3 = \text{alph}(\text{Dom } \sigma_3) \quad Y_3 = \text{alph}(\text{Im } \sigma_3).$$

We can suppose that $X_1 \subseteq X$ and that $Y_3 \subseteq Y$. Moreover, because we can rename
the different alphabets, we can suppose that:

$$X \cap (Y_1 \cup X_2 \cup Y_2 \cup Y_3) = \emptyset, \tag{1}$$
$$Y \cap (Y_1 \cup X_2 \cup Y_2 \cup Y_3) = \emptyset. \tag{2}$$

We consider the transduction $(\cap X^*)(\sigma_1 + \sigma_2 + \sigma_3)^+(\cap Y^*)$.

The equation (1) means that a word over X has no image by σ_2 or σ_3 and that
σ_1 cannot be applied after σ_1 or σ_2. On the other hand, the equation (2) signifies
that σ_2 and σ_3 cannot be applied after σ_3.

According to Lemma 4.2, we can suppose that $Y_1 \cap X_3 = \emptyset$. Hence σ_3 cannot
follow σ_1. Let us recall that by hypothesis, we have $X_1 \cap Y_3 = \emptyset$, this implies that
σ_1 cannot follow σ_3.

Hence we have $\sigma_1 \sigma_2^+ \sigma_3 = (\cap X^*)(\sigma_1 + \sigma_2 + \sigma_3)^+(\cap Y^*)$. $\square$

If the domain and the image of the transduction are not disjoint, this proof
does not work since we cannot prevent the application of σ_1 after σ_3 and, hence,
to iterate the complete transduction. However, by renaming the alphabets, we
deduce the next corollary from the previous one.

Corollary 4.5. *A transduction τ belongs to* $\text{UCI}(\mathcal{T})$ *if and only if there exist a
finite alphabet X, a rational transduction σ and a letter-to-letter morphism φ such
that $\tau = (\cap X^*)\sigma^+\varphi$.*

In the next section we show that the iterated transduction of Theorem 4.3 can
be replaced by a transduction in a restricted class of length-preserving rational
transductions named "one-step transductions".

5. ONE-STEP TRANSDUCTIONS AND CONTEXT-SENSITIVE LANGUAGES

The class of transductions we define is equivalent to the application of a rule of
a finite rewriting system which contains only length-preserving rules.

Definition 5.1. Let X be a finite alphabet and let P be a finite subset of $X^* \times X^*$
such that $(u, v) \in P \Rightarrow |u| = |v|$. The one-step transduction τ associated with
(X, P) is defined by:

$$\tau = \bigcup_{(u,v)\in P} \{(xuy, xvy) \mid x, y \in X^*\}.$$

The class of one-step transductions is denoted by $\mathcal{O}$.

It is obvious that $\mathcal{O}$ is properly included in $\mathcal{T}$. The following proposition states that we can use the iteration of a one-step transduction to characterize $\text{UCI}(\mathcal{T})$.

Proposition 5.2. *The family $\text{UCI}(\mathcal{T})$ is equal to the family $\mathcal{TO_+T}$. This means that a transduction τ belongs to $\text{UCI}(\mathcal{T})$ if and only if $\tau = \sigma_1 \sigma_2^+ \sigma_3$ for some $\sigma_1, \sigma_3 \in \mathcal{T}$ and $\sigma_2 \in \mathcal{O}$.*

Proof. It is clear that it suffices to prove that $\mathcal{T}_+$ is included in $\mathcal{TO_+T}$. Let $\tau \in \mathcal{T}$ be a l.p. rational transduction which does not contain the couple $(\varepsilon, \varepsilon)$. We construct three rational transductions σ_1, σ_2 and σ_3 (all length-preserving) such that $\tau^+ = \sigma_1 \sigma_2^+ \sigma_3$ and $\sigma_2 \in \mathcal{O}$.

Since the transduction τ is rational, it is realized by a transducer $T = (\Sigma, \Sigma', Q, \delta, \{q_0\}, F)$ with $\Sigma = \text{alph}(\text{Dom } \tau)$ and $\Sigma' = \text{alph}(\text{Im } \tau)$. Since τ is length-preserving, we can suppose that $\text{Dom } \delta \subseteq Q \times \Sigma$ and that for each $(q, a) \in Q \times \Sigma$, we have $\delta(q, a) \subseteq Q \times \Sigma'$ (see [4], p. 265).

- The transduction σ_1 marks the first and the last letters of each word of Σ^* and places the initial state q_0 on the first letter, whereas the remaining letters are unchanged. Formally, the transduction σ_1 is defined by:

$$\sigma_1 = \{(aub, (q_0, \dot{a})u\underline{b}) \mid a, b \in \Sigma, u \in \Sigma^*\} \cup \{(a, (q_0, \dot{\underline{a}})) \mid a \in \Sigma\}.$$

- The next transduction, σ_2 simulates, by iteration, several computations of the transducer T. It is the one-step transduction associated with the couple $(X \cup Q \times X, P)$ where X is the alphabet containing letters of Σ, Σ' and these letters with the begin or end marks (or both), and P is the finite relation defined by $P = P_1 \cup P_2 \cup P_3$:
  - the set P_1 initializes in a non-deterministic way a new run of the transducer:

  $$P_1 = \{(\dot{a}, (q_0, \dot{a})) \mid a \in \Sigma\} \cup \{(\dot{\underline{a}}, (q_0, \dot{\underline{a}})) \mid a \in \Sigma\},$$

  - the set P_2 simulates a transition of the transducer:

  $$P_2 = \bigcup_{(q',a') \in \delta(q,a)} \left\{ \begin{array}{l} ((q,a)b, a'(q',b)), ((q,\dot{a})b, \dot{a}'(q',b)), \\ ((q,a)\underline{b}, \dot{a}'(q',\underline{b})), ((q,\dot{a})\underline{b}, \dot{a}'(q',\underline{b})) \end{array} \right\},$$

  - the set P_3 deletes the state on the last letter when this state leads to a final state by reading this last letter:

  $$P_3 = \bigcup_{\substack{(q',a') \in \delta(q,a) \\ q' \in F}} \{((q,\underline{a}), \underline{a}'), ((q,\dot{\underline{a}}), \underline{a}')\}.$$

- The last transduction σ_3 verifies that there is no state in the word and deletes the marks:

$$\sigma_3 = \{(\grave{a}u\underline{b}, aub) \mid a, b \in \Sigma', u \in \Sigma'^*\} \cup \{(\underline{\grave{a}}, a) \mid a \in \Sigma'\}.$$

Namely, applications of $\sigma_1\sigma_2^+\sigma_3$ look like the following:

$$
\begin{array}{lll}
a_0a_1a_2a_3 & \underset{\sigma_1}{\longmapsto} & (q_0, \dot{a}_0)a_1a_2\underline{a_3} \\[2mm]
& \underset{\sigma_2\ (P_2)}{\longmapsto} & \dot{a}_0'(q_1, a_1)a_2\underline{a_3} & \text{if } (q_1, a_0') \in \delta(q_0, a_0) \\[2mm]
& \underset{\sigma_2\ (P_2)}{\longmapsto} & \dot{a}_0'a_1'(q_2, a_2)\underline{a_3} & \text{if } (q_2, a_1') \in \delta(q_1, a_1) \\[2mm]
& \underset{\sigma_2\ (P_1)}{\longmapsto} & (q_0, \dot{a}_0')a_1'(q_2, a_2)\underline{a_3} \\[2mm]
& \underset{\sigma_2\ (P_2)}{\longmapsto} & \dot{a}_0''(q_1', a_1')(q_2, a_2)\underline{a_3} & \text{if } (q_1', a_0'') \in \delta(q_0, a_0') \\[2mm]
& \underset{\sigma_2\ (P_2)}{\longmapsto} & \dot{a}_0''(q_1', a_1')a_2'(q_3, \underline{a_3}) & \text{if } (q_3, a_2') \in \delta(q_2, a_2) \\[2mm]
& \underset{\sigma_2\ (P_2)}{\longmapsto} & \dot{a}_0''a_1''(q_2', a_2')(q_3, \underline{a_3}) & \text{if } (q_2', a_1'') \in \delta(q_1', a_1') \\[2mm]
& \underset{\sigma_2\ (P_3)}{\longmapsto} & \dot{a}_0''a_1''(q_2', a_2')\underline{a_3'} & \text{if } (q, a_3') \in \delta(q_3, a_3) \\[1mm]
& & & \text{for some } q \in F \\[2mm]
& \underset{\sigma_2\ (P_2)}{\longmapsto} & \dot{a}_0''a_1''a_2''(q_3', \underline{a_3}) & \text{if } (q_3', a_2'') \in \delta(q_2', a_2') \\[2mm]
& \underset{\sigma_2\ (P_3)}{\longmapsto} & \dot{a}_0''a_1''a_2''\underline{a_3''} & \text{if } (q', a_3'') \in \delta(q_3', a_3') \\[1mm]
& & & \text{for some } q' \in F \\[2mm]
& \underset{\sigma_3}{\longmapsto} & a_0''a_1''a_2''a_3''.
\end{array}
$$

This derivation shows how the word $a_0a_1a_2a_3$ may be transformed into $a_0''a_1''a_2''a_3''$ where

$$a_0a_1a_2a_3 \underset{\tau}{\longmapsto} a_0'a_1'a_2'a_3' \underset{\tau}{\longmapsto} a_0''a_1''a_2''a_3''.$$

Notice that these two derivations are not separated, *i.e.* the transduction on $a_0'a_1'a_2'a_3'$ begins at the fourth step while the first one is not finished.

The reader shall easily verify by himself that $\tau^+ = \sigma_1\sigma_2^+\sigma_3$.

Nevertheless, if $(\varepsilon, \varepsilon)$ belongs to τ, we use the same construction but we add the couple $(\varepsilon, \varepsilon)$ to σ_1, P and σ_3. $\qquad\qquad\square$

In order to prepare the next results, we need to consider context-sensitive languages which are closely related with $\mathcal{O}_+$.

Lemma 5.3. *Let $A \in \mathrm{CS}(\Sigma^*)$ be a context-sensitive language over Σ. There exist a regular language R and a one-step transduction τ such that $A = R\tau^+(\cap\Sigma^*)$.*

Proof. First, we consider a language A over Σ which does not contain the empty word. Since A is a context-sensitive language, it is generated by a length-increasing grammar $G = (\Sigma, V, P, S)$ where V is the set of variables, P the set of productions

included in $(\Sigma \cup V)^* \times (\Sigma \cup V)^*$ such that for every $(u, v) \in P$ we have $|v| \geq |u|$, and S is the axiom.

We build a one-step transduction τ associated with $(\Sigma \cup V \cup \{\$\}, Q)$, where $\$$ is a new letter which does not belong to Σ or V, and Q is the set of rewriting rules defined by:

$$Q = \left\{ (u\$^{|v|-|u|}, v) \mid (u, v) \in P \right\} \cup \left\{ (a\$, \$a) \mid a \in \Sigma \cup V \right\}.$$

It is easy to see that $A = (S\$^*)\tau^+(\cap \Sigma^*)$.

Let A be a context-sensitive language which contains the empty word. We use the same construction for $A \setminus \{\varepsilon\}$ and we add the couple $(\varepsilon, \varepsilon)$ to Q. Then, we get $A = (\{\varepsilon\} \cup S\$^*)\tau^+(\cap \Sigma^*)$. $\qquad\qquad\square$

Lemma 5.4. *The class of context-sensitive languages is closed under transductions of* $\mathrm{UCI}(\mathcal{T})$.

Proof. Clearly, it suffices to show that the class of ε-free context-sensitive languages is closed under iterated one-step transduction (Prop. 5.2). Let $A \in \mathrm{CS}_\varepsilon(\Sigma^*)$ be an ε-free context-sensitive language and τ a one-step transduction associated with (X, Q).

Since context-sensitive languages are closed under length-preserving rational transductions, the language $B = A\tau$ is an ε-free context-sensitive language. This language is generated by a grammar $G = (X, V, P, S)$ in Kuroda normal form [10]:

$$\forall u \to v \in P \quad (u, v) \in (V \times X) \cup (V \times V^2) \cup (V^2 \times V^2).$$

We construct a grammar $G' = (X, V, P', S)$ where the set of productions is P increased by the rules of Q: $P' = P \cup Q$. Since each rules in P has a left part in V^* and since each rules in Q concerns only words over X, it is easy to see that $\mathcal{L}(G')$, the language generated by G', is equal to $A\tau^+$. Moreover, G' is a length-increasing grammar, so $A\tau^+$ is context-sensitive. $\qquad\qquad\square$

From these two lemmas, we can deduce the following result.

Corollary 5.5. *Let* $A \subseteq \Sigma^*$ *be a language over* Σ. *It is a context-sensitive language if and only if there exist a finite alphabet* X *and a length-preserving rational transduction* τ *such that* $A = (X^*)\tau^+(\cap \Sigma^*)$.

Proof. Let $A \in \mathrm{CS}(\Sigma^*)$ be a context-sensitive language. By Lemma 5.3 we know that $A = R\sigma^+(\cap \Sigma^*)$ where R is a regular language over an alphabet X and τ is a one-step transduction. It is easy to construct a length-preserving rational transduction σ' from $\{a\}^*$ into X^* such that $R = \{a\}^*\sigma'$. Hence, we have $A = \{a\}^*\sigma'\sigma^+(\cap \Sigma^*)$. According to Proposition 4.4, we know that $\sigma'\sigma^+ = (\cap\{a\}^*)\tau^+(\cap \Sigma^*)$ for some $\tau \in \mathcal{T}$. The equality $A = \{a\}^*\tau^+(\cap \Sigma^*)$ then clearly holds.

Let τ be a length-preserving rational transduction and let X be a finite alphabet. By Proposition 5.2 we know that $\tau^+ = \sigma_1 \sigma_2^+ \sigma_3$ where σ_1 and σ_3 are two length-preserving rational transductions and σ_2 a one-step transduction. Let us denote A the language $X^* \tau^+ (\cap \Sigma^*)$. We have $A = X^* \sigma_1 \sigma_2^+ \sigma_3 (\cap \Sigma^*)$. The language $X^* \sigma_1$ is clearly regular. By Lemma 5.4, we know that $X^* \sigma_1 \sigma_2^+$ is context-sensitive. Since the class of context-sensitive languages is closed under $\mathcal{T}$ and under intersection with regular sets, A is also context-sensitive.   □

The characterization of context-sensitive languages by one-step transductions allows us to show the next result which concerns context-sensitive transductions.

6. CONTEXT-SENSITIVE TRANSDUCTIONS

The purpose of this section is to establish a similar result for transductions of UCI($\mathcal{T}$) to the one stated by the Nivat's theorem for rational transductions [18]. This theorem can be stated as follow.

Theorem 6.1 (Nivat [18]). *A transduction τ is rational if and only if there exist two morphisms φ and ψ and a regular language A such that:*

$$\tau = \varphi^{-1}(\cap A)\psi.$$

Moreover, we can always find two morphisms φ and ψ and a language A which satisfy the following property:

$$\forall a \in \text{alph}(A) \quad a\varphi.a\psi \neq \varepsilon. \tag{P}$$

We extend this characterization to define "context-sensitive transductions".

Definition 6.2. Let τ be a transduction. The transduction τ is a context-sensitive transduction if and only if there exist two morphisms φ, ψ and a context-sensitive language A which satisfy the property (P) and such that $\tau = \varphi^{-1}(\cap A)\psi$.

We force the morphisms to verify the property (P) because without this restriction we obtain the class of "recursively enumerable transductions" which is the class of transductions (computations) which can be realized by a Turing Machine.

Indeed, let $B \in$ r.e. be a recursively enumerable language. We know that B is the image of a context-sensitive language A by a morphism π (which can be erasing) [14]. Hence, the intersection with B can be realized with the intersection with A: $(\cap B) = \pi^{-1}(\cap A)\pi$. Therefore, without the restriction we can obtain all "computable" transductions. But, it seems to be preferable to restrict our study to a less powerful class.

Here, we are interested in the family of length-preserving context-sensitive transductions which is denoted by $\mathcal{T}_{\text{CS}}$.

In the case of length-preserving rational transductions, we can assure that h and g are two letter-to-letter morphisms (it is a result attributed to Eilenberg and Schützenberger presented in [4] – see also [15]). This result means that

$\mathcal{T} = \mathcal{M}^{-1}(\cap\text{Rec})\mathcal{M}$. We show that if a context-sensitive transduction is length-preserving, we can use letter-to-letter morphisms in its definition. In other words, we have $\mathcal{T}_{CS} = \mathcal{M}^{-1}(\cap CS)\mathcal{M}$.

Lemma 6.3. *Let* $\tau \in \mathcal{T}_{CS}$ *be a length-preserving context-sensitive transduction. There exist two letter-to-letter morphisms* φ, ψ *and a context-sensitive language* A *such that* $\tau = \varphi^{-1}(\cap A)\psi$.

Proof. Let $\tau \in \mathcal{T}_{CS}$ be a l.p. transduction defined by $\tau = \sigma^{-1}(\cap B)\sigma'$ with σ, σ' two morphisms, and B a context-sensitive language with the following property:

$$\forall a \in \text{alph}(B) \quad a\sigma.a\sigma' \neq \varepsilon.$$

We can suppose that $\text{alph}(B) = \text{alph}(\text{Dom } \sigma) = \text{alph}(\text{Dom } \sigma')$. We consider the following alphabets:

$$X = \text{alph}(\text{Im } \sigma), \quad Y = \text{alph}(\text{Im } \sigma'), \quad \Sigma = \text{alph}(\text{Dom } \sigma').$$

We distinguish two cases:

- If $X \cap Y = \emptyset$. We consider the morphism h from Σ^* into $(X \cup Y)^*$ defined by $ah = a\sigma.a\sigma'$. Notice that h is non-erasing. We denote by θ the partial commutation where the letters of X commute with the letters of Y:

$$\theta = \{(u, v) \mid u\Pi_X = v\Pi_X \wedge u\Pi_Y = v\Pi_Y\}.$$

  Let g be the morphism from $(X \times Y)^*$ into $(X \cup Y)^*$ such that $(x, y)g = xy$. It is easy to see that

$$\tau = \Pi_1^{-1}(\cap A)\Pi_2 \quad \text{with } A = Bh\theta g^{-1}.$$

  Since the class of context-sensitive languages is closed under non-erasing morphisms, inverse morphisms and partial commutations (it is a corollary of Lem. 5.4 – see also [3, 22]), the language A is also context-sensitive and the property holds.
- If X and Y are not disjoint. It suffices to rename the alphabet Y with a bijective letter-to-letter morphism ρ and to apply the previous construction for $\sigma^{-1}(\cap B)\sigma'\rho$.

$\square$

The main result of this section is the following equality.

Theorem 6.4. *The classes* $\text{UCI}(\mathcal{T})$ *and* $\mathcal{T}_{CS}$ *coincide.*

In order to obtain this result, we prove separately each inclusion.

Lemma 6.5. $\mathcal{T}_{CS} \subseteq \text{UCI}(\mathcal{T})$.

Proof. It is clear that it suffices to show that $(\cap \mathrm{CS}) \subset \mathcal{T}\mathcal{O}_+\mathcal{T}$. Hence, for a given context-sensitive language $A \in \mathrm{CS}(\Sigma^*)$, we build three l.p. rational transductions σ_1, σ_2 and σ_3, where σ_2 is one-step, such that $(\cap A) = \sigma_1\sigma_2^+\sigma_3$.

According to Lemma 5.3, we have $A = R\tau^+(\cap \Sigma^*)$ where $R \in \mathrm{Rec}(X^*)$ is a regular language and τ is a one-step transduction associated with the couple (X, P).

The role of the three transductions σ_1, σ_2 and σ_3 is to verify that a word $u \in \Sigma^+$ can belong to $w\tau^+$ for a word $w \in X^*$. These transductions are defined as follow:

- the first transduction σ_1 generates from u every words of R of the same length as u. It also memorizes the word u:

$$\sigma_1 = \{(u,v) \in \Sigma^* \times (\Sigma \times X)^*) \mid v\Pi_1 = u \wedge v\Pi_2 \in R\}.$$

  We can also define σ_1 as $\Pi_1^{-1}(\cap(R\Pi_2^{-1}))$, hence it is clear that σ_1 is rational,

- the transduction σ_2 applies τ to the second component of words of $(\Sigma \times X)^*$:

$$\sigma_2 = \{(u,v) \in ((\Sigma \times X)^*)^2 \mid u\Pi_1 = v\Pi_1 \wedge v\Pi_2 \in u\Pi_2\tau\}.$$

It is easy to see that σ_2 is the one-step transduction associated with $(\Sigma \times X, P')$ where:

$$P' = \{(x,y) \mid x\Pi_1 = y\Pi_1 \wedge (x\Pi_2, y\Pi_2) \in P\},$$

- the last transduction σ_3 verifies that the two components are identical and over Σ:

$$\sigma_3 = \{(u,v) \in (\Sigma \times \Sigma)^* \times \Sigma^* \mid u\Pi_1 = u\Pi_2 = v\}.$$

This transduction is obviously rational.

A word $u \in \Sigma^*$ belongs to $\mathrm{Dom}\ \sigma_1\sigma_2^+\sigma_3$ if and only if there exists a word $w \in R$ such that $u \in w\tau^+$; in this case we have $u\sigma_1\sigma_2^+\sigma_3 = \{u\}$. Thus, $\sigma_1\sigma_2^+\sigma_3$ is equal to $(\cap A)$. $\square$

The converse inclusion is deduced from the following lemma:

Lemma 6.6. *Let $\tau \subseteq X^* \times Y^*$ be a transduction belonging to $\mathrm{UCI}(\mathcal{T})$. The language*

$$A = \{u \in (X \times Y)^* \mid u\Pi_2 \in u\Pi_1\tau\}$$

is a context-sensitive language and $\tau = \Pi_1^{-1}(\cap A)\Pi_2$.

Proof. We use Proposition 5.2 and we show that for any transduction σ in $\mathcal{O}$, the iterated transduction σ^+ belongs to $\mathcal{T}_{\mathrm{CS}}$. Let (X, P) be the couple alphabet-rule set associated with σ.

Let L be the language over $(X \times X)$ defined by:

$$L = \{u \in (X \times X)^* \mid u\Pi_2 \in u\Pi_1\sigma^+\}.$$

We want to prove that this language is context-sensitive. Let σ' be the one-step transduction associated with $(X \times X, P')$ where the set P' is equivalent to the application of a rule of P on the second component of a word over $(X \times X)$:

$$P' = \{(x,y) \in (X \times X)^* \mid x\Pi_1 = y\Pi_1 \wedge (x\Pi_2, y\Pi_2) \in P\}.$$

We can see that:

$$L = L'\sigma'^+ \text{ with } L' = \{u \in (X \times X)^* \mid u\Pi_1 = u\Pi_2\}.$$

Since the language L' is regular, by Lemma 5.4 we know that L is context-sensitive. By the definition of L, we can easily deduce that $\sigma^+ = \Pi_1^{-1}(\cap L)\Pi_2$. $\qquad\square$

7. RECOGNIZABLE PICTURE LANGUAGES AND TRANSDUCTIONS

In this section, we consider recognizable picture languages in the terminology of Giammarresi and Restivo (see [6] for complete definitions). A picture over an alphabet Σ is a two-dimensional rectangular array of letters of Σ. We denote the set of all pictures over Σ by Σ^{**}. The set of all pictures over Σ of n rows and m columns $(n, m \neq 0)$ is denoted by $\Sigma^{n,m}$.

Let p be a picture. We denote by $p(i,j)$ the letter which occurs at the ith line (from top) and jth column (from left). The projection by π of p is a picture with the same size where the letter at a coordinate (x, y) is the image by π of the letter $p(x, y)$. The set lines(p) (respectively the set columns(p)) represents the set of all rows of p (taken as a string) (resp. all columns of p taken as string from top to bottom).

According to [6] and [11], we can define recognizable picture languages as follows:

Definition 7.1. Let L be a picture language over Σ. The picture language L is recognizable if and only if there exist two recognizable string languages H and V over an alphabet X and a mapping π from X into Σ such that:

$$L = \{f \in \Sigma^{**} \mid \exists p \in \pi^{-1}(f) \text{ s.t. lines}(p) \subseteq H \wedge \text{columns}(p) \subseteq V\}.$$

Because of the strong links between recognizable and local string sets, it is easy to see that the recognizable string languages can always be replaced by local string languages.

For instance, the set of all squares over the alphabet $\Sigma = \{a\}$, denoted by L, is recognizable since it can be obtained by projection of the language K of all pictures over $X = \{0, 1\}$ which have exactly one occurence of the letter 1 on each row and on each column. The language K can be defined by using recognizable string languages:

$$K = \{p \in X^{**} \mid \text{lines}(p) \subseteq 0^*10^* \wedge \text{columns}(p) \subseteq 0^*10^*\}.$$

The pictures of K look like the following:

0	1	0
1	0	0
0	0	1

0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0

The language L is the projection of K by the projection $\pi : X \to \Sigma$ such that $\pi(0) = \pi(1) = a$.

We denote by $\mathrm{fr}_\top(p)$ and $\mathrm{fr}_\perp(p)$ the words which appear at the top of p and at the bottom of p, respectively.

With a picture language, we associate a transduction which is defined as follows:

Definition 7.2. Let L be a picture language. The transduction associated with L is denoted by τ_L and is defined by:

$$\tau_L = \{(\mathrm{fr}_\top(p), \mathrm{fr}_\perp(p)) \mid p \in L\}.$$

The class of transductions associated with recognizable picture languages is denoted by $\mathcal{T}_{\mathrm{Rec}(PL)}$.

The idea is to consider pictures as computations over the words which occur on the first lines. We introduce this new family of transductions because we have the next result:

Proposition 7.3. *The class $\mathcal{T}_{\mathrm{Rec}(PL)}$ coincides with the class of transductions of* $\mathrm{UCI}(\mathcal{T})$ *which do not contain the couple* $(\varepsilon, \varepsilon)$.

This proposition is deduced from Lemmas 7.5 and 7.6 but we first need a technical lemma:

Lemma 7.4. *Let L be a recognizable picture language. There exists a recognizable picture language K containing no picture of height one such that $\tau_L = \tau_K$.*

Proof. The language L is defined with two recognizable string languages H and V over an alphabet X and a projection π according to Definition 7.1. Let V' be the recognizable string language defined by:

$$V' = XV \cap (\{aa \mid a \in X\}X^*).$$

It is clear that the recognizable picture language

$$K = \{f \in \Sigma^{**} \mid \exists p \in \pi^{-1}(f) \text{ s.t. lines}(p) \subseteq H \wedge \text{columns}(p) \subseteq V'\}$$

verifies $\tau_K = \tau_L$.   □

Lemma 7.5. *The class $\mathcal{T}_{\mathrm{Rec}(PL)}$ is included in* $\mathrm{UCI}(\mathcal{T})$.

Proof. Let us consider a recognizable picture language L over an alphabet Σ. According to Lemma 7.4, we can suppose that L does not contain pictures of height one. The language L can be defined with two local string languages H and V over an alphabet A and a projection π from A into Σ:

$$L = \{f \in \Sigma^{**} \mid \exists p \in \pi^{-1}(f) \text{ s.t. lines}(p) \subseteq H \wedge \text{columns}(p) \subseteq V\}.$$

First we consider the language

$$L' = \{p \in A^{**} \mid \text{lines}(p) \subseteq H \wedge \text{columns}(p) \subseteq V\}$$

and we show that $\tau_{L'}$ belongs to UCI($\mathcal{T}$). The idea is to define two alphabets X and Y included in Σ and a length-preserving rational transduction τ in order to define a picture language K:

$$\forall p \in A^{n,m} \quad p \in K \iff \begin{cases} \text{fr}_\top(p) \in X^*, \\ \forall 1 < i \leq n \\ \quad p(i,1) \ldots p(i,m) \in (p(i-1,1) \ldots p(i-1,m))\tau, \\ \text{fr}_\perp(p) \in Y^*. \end{cases}$$

This property means that $\tau_K = (\cap X^*)\tau^+(\cap Y^*)$ and then that τ_K belongs to UCI($\mathcal{T}$). The next step will be to show that $K = L'$.

Because L does not contain pictures with less than two rows, we can suppose that the language V does not contain words of length less than two. Then V, which is local, can be defined by three sets $B \subseteq A$, $F \subseteq A^2$ and $E \subseteq A$:

$$V = BA^* \cap \{w \in A^* \mid \text{Fac}_2(w) \subseteq F\} \cap A^*E.$$

The set B corresponds to the set of authorized beginnings, E to the set of authorized endings and F to the set of authorized factors of length two. Moreover, since the words of V are longer than two, we have $B \cap E = \emptyset$.

We define $X = B$, $Y = E$ and the transduction τ is defined by:

$$\tau = \left\{ (a_0 \ldots a_n, b_0 \ldots b_n) \mid \begin{array}{l} a_0 \ldots a_n, b_0 \ldots b_n \in H \\ \forall 1 \leq i \leq n \quad a_i b_i \in F \end{array} \right\}.$$

This transduction is rational since it can be defined by a bimorphism (see Nivat's theorem recalled as Th. 6.1). We denote by R the regular string language defined by:

$$R = \left\{ (a_0, b_0) \ldots (a_n, b_n) \in (A \times A)^* \mid \begin{array}{l} a_0 \ldots a_n, b_0 \ldots b_n \in H \\ \forall 1 \leq i \leq n \quad a_i b_i \in F \end{array} \right\}.$$

It is clear that $\tau = \Pi_1^{-1}(\cap R)\Pi_2$.

For now we show that $K = L'$. Let $p \in A^{n,m}$ be a picture belonging to K. Since X and Y are disjoint ($B \cap E = \emptyset$), we know that $n \geq 2$. By definition of K, we have

$$p(1,1)\ldots p(1,m) \quad \in \quad X^*, \tag{3}$$
$$\forall 1 < i \leq n \quad p(i,1)\ldots p(i,m) \quad \in \quad (p(i-1,1)\ldots p(i-1,m))\tau, \tag{4}$$
$$p(n,1)\ldots p(n,m) \quad \in \quad Y^*. \tag{5}$$

From the equation (4), we can deduce that

$$\forall 1 \leq i \leq n \quad p(i,1)\ldots p(i,m) \quad \in \quad H, \tag{6}$$
$$\forall 1 < i \leq n \forall 1 \leq j \leq m \quad p(i-1,j)p(i,j) \quad \in \quad D. \tag{7}$$

Hence, by equations (3, 5) and (7), we obtain:

$$\forall 1 \leq i \leq m \quad p(1,i)\ldots p(n,i) \in V.$$

Combined with the equation (6), it means that $p \in L'$. This means that $L' \subseteq K$ by construction of K.

Then we know that $\tau_{L'} = (\cap X^*)\tau^+(\cap Y^*) \in \mathrm{UCI}(\mathcal{T})$. The transduction associated with L is now easily defined in terms of composition and iteration of l.p. rational transductions: $\tau_L = \pi^{-1}\tau_{L'}\pi$. So τ_L belongs to $\mathrm{UCI}(\mathcal{T})$. $\qquad\square$

Lemma 7.6. *Let $\tau \in \mathrm{UCI}(\mathcal{T})$ be a transduction which does not contain the couple $(\varepsilon, \varepsilon)$. It also belongs to $\mathcal{T}_{\mathrm{Rec}(PL)}$.*

Proof. Let τ be a transduction of $\mathrm{UCI}(\mathcal{T})$ which does not contain the couple $(\varepsilon, \varepsilon)$. According to Corollary 4.5, we know that $\tau = (\cap X^*)\sigma^+\varphi$ where X is a finite alphabet, φ a letter-to-letter morphism and τ a length-preserving rational transduction. We denote by Y and Z respectively the alphabets of the domain of φ and of the image of φ. Let Σ be the alphabet used in the iteration: $\Sigma = \mathrm{alph}(\mathrm{Dom}\,\sigma \cup \mathrm{Im}\,\sigma)$. We can suppose that X and Y are included in Σ.

We construct a recognizable picture language K such that $\tau_K = \tau$. We use the alphabet $A = (\Sigma \times \Sigma)$. First, we define a recognizable string language L_σ which corresponds to one application of σ:

$$L_\sigma = \{w \in A^* \mid w\Pi_2 \in w\Pi_1\sigma\}.$$

Let us consider a picture of two rows respectively occupied by the words u and v of L_σ. This picture looks like the following:

$u \rightarrow$	(x_0, y_0)	(x_1, y_1)	$\cdots$	(x_n, y_n)
$v \rightarrow$	(x'_0, y'_0)	(x'_1, y'_1)	$\cdots$	(x'_n, y'_n)

If we impose for every $0 \leq i \leq n$ the equality $y_i = x'_i$, we have $v\Pi_2 \in u\Pi_1\sigma^2$. At reverse, for each couple of words w, w' in Σ^* such that $w' \in w\sigma^2$, there exists a picture with $u\Pi_1 = w$ and $v\Pi_2 = w'$. It is the main idea of the construction. Let us consider the string language C' over A defined by:

$$C' = \{w \in A^* \mid \forall (x,y)(x',y') \in \text{Fac}_2(w) \quad y = x'\},$$

and the recognizable picture language K' defined by:

$$K' = \{p \in A^{**} \mid \text{lines}(p) \subseteq L_\sigma \wedge \text{columns}(p) \subseteq C'\}.$$

It is easy to see that $\Pi_1^{-1}\tau_{K'}\Pi_2 = \sigma^+$. For a given picture p of K' of height k we have $\text{fr}_\perp(p)\Pi_2 \in \text{fr}_\top(p)\Pi_1\sigma^k$. Conversely, for each couple of words w, w' in Σ^* such that $w' \in w\sigma^k$, there exists a picture p in K' such that $\text{fr}_\top(p)\Pi_1 = w$ and $\text{fr}_\perp(p)\Pi_2 = w'$.

To conclude the proof, we just need to assure that the first row contains only words over X and that the last row contains words over Y and to apply the morphism φ. Let L and C be the recognizable string languages defined by:

$$
\begin{aligned}
L &= X^* \cup L_\sigma \cup Z^*, \\
C &= XC'Y \cap (\{x(x,y) \mid x \in X, y \in \Sigma\}A^*) \\
&\quad \cap (A^*\{(x,y)(y\varphi) \mid x \in \Sigma, y \in Y\}).
\end{aligned}
$$

The language K is defined from L and C:

$$K = \{p \in A^{**} \mid \text{lines}(p) \subseteq L \wedge \text{columns}(p) \subseteq C\}.$$

The result clearly holds since we have $\tau_K = \tau$. $\qquad\qquad\square$

Using Corollary 5.5 and Proposition 7.3, we easily get a result concerning recognizable picture language theory [12].

Corollary 7.7 (Latteux and Simplot [12]). *The family of frontiers of recognizable picture languages is exactly the family of ε-free context-sensitive languages.*

In this corollary, "frontiers" mean the "bottom lines" of the pictures.

8. THE UCI-CLOSURE OF LENGTH-PRESERVING RATIONAL FUNCTIONS

We now turn to iterations of functions. Indeed, it is natural to wonder if we obtain another class if we take length-preserving rational functions instead of rational transductions. So we introduce the family $\text{UCI}(\mathcal{F})$ and we show that it coincides with $\text{UCI}(\mathcal{T})$.

To use functions in place of transductions is a way to reduce the non-determinism. When we use non-functional transductions, during the computation we have to

choose one word in the possible images of the current word, but with functions we have no choice.

Although, there is non-determinism which occurs in the application of the functions. That why we also consider sub-sequential functions and ts-sequential functions.

In order to define these two families of functions, we need to introduce deterministic finite transducers. There exist several notion of determinism for transducer, here we use a natural definition which means that at every step of the computation we have no choice to do.

Let $T = (X, Y, Q, \delta, I, F)$ be a finite transducer. The transducer T is deterministic if

> I is a singleton
> δ is functional
> $\forall(q, u) \in \text{Dom } \delta \quad \forall v \in \text{LFac}(u) \setminus \{u\} \quad \delta(q, v) = \emptyset.$

A rational function is sub-sequential if it can be realized by a deterministic transducer with an output function associated with final states – the word given by this function is concatenated at the end of the output word (see [2] for more precisions). A rational function which is realized by a deterministic finite transducer is a called ts-sequential for sequential with terminal states.

The class of all length-preserving sub-sequential functions is denoted by $\mathcal{S}$, the class of all length-preserving ts-sequential functions is denoted by $\mathcal{S}_{\text{ts}}$ and we have $\mathcal{S}_{\text{ts}} \subset \mathcal{S}$.

We can notice that for every l.p. ts-sequential function (and then l.p. sub-sequential functions) we cannot construct a letter-to-letter deterministic finite transducer (*i.e.* every transition is labeled with a letter as input and a letter as output) – for instance, consider the function which gives 00 for 10 and 11 for 11.

The class of letter-to-letter ts-sequential functions, denoted by $\mathcal{S}_{\text{ts},\text{ll}}$, is then the simplest class. In this class, when we read a letter we can output a letter with no choice. We have the following obvious inclusions:

$$\mathcal{M} \subset \mathcal{S}_{\text{ts},\text{ll}} \subset \mathcal{S}_{\text{ts}} \subset \mathcal{S} \subset \mathcal{F} \subset \mathcal{T}.$$

Our results concern the class $\mathcal{S}_{\text{ts},\text{ll}}$ and are naturally extended to the other classes that contain this class and which are more classic.

The aim is then to show that the UCI-closure of $\mathcal{S}_{\text{ts},\text{ll}}$ is equal to $\text{UCI}(\mathcal{T})$. It suffices to show that $\text{Rat}(\mathcal{S}_{\text{ts},\text{ll}})$ contains all l.p. rational transductions. The first step is to show that inverse letter-to-letter morphisms are in $\text{Rat}(\mathcal{S}_{\text{ts},\text{ll}})$.

Lemma 8.1. *Let $h \in \mathcal{M}$ be a letter-to-letter morphism. There exist three letter-to-letter ts-sequential functions $\sigma_1, \sigma_2, \sigma_3$ such that $h^{-1} = \sigma_1 \sigma_2^+ \sigma_3$. This means that $\mathcal{M}^{-1} \subset \mathcal{S}_{\text{ts},\text{ll}} \mathcal{S}_{\text{ts},\text{ll}+} \mathcal{S}_{\text{ts},\text{ll}}.$*

Proof. Let h be a letter-to-letter morphism from X^* into Y^*. The letters of X are denoted by $\{0, 1, \ldots, k\}$. The idea is to enumerate all words over X with the
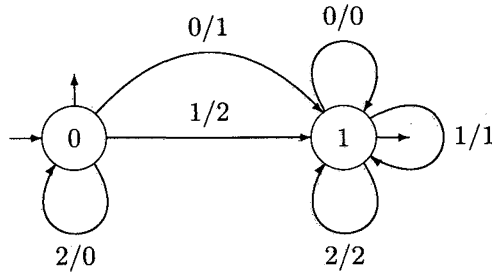
FIGURE 3. A transducer which realizes the successor function.

same length than the input word and to compare their image by h with the input word. We define an order $\prec$ over X^*:

$$u \preceq v \iff |u| = |v| \wedge \tilde{u} \preceq_{lex} \tilde{v},$$

where $\preceq_{lex}$ is the classical lexicographic order. It is clear that $\preceq$ is total over each Σ^i for $i \in \mathbb{N}$. For instance, if $X = \{0, 1, 2\}$, we have:

$$000 \preceq 100 \preceq 200 \preceq 010 \preceq 110 \preceq 210 \preceq 020 \preceq 120 \preceq 220 \preceq 001 \prec \cdots \preceq 222.$$

Hence we define a function σ from X^* into X^* which corresponds to the successor function and where the successor of a word 2^i is 0^i:

$$\forall u \in X^* \setminus \{k\}^* \quad u\sigma = v \text{ such that } \begin{cases} u \neq v \\ u \preceq v \\ \nexists w \in X^* \setminus \{u, v\} \quad u \preceq w \preceq v, \end{cases}$$

$$\forall u \in \{k\}^* \quad u\sigma = 0^{|u|}.$$

It is easy to see that σ belongs to $\mathcal{S}_{ts,ll}$. For instance, if $X = \{0, 1, 2\}$, the function σ is realized by the transducer given Figure 3. And we have $0^i \sigma^+ = X^i$ for all i.

We now define the three letter-to-letter ts-sequential functions:

$$
\begin{aligned}
\sigma_1 : \quad & Y^* && \to (Y \times X)^* \\
& u && \mapsto v && \text{such that } v\Pi_1 = u \wedge v\Pi_2 = 0^{|u|} \\
\sigma_2 : \quad & (Y \times X)^* && \to (Y \times X)^* \\
& u && \mapsto v && \text{such that } v\Pi_1 = u\Pi_1 \wedge v\Pi_2 = u\Pi_2\sigma \\
\sigma_3 : \quad & (Y \times X)^* && \to X^* \\
& u && \mapsto v && \text{such that } v = u\Pi_2 \wedge u\Pi_2 h = u\Pi_1.
\end{aligned}
$$

It is easy to see that, for all $u \in Y^*$, the set $u\sigma_1\sigma_2^+$ contains all words over $(Y \times X)$ with u on the first components and any word over X on the second components.

The function σ_3 selects the word on second components when its image by h is u. Then we have $\sigma_1 \sigma_2^+ \sigma_3 = h^{-1}$.   □

It is then easy to show the main result of this section.

Theorem 8.2. *The classes* $\mathrm{Rat}(\mathcal{S}_{ts,ll})$, $\mathrm{Rat}(\mathcal{S}_{ts})$, $\mathrm{Rat}(\mathcal{S})$, $\mathrm{UCI}(\mathcal{F})$ *and* $\mathrm{UCI}(\mathcal{T})$ *coincide.*

Proof.   Since we have $\mathrm{Rat}(\mathcal{S}_{ts,ll}) \subseteq \mathrm{Rat}(\mathcal{S}_{ts}) \subseteq \mathrm{Rat}(\mathcal{S}) \subseteq \mathrm{UCI}(\mathcal{F}) \subseteq \mathrm{UCI}(\mathcal{T})$, it suffices to show that $\mathrm{Rat}(\mathcal{S}_{ts,ll}) = \mathrm{UCI}(\mathcal{T})$.   We just have to notice that $\mathcal{T} = \mathcal{M}^{-1}(\cap \mathrm{Rec})\mathcal{M}$ is included in $\mathrm{Rat}(\mathcal{S}_{ts,ll})$ by using Lemma 8.1 and the fact that the intersection with a regular language and a morphism are letter-to-letter ts-sequential functions.   □

The next question concerns the existence of a representation like in Theorem 4.3.

Proposition 8.3. *The family* $\mathrm{UCI}(\mathcal{T})$ *is equal to the* $\mathcal{S}_{ts,ll}(\mathcal{S}_{ts,ll} + \mathcal{S}_{ts,ll}) + \mathcal{S}_{ts,ll}$. *This means that a transduction* τ *belongs to* $\mathrm{UCI}(\mathcal{T})$ *if and only if* $\tau = \sigma_1(\sigma_2 + \sigma_3)^+ \sigma_4$ *for some* $\sigma_1, \sigma_2, \sigma_3, \sigma_4 \in \mathcal{S}_{ts,ll}$.

Proof.   Let τ be a transduction belonging to $\mathrm{UCI}(\mathcal{T})$. According to Corollary 4.5 and Lemma 8.1, we have

$$\tau = (\cap X^*)(\sigma_1 \sigma_2^+ \sigma_3)^+ \varphi$$

for some alphabet X, some functions $\sigma_1, \sigma_2, \sigma_3 \in \mathcal{S}_{ts,ll}$ and some letter-to-letter morphism φ. By renaming the different alphabets used in the functions, we can suppose that:

$$\mathrm{alph}(\mathrm{Dom}\, \sigma_1) \cap \mathrm{alph}(\mathrm{Im}\, \sigma_2) = \mathrm{alph}(\mathrm{Dom}\, \sigma_1) \cap \mathrm{alph}(\mathrm{Im}\, \sigma_1)$$
$$= \mathrm{alph}(\mathrm{Im}\, \sigma_1) \cap \mathrm{alph}(\mathrm{Dom}\, \sigma_3) = \mathrm{alph}(\mathrm{Im}\, \sigma_1) \cap \mathrm{alph}(\mathrm{Dom}\, \varphi)$$
$$= \mathrm{alph}(\mathrm{Dom}\, \sigma_2) \cap X = \mathrm{alph}(\mathrm{Dom}\, \sigma_2) \cap \mathrm{alph}(\mathrm{Im}\, \sigma_3)$$
$$= \mathrm{alph}(\mathrm{Im}\, \sigma_2) \cap \mathrm{alph}(\mathrm{Dom}\, \varphi) = \mathrm{alph}(\mathrm{Dom}\, \sigma_3) \cap X$$
$$= \mathrm{alph}(\mathrm{Dom}\, \sigma_3) \cap \mathrm{alph}(\mathrm{Im}\, \sigma_3) = \emptyset.$$

Hence, we obtain $\tau = (\cap X^*)(\sigma_1 + \sigma_2 + \sigma_3)^+ \varphi$. Since the alphabets of the domains of σ_1 and σ_3 are disjoint, their union is still a letter-to-letter ts-sequential function and the result holds.   □

We are not able to obtain a finer characterization even by using more general l.p. rational functions such as $\mathcal{S}$, $\mathcal{S}_{ts}$ or $\mathcal{F}$. But it is still an open question whether or not $\mathrm{UCI}(\mathcal{T})$ and $\mathcal{F}\mathcal{F}_+\mathcal{F}$ coincide.

But these results allow us to answer to an open question of Wood who studies in [24] the generation of context-sensitive languages by using iteration of propagating deterministic GSM maps. A propagating deterministic GSM map is a rational function which can be realized by a deterministic transducer $T = (X, Y, Q, \delta, I, F)$ such that $\mathrm{Dom}\, \delta \subseteq Q \times X$ and $\mathrm{Im}\, \delta \cap Q \times \{\varepsilon\} = \emptyset$. We call these maps propagating ts-sequential functions and the interesting point is that this class contains the

class $\mathcal{S}_{\mathrm{ts},\mathrm{ll}}$. The author defines, for $n > 0$, a class of languages called $\mathcal{L}(n\text{-EPD}\Gamma)$ which corresponds to the classes of languages which can be generated by applying to a letter the iteration of the union of n propagating ts-sequential functions. Wood shows that the class $\mathcal{L}(3\text{-EPD}\Gamma)$ coincides with the class of ε-free context-sensitive languages and asks whether or not this holds for the classes $\mathcal{L}(2\text{-EPD}\Gamma)$ and $\mathcal{L}(1\text{-EPD}\Gamma)$.

By using Corollary 5.5 and Proposition 8.3, we show that $\mathcal{L}(2\text{-EPD}\Gamma)$ also coincides with the class of context-sensitive languages.

Proposition 8.4. *Let A be an ε-free language over the alphabet Σ. It is a context-sensitive language if and only if there exist two propagating ts-sequential fonctions φ, ψ and a letter S such that $A = S(\varphi + \psi)^+(\cap \Sigma^+)$. This means that $\mathcal{L}(2\text{-EPD}\Gamma)$ = CS_ε.*

Proof. The right to left implication is obvious since $\mathcal{L}(2\text{-EPD}\Gamma)$ is included in $\mathcal{L}(3\text{-EPD}\Gamma) = CS_\varepsilon$. For the reverse implication, we use Corallary 5.5 which can be written like this: $A = a^* \tau^+ (\cap \Sigma^*)$ where τ is a transduction of $UCI(\mathcal{T})$. Since A is ε-free, we have $A = a^+ \tau^+ (\cap \Sigma^+)$.

Moreover, by using the same idea than in Proposition 8.3, we have:

$$A = a^*(\sigma_1 + \sigma_2)^+(\cap \Sigma^+),$$

where σ_1 and σ_2 are two letter-to-letter ts-sequential functions. Let S be a new letter which does not belong to any alphabet. We define two propagating ts-sequential functions σ_1' and σ_2':

$$\sigma_1' = \{(S^n, S^{n+1}) \mid n \in \mathbb{N}^*\},$$
$$\sigma_2' = \{(S^n, a^n) \mid n \in \mathbb{N}^*\}.$$

It is easy to see that A can be defined as (we use the fact that the alphabets of the domains of σ_1' and σ_2' are different thant those of σ_1 and σ_2):

$$A = S(\sigma_1' + \sigma_2')^+(\sigma_1 + \sigma_2)^+(\cap \Sigma^+) = S(\sigma_1' + \sigma_2' + \sigma_1 + \sigma_2)^+(\cap \Sigma^+).$$

To conclude, it suffices to notice that the alphabets of the domain of σ_1 and σ_1' (resp. σ_2 and σ_2') are disjoint and that this implies that $\varphi = \sigma_1 + \sigma_1'$ (resp. $\psi = \sigma_2 + \sigma_2'$) is a propagating ts-sequential function. $\square$

9. REPRESENTATION THEOREM AND CLOSURE PROPERTIES

If we recapitulate results stated by Theorem 4.3, Proposition 5.2, Theorem 6.4, Proposition 7.3, Theorem 8.2 and Proposition 8.3, we get the following representation theorem.

Theorem 9.1 (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 ($\tau \in \mathrm{UCI}(\mathcal{T})$);*

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 $\tau = \varphi^{-1}(\cap A)\psi$ ($\tau \in \mathcal{T}_{\mathrm{CS}}$);*

5. *there exists a recognizable picture language L such that $\tau \setminus \{(\varepsilon, \varepsilon)\} = \tau_L$ ($\tau \in \mathcal{T}_{\mathrm{Rec}(PL)}$);*

6. *the transduction τ can be defined by using union, composition and iteration of letter-to-letter ts-sequential functions ($\tau \in \mathrm{Rat}(\mathcal{S}_{\mathrm{ts,ll}})$);*

7. *there exist four letter-to-letter ts-sequential functions σ_1, σ_2, σ_3 and σ_4 such that $\tau = \sigma_1(\sigma_2 + \sigma_3)^+ \sigma_4$ ($\tau \in \mathcal{S}_{\mathrm{ts,ll}}(\mathcal{S}_{\mathrm{ts,ll}} + \mathcal{S}_{\mathrm{ts,ll}})_+\mathcal{S}_{\mathrm{ts,ll}}$).*

These different characterizations of $\mathrm{UCI}(\mathcal{T})$ give to this class a kind of robustness and allow us to show some closure properties. We already know that $\mathrm{UCI}(\mathcal{T})$ is closed under union, composition, iteration and inverse. The following proposition states two additional closure properties which are not obvious with the definition of the class.

Proposition 9.2. *The class of transductions $\mathrm{UCI}(\mathcal{T})$ is closed under intersection and difference.*

Proof. Let $\tau, \sigma \in \mathrm{UCI}(\mathcal{T})$ be two transductions from X^* into Y^*. According to Lemma 6.6, we have $\tau = \Pi_1^{-1}(\cap A)\Pi_2$ and $\sigma = \Pi_1^{-1}(\cap B)\Pi_2$ where A and B are two context-sensitive languages over $X \times Y$. The intersection and the difference of τ and σ can be written as follows:

$$
\begin{aligned}
\tau \cap \sigma &= \{(u,v) \mid (u,v) \in \tau \wedge (u,v) \in \sigma\}, \\
&= \Pi_1^{-1}(\cap(A \cap B))\Pi_2, \\
\tau \setminus \sigma &= \{(u,v) \mid (u,v) \in \tau \wedge (u,v) \notin \sigma\}, \\
&= \Pi_1^{-1}(\cap(A \cap \overline{B})\Pi_2.
\end{aligned}
$$

We simply use the fact that the class of context-sensitive languages is closed under intersection and complement [9,21]. So $A \cap B$ and $A \cap \overline{B}$ are also context-sensitive and the closure properties hold.   □

Let us remark that the closure under intersection as well as several results – like the first characterization given in Theorem 4.3 – can be shown by using recognizable picture languages. Notice that the class of recognizable picture languages is not closed under complement [6] and then neither under difference.

This last closure property allows us to obtain an unexpected characterization of UCI($\mathcal{T}$) with one-step transductions.

Proposition 9.3. *The family* UCI($\mathcal{T}$) *is equal to the smallest family of transductions which contains* $\mathcal{O}$ *and is closed under union, composition, iteration and difference.*

Proof. We use the proof of the Proposition 5.2. In this proof, we show that a transduction $\tau \in$ UCI($\mathcal{T}$) is equivalent to $\sigma_1 \sigma_2^+ \sigma_3$ where σ_1 marks the begin and the end letters, σ_2 is a one-step transduction and σ_3 is an intersection with a free monoid followed by an unmarking.

We just have to show that marking, intersection with a free monoid and unmarking can be realized by union, composition and difference of one-step transductions.

It is obvious that the identity over the alphabet Σ – in other words the intersection with Σ^* – is the one-step transduction associated with the couple $(\Sigma, \{\varepsilon\})$.

Let $\sigma \subseteq X^* \times (X \cup \underline{X})^*$ be a l.p. rational transduction which marks the end letter of the input words:

$$\sigma = \{(wa, w\underline{a}) \mid w \in X^*, a \in X\}.$$

Let ρ and ρ' be the one-step transductions associated respectively with $(X \cup \underline{X}, P)$ and $(X \cup \underline{X}, P')$ where P and P' are defined by:

$$
\begin{aligned}
P &= \{(a, \underline{a}) \mid a \in X\}, \\
P' &= \{(ab, \underline{a}b) \mid a, b \in X\}.
\end{aligned}
$$

The transduction σ is equivalent to $(\cap X^*)(\rho \setminus \rho')$. In the same way a transduction which marks the first letter is realized by difference on two one-step transduction.

It is obvious that the unmarking – where there is only one marked letter – can be realized by a one-step transduction. $\qquad\square$

10. CONCLUSION

The class UCI($\mathcal{T}$) is an interesting class of transductions which corresponds to length-preserving computations where the work space is bounded by the size of the input. Nevertheless, the computation cannot always be deterministic. If we show in Proposition 8.3 that we can use ts-sequential functions, but the iterated function is a union of two functions and then can be non-deterministic. It is an open question whether or not this class, which corresponds to length-preserving context-sensitive transductions, is equal to l.p. deterministic context-sensitive transductions. This last class is studied in a paper in preparation [20] and shown to be equal to $\mathcal{F}\mathcal{F}_+\mathcal{F}$.

We can notice that it is undecidable whether or not a given context-sensitive transduction is a rational transduction (since it is equivalent to test the recognizability of a context-sensitive language). But we can consider restricted classes of context-sensitive transductions. For instance iterated one-step transductions with only one rule have already been studied in [16] and the decidability of the rationality has been conjectured:

Conjecture 10.1 (Lilin 91 [16]). *Let τ be a one-step transduction associated with (X, P) where $P = \{u \rightarrow v\}$ and $u \neq v$. The transduction τ^+ is rational if and only if there do not exist words $x, y, z \in X^*$ such that $u = xyz$ and $v = zyx$.*

REFERENCES

[1] A. Avizienis, Signed-digit number representations for fast parallel arithmetic. *IRE Trans. Electronic Computers* **10** (1961) 389–400.

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

[3] M. Clerbout and M. Latteux, Partial commutations and faithful rational transductions. *Theoret. Comput. Sci.* **34** (1984) 241-254.

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

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

[6] D. Giammarresi and A. Restivo, Two-dimensional languages, in *Handbook of Formal Languages*, edited by A. Salomaa and G. Rozenberg, Vol. 3. Springer-Verlag, Berlin (1997) 215–267.

[7] S.A. Greibach, Full AFL's and nested iterated substitution. *Inform. and Control (Shenyang)* **16** (1970) 7–35.

[8] T. Harju and J. Karhumäki, Morphisms, in *Handbook of Formal Languages*, edited by A. Salomaa and G. Rozenberg, Vol. 1. Springer-Verlag, Berlin (1997) 439–510.

[9] N. Immerman, Nondeterministic space is closed under complementation. *SIAM J. Comput.* **17** (1988) 935–938.

[10] S.-Y. Kuroda, Classes of languages and linear bounded automata. *Inform. and Control (Shenyang)* **7** (1964) 207–223.

[11] M. Latteux and D. Simplot, Recognizable picture languages and domino tiling. *Theoret. Comput. Sci.* **178** (1997) 275–283.

[12] M. Latteux and D. Simplot, Context-sensitive string languages and recognizable picture languages. *Inform. and Comput.* **138** (1997) 160–169.

[13] 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.

[14] M. Latteux and P. Turakainen, On characterizations of recursively enumerable languages. *Acta Inform.* **28** (1990) 179-186.

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

[16] É. Lilin, Une généralisation des semi-commutations. Tech. Rep. it-210, L.I.F.L., Université Lille 1, France (1991).
[17] J.-M. Muller, Some characterizations of functions computable in on-line arithmetic. *IEEE Trans. Comput.* **43** (1994) 752–755.
[18] M. Nivat, Transductions des langages de Chomsky. *Ann. Inst. Fourier (Grenoble)* **18** (1968) 339–456.
[19] M.P. Schützenberger, Sur les relations rationnelles entre monoïdes libres. *Theoret. Comput. Sci.* **3** (1976) 243–259.
[20] D. Simplot and A. Terlutte, Closure under union and composition of iterated rational transductions (in preparation).
[21] R. Szelepcsényi, The method of forced enumeration for nondeterministic automata. *Acta Inform.* **26** (1988) 279–284.
[22] M. Szijártó, A classification and closure properties of languages for describing concurrent system behaviours. *Fund. Inform.* **4** (1981) 531-549.
[23] P. Turakainen, Transducers and compositions of morphisms and inverse morphisms, in *Studies in honour of Arto Kustaa Salomaa on the occasion of his fiftieth birthday. Ann. Univ. Turku. Ser. A I* **186** (1984) 118–128.
[24] D. Wood, Iterated a-NGSM maps and Γ systems. *Inform. and Control (Shenyang)* **32** (1976) 1–26.