# Families of automata characterizing context-sensitive languages 

Christophe Morvan, Chloé Rispal

## To cite this version:

Christophe Morvan, Chloé Rispal. Families of automata characterizing context-sensitive languages.
Acta Informatica, 2005, 41, pp.293-314. 10.1007/s00236-004-0160-0 . inria-00525422

HAL Id: inria-00525422
https://inria.hal.science/inria-00525422
Submitted on 11 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Families of automata characterizing context-sensitive languages 

Christophe Morvan, Chloé Rispal<br>Institut Gaspard Monge, Université de Marne-la-Vallée 5, bd Descartes, 77454 Marne-la-Vallée cedex 2 France<br>e-mail: \{christophe.morvan|chloe.rispal\}@univ-mlv.fr


#### Abstract

In the hierarchy of infinite graph families, rational graphs are defined by rational transducers with labelled final states. This paper proves that their traces are precisely context-sensitive languages and that this result remains true for synchronized rational graphs.


## 1 Introduction

During the last fifteen years, there has been a great deal of interest around families of infinite graphs. The decidable properties of these families provide a nice framework for validation and verification. Muller and Schupp introduced in [16] the transition graphs of pushdown automata and proved that their monadic second order theory was decidable. A few years later, Courcelle extended this result to regular graphs generated by deterministic graph grammars, [7]. In 1996 Caucal used inverse rational substitution (followed by a rational restriction) to define the prefix-recognizable graphs; they have a decidable second order monadic theory [4]. The automatic graphs form a more general family of graphs. They are automatic structures, defined in 2000 by Blumensath and Grädel [1], and have, thus, a decidable first order theory. Very recently Colcombet considered an interesting extension of prefix-recognizable graphs, namely the VRP-graphs (vertex replacement with product) [6]. They are obtained using vertex replacement systems and a graph product. Their first-order theory with accessibility is decidable.
The study of infinite graph families is also naturally linked to language theory. Precisely, the transition graphs of pushdown automata and prefix-recognizable graphs are defined from language theory. Recently, Urvoy extended the work of Ginsburg and Greibach [20] to define abstract families of graphs [22]. The connection between families of graphs and language theory is even deeper: they constitute an elegant characterization of families of languages. If we consider the trace of a graph as the language of path labels leading from an initial set of vertices to a final set of vertices, then traces form one of the most important link between graphs and languages. For example, it is well known that the traces of finite graphs are regular languages [11]. By construction, the traces of the transition graphs of pushdown automata are the context-free languages. These languages are also the traces of prefix-recognizable graphs [4]. At this time, the languages corresponding to the VRPgraphs is still unknown. In 2001 Caucal used Turing machines to define a class of graphs whose traces are recursively enumerable languages [3].

In this paper we establish a new correspondence between the Chomsky hierarchy [5] and families of graphs. We prove that the traces of rational graphs (generated by labelled rational transducers [14]), are context-sensitive languages. We show that this result remains true if we restrict to synchronized graphs [18]. In those cases the traces correspond to path labels between finite sets. Extending initial and final sets to rational sets, letter-to-letter rational graphs also trace context-sensitive languages.
This article is organized in three sections. The first one uses finite transducers, that is finite automata labelled with pairs, to define the rational graphs. Some basic results and definitions about context-sensitive languages are also recalled. The second section proves that the trace of any rational graph can be recognized using a linear bounded Turing machine, and is therefore a context-sensitive languages. Finally, the third section uses the Penttonen normal form [17] to prove that any context-sensitive language is the trace of a rational graph. Indeed, it proves that the synchronized rational graphs, which is a proper subclass of rational graphs [1], [21], are sufficient to obtain any context-sensitive language.

## 2 Preliminary definitions

In this section, we recall basic definitions concerning infinite graphs and context-sensitive languages. In the first part, rational graphs, synchronized graphs and letter-to-letter graphs are defined from transducers. Then, context-sensitive languages are characterized both from Turing machines and from Penttonen's rewriting systems in the second part.

### 2.1 Graphs and transducers

Let $\mathcal{A}$ be a finite set of labels. A simple arc labelled graph is a subset of $V \times \mathcal{A} \times V$ where $V$ is an arbitrary set of vertices.
We denote by $s \underset{G}{\frac{a}{b}} t$ the existence of the arc $(s, a, t)$ in the graph $G$ or simply by $s \xrightarrow{a} t$ when there is no ambiguity.
A path $s \stackrel{u}{\Longrightarrow} t$ of a graph $G$ leading from a vertex $s$ to a vertex $t$ and of label $u$ is a finite sequence $\left(s_{0}, a_{1}, s_{1}\right) \ldots\left(s_{n-1}, a_{n}, s_{n}\right)$ of arcs of $G$ such that $u=a_{1} \ldots a_{n}, s=s_{0}$ and $t=s_{n}$. A trace of a graph $G$ is the language $L(G, I, F)$ of path labels leading from a set $I$ of initial vertices to a set $F$ of final vertices:

$$
L(G, I, F)=\{u \mid \exists s \in I \exists t \in F, s \underset{G}{u} t\}
$$

An automaton $A$ is a triple $(G, I, F)$ where $G$ is a finite graph and $I$ and $F$ are initial and final sets of states. The language, $L(A)$, recognized by $A$ is the trace $L(G, I, F)$.
Let $\Sigma$ be a finite alphabet. We denote by $\Sigma^{*}$ the set of finite words over letters of $\Sigma$, and we write $\varepsilon$ for the empty word.
A transducer $T$ is a finite automaton where labels have been modified to recognize relations instead of sets of words. It is defined by a finite subset of $Q \times \Sigma^{*} \times \Sigma^{*} \times Q$ of labelled $\operatorname{arcs}$, where $Q$ is a finite set of states, by a set $I \subseteq Q$ of initial states, and by a set $F \subseteq Q$ of final states. So a transducer is labelled by pairs of words. Any transition $(p, u, v, q)$ of a transducer $T$ is denoted by $p \xrightarrow[T]{u / v} q$ or by $p \xrightarrow{u / v} q$ when $T$ is understood.
A path $p_{0} \xrightarrow{u_{1} / v_{1}} p_{1} \ldots p_{n-1} \xrightarrow{u_{n} / v_{n}} p_{n}$ with $u=u_{1} \ldots u_{n}$ and $v=v_{1} \ldots v_{n}$ is labelled $u / v$ and is denoted by $p_{0} \stackrel{\sim}{u / v} p_{n}$. A path is successful if it leads from an initial state to a final one. A pair $(u, v) \in \Sigma^{*} \times \Sigma^{*}$ is recognized by a transducer if there exists a successful path labelled $u / v$.

Definition 2.1. A relation is rational if it is recognized by a transducer.
We denote by $\operatorname{Rat}\left(\Sigma^{*} \times \Sigma^{*}\right)$ the set of binary rational relations. The following transducer, with initial state 0 (marked by an incoming arrow) and final state 1 (marked by a double circle) recognizes the rational relation $\left\{\left(A^{n} B^{m}, B^{n} A^{2 m}\right) \mid n \geq 0, m>0\right\}$.


## 2.2 hierarchy of rational graphs

Using words as vertices, infinite graphs can be defined by the relations between the extremities of its arcs. Given any graph $G \subseteq \Sigma^{*} \times \mathcal{A} \times \Sigma^{*}$, we denote by $\underset{G}{a}$ the relation $\{(s, t) \mid(s, a, t) \in G\}$. The graph $G$ is rational if for each $a \in \mathcal{A}$, the relation $\underset{G}{a}$ is rational.
For instance, the following graph, on the left, called the grid is a rational graph since it is defined by the following transducer, on the right.


Subfamilies of rational graphs are defined from subsets of rational relations.
If a transducer has labels over $\Sigma \times \Sigma$ it is called a letter-to-letter transducer: it is a transducer labelled by pairs of letters instead of pairs of words.

Definition 2.2. A relation is letter-to-letter if it is recognized by a letter-to-letter transducer.
A graph $G$ is a letter-to-letter graph if for each $a \in \mathcal{A}$, the relation $G_{a}$ is letter-to-letter. Another particular subset of rational relations called left-synchronized relations has been studied by Elgot and Mezei [8] and then by Frougny and Sakarovitch [10]. Those relations are recognized by letter-to-letter transducers with rational terminal functions completing one side of the recognized pairs. The terminal function associates a relation to each terminal state of the transducer. Then, the relation defined is the set of labels of path ending at a state $q$, concatenated with pairs of the terminal function's value in $q$. For example, a pair $(u, v)$ belongs to a synchronized relation $R$, if there exists two pairs of words $\left(u^{\prime}, v^{\prime}\right)$ and $\left(u^{\prime \prime}, v^{\prime \prime}\right)$ such that $(u, v)=\left(u^{\prime}, v^{\prime}\right) \cdot\left(u^{\prime \prime}, v^{\prime \prime}\right)$ with the following condition: there exists a terminal state $q$, a path labelled $\left(u^{\prime}, v^{\prime}\right)$ leading to $q$, and $\left(u^{\prime \prime}, v^{\prime \prime}\right)$ belongs to the value of the terminal function in $q$. Formally:

Definition 2.3. A relation over $\Sigma^{*} \times \Sigma^{*}$ is left-synchronized if it is recognized by a letter-to-letter transducer $T$ with terminal function $f$ taking values in

$$
\operatorname{Dif}_{\text {Rat }}=\left(\operatorname{Rat}\left(\Sigma^{*}\right) \times\{\varepsilon\}\right) \cup\left(\{\varepsilon\} \times \operatorname{Rat}\left(\Sigma^{*}\right)\right)
$$

That is, a left-synchronized relation is a finite union of elementary relations of the form $R . S$ where $R \in \operatorname{Rat}\left((\Sigma \times \Sigma)^{*}\right)$ and $S \in \operatorname{Dif}_{\text {Rat }}$. Right-synchronized relations are defined symmetrically with an initial rational function. A rational relation is synchronized if it is left-synchronized or right-synchronized.
Example 2.4. Let us consider the relation $\left.\right|_{2}$ defined by $\left.x\right|_{2} y$ if $x$ is a power of 2 dividing $y$. Provided integers are coded in base 2 (with lowest bits on the left), the relation $\left.\right|_{2}$ is leftsynchronized. This relation is recognized by the following letter-to-letter transducer with the terminal function $f$ defined by $f(q)=(\varepsilon, 0)^{*}(\varepsilon, 1)\{(\varepsilon, 0),(\varepsilon, 1)\}^{*}$ and $f(r)=(\varepsilon, \varepsilon)$.


As the terminal function is rational, it can be introduced in the transducer adding states and transitions. A left-synchronized transducer is a transducer such that each path leading from an initial vertex to a final one can be divided into two parts: the first one only contains arcs of the form $\{p \xrightarrow{A / B} q \mid p, q \in Q \wedge A, B \in \Sigma\}$ while the second part contains either arcs of the form $\{p \xrightarrow{A / \epsilon} q \mid p, q \in Q \wedge A \in \Sigma\}$ or arcs of the form $\{p \xrightarrow{\epsilon / B} q \mid p, q \in Q \wedge B \in \Sigma\}$ (but not both).
Remark 2.5. Automatic structures, [1], or automatic groups, [9], are defined by automatic relations which are equivalent to synchronized relations.

Example 2.6. The following left-synchronized transducer recognizes the left-synchronized relation of Example 2.4.


A graph $G$ is left-synchronized if for each $a \in \mathcal{A}$, the relation $G_{a}$ is left-synchronized.
Subfamilies of rational relations are closed under union, intersection and complementation.
Theorem 2.7. [8] The rational left-synchronized relations (respectively letter to letter relations) form a boolean algebra.

A very important consequence of this result is the decidability of the first order theory of the graphs defined using synchronized relations.
We also use particular left-synchronized relations. A binary relation $R$ is recognizable if it is a finite union of products $S \times T$ where $S, T \in \operatorname{Rat}\left(\Sigma^{*}\right)$. A binary relation $R$ over words is of bounded length difference if there exists an integer $b$ such that $||u|-|v|| \leq b$ for any $(u, v) \in R$.
Proposition 2.8. [10] The family of synchronized relations contains the recognizable relations and the rational relations of bounded length difference.

When working with rational or synchronized graphs, it is sufficient to consider the traces between singletons instead of rational sets.
Lemma 2.9. Let $G \subseteq \Sigma^{*} \times \mathcal{A} \times \Sigma^{*}$ be a left-synchronized graph.
Let $I, F \in \operatorname{Rat}\left(\Sigma^{*}\right)$ and $\$, \# \notin \Sigma$.
There exists a left-synchronized graph $H \subseteq\left(\Sigma^{*} \cup\{\$, \#\}\right) \times \mathcal{A} \times\left(\Sigma^{*} \cup\{\$, \#\}\right)$ such that $L(G, I, F)=\bar{L}(H,\{\$\},\{\#\})$.
Proof. i) For all $a \in \mathcal{A}$, we define:

$$
F_{a}:=\operatorname{Dom}\left(\frac{a}{G} \cap \Sigma^{*} \times F\right)
$$

the set of vertices which are source of an arc leading to a final state. This set is rational being the domain of a rational relation. Then we create new arcs leading from those vertices to the vertex $\#$. More precisely, for all $a \in \mathcal{A}$, we define the arcs of the graph $G^{\prime}$ to be as follows:

$$
\left.\xrightarrow[G^{\prime}]{a}:=\underset{G}{a} \cup F_{a \times a \times\{ } \times \#\right\}
$$

This relation is left-synchronized as the union of a left-synchronized relation with a recognizable set. Moreover and by construction,

$$
L(G, I, F)=L\left(G^{\prime}, I,\{\#\}\right)
$$

ii) By a symmetric argument, a graph $G^{\prime \prime}$ is defined such that,

$$
L\left(G^{\prime}, I,\{\#\}\right)=L\left(G^{\prime \prime},\{\$\},\{\#\}\right) .
$$

### 2.3 Context-sensitive languages

In Chomsky's hierarchy of languages, the family of context-sensitive languages (Csl) is located between recursively enumerable and context-free languages. There are many different ways to characterize this family of languages. In the following, we recall two of those characterizations. The first one, due to Kuroda [12], defines context-sensitive languages as the languages recognized by Linearly Bounded Turing machines (LBM). The second characterization due to Penttonen [17], is based on a particular rewriting system.

## Context-sensitive languages from Turing machines

A linearly bounded machine is a Turing machine such that the size of the tape is bounded, linearly, by the length of the input. These machines are a classical characterization of Csl.
Theorem 2.10. [12] Context-sensitive languages are the set of languages recognized by linearly bounded Turing machines.

## Penttonen's characterization of Context-sensitive languages

A different characterization of Csl, due to Penttonen [17], is based on a rewriting system using particular rules.
Definition 2.11. A rewriting system $\Gamma=\Gamma_{1} \cup \Gamma_{2}$ is a 2-system if every rule of $\Gamma_{2}$ is of the form $A B \rightarrow A C$ with $B \neq C$ and every rule of $\Gamma_{1}$ is of the form $A \rightarrow a$ where $A, B, C$ are letters of the non-terminal alphabet $\Sigma$ and $a \in \mathcal{A}$.

Context-sensitive languages are obtained by derivation of a 2 -system from a linear language.
A language is linear if it can be generated by a grammar whose rules are of the form $Z \longrightarrow W$, where $Z$ is a non-terminal, $W$ is a word over terminal and non-terminal symbols, with at most one non-terminal.

Theorem 2.12. [17] There exists a linear language $L_{\text {Lin }}$ such that every context-sensitive language is $\left\{v \in \mathcal{A}^{*} \mid \exists u \in L_{\text {Lin }}, u \underset{\Gamma}{*} v\right\}$ for some 2 -system $\Gamma$.

## 3 From rational graphs to context-sensitive languages

In this section, we prove that the traces of rational graphs between initial and final rational sets of vertices are context-sensitive languages, this exposition is a detailed (and simplified) version of the first section of [15].

### 3.1 First approach

As we have seen in Section 2.3, a common characterization of Csl is given by LBM. The first idea is to simulate a rational graph with an LBM. Any vertex of the graph would be stored on the tape, and the machine would compute the next vertex.
This basic approach fails to recognize the traces of rational graphs: the length of the vertices may grow exponentially. Example 3.1 illustrates this situation.

Example 3.1. The transducer, having a single state $p$ (initial and final labelled $a$ ) and a single transition $p \xrightarrow{A / A A} p$ defines the following graph:


The trace of this graph between $A$ and $A^{*}$ is obviously $a^{*}$. The problem is that the length of the vertices is exponential in the length of the recognized word. For example, the path recognizing $a^{3}$ is the following:

$$
A \xrightarrow{a} A A \xrightarrow{a} A^{4} \xrightarrow{a} A^{8}
$$

More generally: $a^{n}$ leads from $A$ to $A^{2^{n}}$. Therefore, it is not straightforward to construct a linear bounded machine recognizing the language of the transducer.

The last remarks leads to encode the vertices of the graph in order to keep their length linear. In this case it becomes difficult to compute the "next vertex function". Especially if some branches of the transducer produce a sub-graph with a linear growth, and some other with an exponential growth.
The next section exposes a different approach which avoids those difficulties.

### 3.2 Construction of the LBM

Let $G \in \Sigma^{*} \times \mathcal{A} \times \Sigma^{*}$ be a rational graph recognized by a transducer $T$. For each $a$ in $\mathcal{A}$, we denote by $T_{a}$ the transducer recognizing $\underset{{ }_{c}}{a}$. We construct a LBM recognizing the trace $L(G, I, F)$, where $I$ and $F$ are rational sets.
Roughly speaking, our solution is to simulate the transitions of $G$ in parallel: for example, let us consider a path $U \xrightarrow{a} V \xrightarrow{b} W$ in $G$ and suppose that the first transition of $T_{a}$ is of the form $q \xrightarrow{\epsilon / X} q^{\prime}$. Then $X$ is the first element of $V$, thus we can activate $T_{b}$ knowing that $X$ is the first non-empty left-hand-side label.
Using this observation we only need to keep on the tape of the machine one state for each transducer $T_{a}$, plus some bounded information corresponding to what it might consume and what it has produced (and that has not been consumed yet).
By Lemma 2.9 we suppose that $I$ and $F$ are singletons containing respectively $\$$ and $\#$.
A transducer is normalized if all its transitions are of the form, $p \xrightarrow{u / v} q$, where $|u|+|v|=1$. It is straightforward to see that any rational graph can be generated by a normalized transducer.
In order to present the LBM that we construct, we simply give moves corresponding to obvious sequences of ordinary LBM transitions. Let $M=(Q, W, \triangleright, F, \mathcal{R})$ be a LBM, where $W$ contains the elements of $\Sigma$, the states of $T, \varepsilon$ and left and right end-markers respectively denoted by $\Delta$ and $\nabla$. The elements of $Q$ are not described in details, we only need to specify two macro states (allowing to initiate a move): $\triangleright$ and $\downarrow$, $\triangleright$ being the initial state of the machine. The set $F$ contains a single state $(\diamond)$.
The initial configuration is the following:

$$
\Delta \triangleright w \square^{|w|+1} \nabla .
$$

There are $|w|+1$ blank symbols after $w$ because the first transitions produce this configuration:

$$
\Delta \$ i_{w(0)} \varepsilon i_{w(1)} \cdots i_{w(|w|)} \varepsilon \nabla .
$$

For this configuration, each symbol $i_{w(k)}$ is the initial state of the transducer corresponding to the letter $w(k)$ (denoted by $\left.T_{w(k)}\right)$. In each configuration of this machine, the even positions correspond to states of the transducers (the machine has to simulate $|w|$ transducers). For example, let us suppose that $A q_{a} B q_{b} C$ is a factor of some configuration of the machine, the letter $A$ corresponds to the left hand side of a transition in transducer $T_{a}$ starting from $q_{a}, B$ corresponds to the right hand-side of some transition in $T_{a}$ ending in state $q_{a}$ (it can be interpreted as: transducer $T_{a}$ has produced $B$ and has to consume $A)$. It is the same for state $q_{b}$. There are three different moves of the machine:

| Label | Transducer | Initial position | Final position | Comment |
| :---: | :---: | :---: | :---: | :---: |
| Move (a) | $q_{a} \xrightarrow[T_{a}]{A / \varepsilon} q_{a}^{\prime}$ | - $A q_{a} C q_{b} \cdots$ | $\cdots \varepsilon q_{a}^{\prime} \bullet C q_{b} \cdot \cdot$ | $\begin{array}{\|llll} \hline A, C & \in & \Sigma \cup \\ \{s, \#, \varepsilon\} & q_{a} & \text { state } \\ \text { of } T_{a} & & \\ q_{b} \text { state of } T_{b} \text { or } \nabla \\ \hline \end{array}$ |
| Move (b) |  | - $A q_{a} \varepsilon q_{b} \cdots$ | $A q_{a}^{\prime} \bullet B q_{b}$. | $\begin{array}{\|l} \hline A \in \Sigma \cup\{s, \#, \varepsilon\} \\ q_{a} \text { state of } T_{a} \\ q_{b} \text { state of } T_{b} \text { or } \nabla \\ \hline \end{array}$ |
| Move (c) | - | $B q_{b} \bullet C q_{a}$ | $\cdots$ - $\mathrm{q}_{\mathrm{b}} C q_{a}$ | $\begin{aligned} & C=\varepsilon \text { or } \\ & q_{a}=\nabla \text { and } C=\# \end{aligned}$ |

The machine checks for success each time state reaches \# followed by $\nabla$ (if $\nabla$ follows any other non- $\varepsilon$ letter there is no transition, the run fails). It also checks for success if reaches $\varepsilon$ followed by a $f_{w_{i}}$ (a final state of $T_{w_{i}}$ ). In those cases, the machine checks whether for all $i, q_{a_{i}}$ equals $f_{a_{i}}$ and $A_{i}$ equals $\varepsilon$; if it is the case, it is a success. Indeed, it means that everything that has been produced has been consumed, and that each transducer is in an acceptable state (a final state). If there is no success, the machine proceeds to move (a), (b) or (c).

Lemma 3.2. The languages recognized by the machine $M$ is the trace, $L(G, \$, \#)$, of $G$.
Proof. First, we prove that $L(M) \subseteq L_{G}$. Consider a word $w \in L(M)$. From a successful run in $M$, we can deduce paths in the transducers corresponding to the letters of $w$ : all moves done by the machine can be done by a transducer, except those to the left which correspond to a "change of transducer".
Second, we prove that $L_{G} \subseteq L(M)$. Let $w$ be a word in $L_{G}$, and suppose that $n=|w|$. There is a path in $G$ between $\$$ and \# labelled $w: \$ \xrightarrow{w_{1}} u_{1} \xrightarrow{w_{2}} u_{2} \cdots u_{n-1} \xrightarrow{w_{n}}$ \#. Therefore we have: $\$ T_{w_{1}} u_{1}, u_{1} T_{w_{2}} u_{2}, \cdots u_{n-1} T_{w_{n}} \#$.
To construct a successful run of $M$, we use, for all $i$, a path in $T_{w_{i}}$ labelled $u_{i-1} / u_{i}$. We define a new transducer $T_{w_{i}}^{\prime}$ as a copy of the transducer $T_{w_{i}}$, where each $\varepsilon$ is replaced by a letter $E$ (not in $\Sigma$ ). Thus $u_{i-1} T_{w_{i}} u_{i}$ implies $u_{i-1}^{\prime} T_{w_{i}}^{\prime} u_{i}^{\prime \prime}$ with:

$$
u_{i-1}^{\prime}=E^{k_{1}} u_{i-1}(1) E^{k_{2}} \cdots E^{k_{\left|u_{i-1}\right|} \mid} u_{i-1}\left(\left|u_{i-1}\right|\right) E^{k_{\left|u_{i-1}\right|+1}} .
$$

Each $E$ means that a transition labelled $\varepsilon$ on the left, in $T_{w_{i}}$ has been followed. The word $u_{i}^{\prime \prime}$ is similar. Each letter in $u_{i-1}^{\prime}$ witnesses for a transition in $T_{w_{i}}$, and therefore corresponds to a letter in $u_{i}^{\prime \prime}$ (thus, for all $\left.i,\left|u_{i-1}^{\prime}\right|=\left|u_{i}^{\prime \prime}\right|\right)$.
Now we use these words $u_{i}^{\prime}$ to construct a successful run in $M$. The function: first, over words, returns the first letter of a word (nothing if it is the empty word), and the function tail erases the first letter of a word. This process constructs a successful run of $M$ :

Set $i:=1 / *$ index of the transducer */
Set $A_{1}:=\$$ and, for all $i \geqslant 2, A_{i}:=\varepsilon$
Repeat

## Case:

$u_{i}^{\prime \prime}=\varepsilon \quad:$ Follow corresponding move (c)
$i:=i-1$
first $\left(u_{i-1}^{\prime}\right)=E: A_{i+1}:=\operatorname{first}\left(u_{i}^{\prime \prime}\right)\left(\operatorname{first}\left(u_{i}^{\prime \prime}\right) \neq E\right)$ $\operatorname{tail}\left(u_{i-1}^{\prime}\right)$, tail $\left(u_{i}^{\prime \prime}\right), i:=i+1$
Follow corresponding move (b)
first $\left(u_{i-1}^{\prime}\right)=A_{i}: A_{i}:=\varepsilon$
$\operatorname{tail}\left(u_{i-1}^{\prime}\right), \operatorname{tail}\left(u_{i}^{\prime \prime}\right), i:=i+1$
Follow corresponding move (a)
Else $\quad: i:=i-1$, follow corresponding move (c)
Until (For all $\left.i, u_{i}^{\prime}=u_{i}^{\prime \prime}=\varepsilon\right)$
From the construction of the $u_{i}^{\prime}$ and $u_{i}^{\prime \prime}$, this process will always be able to follow a transition. Since all transitions to the right remove letters, the process eventually meets the "out" condition and therefore succeeds. This process yields a path in $M$, recognizing $w$ which concludes the proof.

This construction is illustrated for the graph of Example 3.1.
Example 3.3. The first step consist of normalizing the transducer, and to separate initial and final states.


Once the transducer is transformed, we consider the trace of the graph from $A$ to $\#$ (the vertex $A$ correspond to the vertex $\$$ in the construction of the machine). Now, let us consider the word $a^{3}$ which labels following path:

$$
A \xrightarrow{a} A^{2} \xrightarrow{a} A^{4} \xrightarrow{a} \#
$$

The configurations of the machine are presented on the left. Internal states corresponding to the process are presented on the right (we omit the initial configuration $\triangle \triangleright$ $a a a \square \square \square \square \nabla)$ :

$$
\begin{aligned}
& \text { Apply moves (a),(c),(b) } \\
& \triangle \varepsilon r \varepsilon r>A p \varepsilon \nabla \\
& u_{0}^{\prime}=E \quad u_{1}^{\prime}=E A E E \quad u_{2}^{\prime}=A E A A A \\
& u_{1}^{\prime \prime}=A \quad u_{2}^{\prime \prime}=A E A A \quad u_{3}^{\prime \prime}=E \# E E E \\
& i:=3 \\
& \text { Apply moves (a),(c),(b) } \\
& \triangle \varepsilon r \varepsilon r \varepsilon u \triangleright \# \nabla \\
& \begin{array}{ccc}
u_{0}^{\prime}=E & u_{1}^{\prime}=E A E E & u_{2}^{\prime}=A A A \\
u_{1}^{\prime \prime}=A & u_{2}^{\prime \prime}=A E A A & u_{3}^{\prime \prime}=E E E
\end{array} \\
& \text { Apply moves (c),(c),(b),(a), then (c), (c), (c) } \\
& u_{0}^{\prime}=E \quad u_{1}^{\prime}=A E E \quad u_{2}^{\prime}=A A \\
& u_{1}^{\prime \prime}=A \quad u_{2}^{\prime \prime}=E A A \quad u_{3}^{\prime \prime}=E E \\
& i:=1
\end{aligned}
$$

Finally, following these moves: (a,b,c,a,b,c,c,a,b), the process finishes. We have computed a successful path from the transducer.

From Lemma 3.2 it is easy to prove the desired result.
Proposition 3.4. Traces of rational graphs are context-sensitive languages.
Proof. First, we transform the graph in order to consider the trace between two vertices. Then we construct the corresponding machine $M$ which recognizes the same language, by Lemma 3.2. Thus the traces of rational graphs are Csl.

## 4 From context-sensitive languages to synchronized graphs

In the previous section, we proved that the traces of rational graphs are the contextsensitive languages. Thus any trace of a synchronized graph is a context-sensitive language. Conversely, we show that any context-sensitive language is the trace of a synchronized graph. The proof uses Penttonen's characterization of Csl. It is a detailed construction of [19].
Let $L$ be a context-sensitive language. We construct a synchronized graph whose traces between two finite sets is $L$.
By Theorem 2.12, there exists a 2 -system $\Gamma$ such that $L$ is obtained by derivation of the linear language $L_{\text {lin }}$. Recall that the derivation rules of non-terminal words are of the form $A B \rightarrow A C$. Consider a transducer having transitions $(A, B) \xrightarrow{A / C}(A, C)$ for each $(A B, A C)$ of $\Gamma_{2}$. Any derivation $A B_{1} \xrightarrow{\Gamma_{2}} A B_{2} \ldots \xrightarrow{\Gamma_{2}} A B_{m}$ of a word of length 2 corresponds to an arc $A^{m} \rightarrow B_{1} B_{2} \ldots B_{m}$ on the graph. Following this idea, we first get a rational synchronized graph $G_{\text {Lin }}$ such that $L=L\left(G_{\text {Lin }}, L_{\text {Lin }},\{\epsilon\}\right)$. Then, we transform $G_{\text {Lin }}$ into a graph $G$ having a rational set of vertices $L_{\text {Rat }}$ such that $L=L\left(G, L_{\text {Rat }},\{\epsilon\}\right)$. Finally, using Lemma 2.9, we obtain finite initial and final sets of states.

### 4.1 Traces from the linear language $L_{\text {Lin }}$

Let $T_{2}$ be the transducer defined from $\Gamma_{2}$ by:

$$
\left.\left.\begin{array}{rlll}
I & \xrightarrow{[A /[B} & (B, A, B) & \text { for all } A, B \in \Sigma  \tag{type1}\\
(A, B, C) & \xrightarrow{B / D} & (A, B, D) & \text { for all } A, B, C, D \in \Sigma \text { such that } B C \xrightarrow{\longrightarrow} B D
\end{array}\right) \text { (type 2) }\right)
$$

This transducer starting at $I$ and ending at $F$ recognizes pairs of the form

$$
\left(\left[A A_{1} \ldots A_{m}\right] B,\left[B B_{1} \ldots B_{m}\right]\right)
$$

meaning that under the successive contexts $A, A_{1}, \ldots, A_{m}$ the letter $B$ can be rewritten successively $B, B_{1}, \ldots, B_{m}$. If the context does not change: $A_{i}=A_{i+1}$, one can apply a rule $A_{i} B_{i} \underset{r_{2}}{\longrightarrow} A_{i+1} B_{i+1}$. If the context changes: $A_{i} \neq A_{i+1}$, we copy the letter $B_{i}=B_{i+1}$. The first component of states of $T_{2}$ stores the first word of the derivation.
Note that the relation $R_{2}$ recognized by $T_{2}$ is of bounded length difference.

Example 4.1. Let $\Gamma_{2}=\{(A B, A C),(A C, A D),(D A, D E),(E A, E E)\}$.
We have $[A A A] B \quad R_{2}[B C D]$ because under the context $A$, letter $B$ can be rewritten to $C$ and then to $D$. The following derivation:

$$
A B A A \underset{r_{2}}{\longrightarrow} A C A A \underset{r_{2}}{\longrightarrow} A D A A \underset{r_{2}}{\longrightarrow} A D E A \underset{r_{2}}{\longrightarrow} A D E E
$$

is represented as follows:

$$
\begin{array}{|c|c|c|c|}
\hline \mathrm{A} & \mathrm{~B} & \mathrm{~A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{C} & \mathrm{~A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{D} & \mathrm{~A} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{D} & \mathrm{E} & \mathrm{~A} \\
\mathrm{~A} & \mathrm{D} & \mathrm{E} & \mathrm{E} \\
\hline
\end{array}
$$

We have [AAAAA]B $R_{2}[B C D D D]$ and $[B C D D D] A R_{2}[A A A E E]$ and $[A A A E E] A \quad R_{2}[A A A A E]$.
Consider a word $X_{1} \in L_{\text {Lin }}$ of length $n$ and a derivation $X_{1} \underset{r_{2}}{\longrightarrow} X_{2} \underset{r_{2}}{\longrightarrow} \underset{r_{2}}{\longrightarrow} X_{m}$ represented by the following Figure.


The transducer $T_{2}$ produces pairs corresponding to $m$ successive letters of adjacent positions: Given the $m$ successive letters at a position $i$, it yields the $m$ successive letters at position $i+1$.
For any words $X, Y \in \Sigma^{*}$ of length $n$, we denote by $X \triangle Y$ the cardinal of $\{1 \leq i \leq$ $n \mid X(i) \neq Y(i)\}$. The following technical lemma states that any two consecutive columns are recognized by $T_{2}$.
Lemma 4.2. The two following properties are equivalent:
a) $X_{1} \underset{r_{2}}{\longrightarrow} X_{2} \underset{r_{2}}{\longrightarrow} \cdots \underset{\Gamma_{2}}{\longrightarrow} X_{m}$
b) $\left[X_{1}(i-1) X_{2}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) R_{2}\left[X_{1}(i) \ldots X_{m}(i)\right]$ for all $2 \leq i \leq\left|X_{1}\right|$ and $\left|X_{j-1}\right|=\left|X_{j}\right|$ and $X_{j-1} \triangle X_{j}=1$ and $X_{j-1}(1)=X_{j}(1)$ for all $2 \leq j \leq m$.

Proof. i) By definition of $\Gamma_{2}$, we have, for all $2 \leq j \leq m$,

$$
\left|X_{j-1}\right|=\left|X_{j}\right| \quad \text { and } X_{j-1} \triangle X_{j}=1 \quad \text { and } \quad X_{j-1}(1)=X_{j}(1) .
$$

We show that

$$
\left[X_{1}(i-1) X_{2}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) \quad R_{2} \quad\left[X_{1}(i) \ldots X_{m}(i)\right]
$$

by induction on $m \geq 1$.
Basis case : $m=1$. For all $2 \leq i \leq\left|X_{1}\right|$, we have

$$
\left[X_{1}(i-1)\right] X_{1}(i) R_{2}\left[X_{1}(i)\right]
$$

considering the path

$$
I \underset{T_{T_{2}}}{\left[X_{1}(i-1) /\left[X_{1}(i)\right.\right.}\left(X_{1}(i), X_{1}(i-1), X_{1}(i)\right) \underset{T_{2}}{] X_{1}(i) /\right]} F
$$

Inductive case : $m \Longrightarrow m+1$.
Suppose the implication for a derivation of length $m$ and let $X_{1} \underset{\Gamma_{2}}{\longrightarrow} \ldots \underset{\Gamma_{2}}{\longrightarrow} X_{m} \underset{\Gamma_{2}}{\longrightarrow} X_{m+1}$.
There exists $2 \leq k \leq\left|X_{1}\right|$ such that $X_{m}(k) \neq X_{m+1}(k)$ and for all $i \neq k, X_{m}(i)=$ $X_{m+1}(i)$.
Let $2 \leq i \leq\left|X_{1}\right|$. We want to show that

$$
\left[X_{1}(i-1) \ldots X_{m}(i-1) X_{m+1}(i-1)\right] X_{1}(i) \quad R_{2} \quad\left[X_{1}(i) \ldots X_{m+1}(i)\right]
$$

By inductive hypothesis, we have

$$
\left[X_{1}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) \quad R_{2} \quad\left[X_{1}(i) \ldots X_{m}(i)\right]
$$

By definition of the transducer $T_{2}$, we have

$$
I \xrightarrow[T_{2}]{\left[X_{1}(i-1) \ldots X_{m}(i-1) /\left[X_{1}(i) \ldots X_{m}(i)\right.\right.}\left(X_{1}(i), X_{m}(i-1), X_{m}(i)\right)
$$

We distinguish the two complementary cases below.
Case 1:i$=k$. Then $X_{m}(i)=X_{m+1}(i)$ and we add an arc of type 3 .

$$
\left(X_{1}(i), X_{m}(i-1), X_{m}(i)\right)^{X_{m+1}(i-1) / X_{m+1}(i)}\left(X_{1}(i), X_{m+1}(i-1), X_{m+1}(i)\right)
$$

Case 2: $i=k$. We have the rule $X_{m}(i-1) X_{m}(i) \quad \Gamma_{2} \quad X_{m+1}(i-1) X_{m+1}(i)$.
The following arc of type 2 is associated to previous rule:

$$
\left(X_{1}(i), X_{m}(i-1), X_{m}(i)\right) \xrightarrow[T_{2}]{X_{m+1}(i-1) / X_{m+1}(i)}\left(X_{1}(i), X_{m+1}(i-1), X_{m+1}(i)\right)
$$

Finally, we add the arc leading to the final state:

$$
\left(X_{1}(i), X_{m+1}(i-1), X_{m+1}(i)\right) \xrightarrow[T_{2}]{] X_{1}(i) /\right]} F
$$

We get the result for $m+1$ and the direct implication.
ii) Conversely, we prove that $(\mathrm{b}) \Longrightarrow(\mathrm{a})$.

Suppose that $\left[X_{1}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) \quad R_{2} \quad\left[X_{1}(i) \ldots X_{m}(i)\right]$ for all $2 \leq i \leq\left|X_{1}\right|$
and $\left|X_{j-1}\right|=\left|X_{j}\right|$ and $X_{j-1} \triangle X_{j}=1$ and $X_{1}(j-1)=X_{1}(j)$ for all $2 \leq j \leq m$.
Let $2 \leq j \leq m$. Let us show that $X_{j-1} \underset{\Gamma_{2}}{\longrightarrow} X_{j}$.
As $X_{j-1} \triangle X_{j}=1$, there exists a unique $2 \leq k \leq\left|X_{1}\right|$ such that $X_{j-1}(k) \neq X_{j}(k)$.
Moreover $X_{j}(1)=X_{j}(1)$ thus $k \neq 1$ and $X_{j-1}(k-1)=X_{j}(k-1)$.
We have $\left[X_{1}(k-1) \ldots X_{m}(k-1)\right] X_{1}(k) \quad R_{2} \quad\left[X_{1}(k) \ldots X_{m}(k)\right]$.
By definition of the transducer $T_{2}$, the following arc exists

$$
\left(X_{1}(k), X_{j-1}(k-1), X_{j-1}(k)\right) \underset{T_{2}}{X_{j}(k-1) / X_{j}(k)}\left(X_{1}(k), X_{j}(k-1), X_{j}(k)\right)
$$

This arc is of type 2 and gives the existence of the following rule of $\Gamma_{2}$

$$
X_{j-1}(k-1) X_{j-1}(k) \longrightarrow X_{j}(k-1) X_{j}(k)
$$

Thus for any $2 \leq j \leq m, X_{j-1} \xrightarrow[\Gamma_{2}]{\longrightarrow} X_{j}$ i.e. a) holds.

The transducer $T_{2}$ recognizes arcs of the form $[U] A \rightarrow[A V]$. In order to create paths on the graph, we add to $T_{2}$ the set of transitions $\{F \xrightarrow{A / A} F \mid A \in \Sigma\}$. New arcs are of the form $[U] A W \rightarrow[A V] W$ where $W$ is a suffix of the initial word of the derivation. If $X_{1} \in L_{\text {Lin }}$ with $\left|X_{1}\right|=n$ and $X_{1} \frac{m-1}{\Gamma_{2}} X_{m}$, the graph $G_{\text {Lin }}$ contains the following path:

$$
\left[X_{1}(1)^{m}\right] X_{1}(2) \ldots X_{1}(n) \rightarrow\left[X_{1}(2) \ldots X_{m}(2)\right] X_{1}(3) \ldots X_{1}(n) \ldots \rightarrow\left[X_{m}(1) \ldots X_{m}(n)\right] .
$$

It remains to add arcs of the form $[U] \rightarrow \varepsilon$ for any word $U$ and to label arcs of $G$. Since $X_{m}$ is derived by $\Gamma_{1}$ in a word of $L$, the last letter of each column gives labels of arcs. Thus, we set $[U A] B W \underset{G_{L_{\text {Li }}}}{a}[B V] W$ for each $a \in \mathcal{A}$ such that $A \underset{\gamma_{1}}{\longrightarrow}$. The graph $G_{\text {Lin }}$ obtained is left-synchronized graph, and verifies that $L=L\left(G_{L i n}, L_{L i n},\{\varepsilon\}\right)$.

### 4.2 Traces from a rational set

The problem is that $L_{\text {Lin }}$ is not rational. In order to reduce $L_{\text {Lin }}$ to a rational set, we complete $T_{2}$ into a transducer generating words of $L_{\text {Lin }}$ successively from left to right. Let $G r$ be a grammar in Greibach normal form generating $L_{L i n}$ from a non-terminal $S$. Each rule of $G r$ is of the form $Z \rightarrow A W$ where $Z \in \Sigma_{r}$ is a non-terminal of $G r$, $A \in \Sigma$ is a terminal (which is also a non-terminal of $\Gamma$ ) and $W \in \Sigma_{r}^{*}$ is a non-terminal word of $G r$. Let the transducer

$$
T_{2}^{\prime}:=T_{2} \cup\left\{F \xrightarrow{Z / U} F^{\prime} \mid(Z, U) \in G r\right\} \cup\left\{F^{\prime} \xrightarrow{Z / Z} F^{\prime} \mid Z \in \Sigma_{r}\right\},
$$

where $F^{\prime}$ is a new state of the transducer. We denote by $R_{2}^{\prime}$ the relation recognized by $T_{2}^{\prime}$ from $I$ to $F^{\prime}$. This relation is still of bounded length difference. Let

$$
L_{\text {Rat }}:=\left\{\left[A^{m}\right] B W \mid S \underset{G r}{2} A B W \wedge A, B \in \Sigma \wedge W \in \Sigma_{r}^{*} \wedge m \geq 1\right\} .
$$

Let us reformulate Lemma 4.2 for derivations starting from $L_{\text {Lin }}$.
Lemma 4.3. Let $X_{1}, \ldots, X_{m} \in \Sigma^{*}$ and $n=\left|X_{1}\right|$.
The two following properties are equivalent:
a) $X_{1} \underset{r_{2}}{\longrightarrow} X_{2} \underset{r_{2}}{\longrightarrow} \cdots \xrightarrow[r_{2}]{\longrightarrow} X_{m}$ and $X_{1} \in L_{\text {Lin }}$
b) There exists $W_{1}, \ldots, W_{n-1} \in \Sigma_{r}^{*}$ such that

$$
\begin{aligned}
& {\left[X_{1}(1) \ldots X_{m}(1)\right] X_{1}(2) W_{1} \in L_{\text {Rat }} \text { and } W_{n-1}=\varepsilon} \\
& \text { and }\left[X_{1}(n-1) \ldots X_{m}(n-1)\right] X_{1}(n) \quad R_{2} \quad\left[X_{1}(n) \ldots X_{m}(n)\right] \\
& \text { and }\left[X_{1}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) W_{i-1} \quad R_{2}^{\prime} \quad\left[X_{1}(i) \ldots X_{m}(i)\right] X_{1}(i+1) W_{i}
\end{aligned}
$$

for all $2 \leq i<n$
and $\left|X_{j-1}\right|=\left|X_{j}\right|$ and $X_{j-1} \triangle X_{j}=1$ and $X_{j-1}(1)=X_{j}(1)$ for all $2 \leq j \leq m$.

Proof. i) We suppose (a) and show (b).
Since $X_{1} \in L_{\text {Lin }}$, we consider the derivation from $S$ to $X_{1}$ according to $G r$ : there exists non-terminal words $W_{1}, \ldots, W_{n-2}$ of $G r$ such that

$$
S \underset{G r}{2} X_{1}(1) X_{1}(2) W_{1} \underset{G r}{\longrightarrow} \ldots \underset{G r}{\longrightarrow} X_{1}(1) \ldots X_{1}(n-1) W_{n-2} \underset{G r}{\longrightarrow} X_{1}(1) \ldots X_{1}(n)
$$

By Lemma 4.2, we have for all $2 \leq i \leq\left|X_{1}\right|$,

$$
\left[X_{1}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) \quad R_{2} \quad\left[X_{1}(i) \ldots X_{m}(i)\right]
$$

$\left|X_{j-1}\right|=\left|X_{j}\right|, \quad X_{j-1} \triangle X_{j}=1$ and $X_{j-1}(1)=X_{j}(1)$ for all $2 \leq j \leq m$.
Let $2 \leq i \leq n-1$. We know that $W_{i}$ is obtained from $W_{i-1}$ by the rewriting of the non-terminal $W_{i-1}(1)$ :

$$
W_{i-1}=Z V \underset{G r}{\longrightarrow} U V=X_{2}(i+1) W_{i} .
$$

We complete the preceding path leading to $F$ with the arc $F \xrightarrow{Z / U} F^{\prime}$ and then with arcs $F^{\prime} \xrightarrow[T_{2}^{\prime}]{Z / Z} F^{\prime}$ for $V$. Thus, we have

$$
\left[X_{1}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) W_{i-1} \quad R_{2}^{\prime} \quad\left[X_{1}(i) \ldots X_{m}(i)\right] X_{1}(i+1) W_{i}
$$

ii) We suppose (b) and show (a).

We cut the paths

$$
\left[X_{1}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) W_{i-1} \quad R_{2}^{\prime} \quad\left[X_{1}(i) \ldots X_{m}(i)\right] X_{1}(i+1) W_{i}
$$

which become

$$
\left[X_{1}(i-1) X_{2}(i-1) \ldots X_{m}(i-1)\right] X_{1}(i) \quad R_{2} \quad\left[X_{1}(i) \ldots X_{m}(i)\right]
$$

By Lemma 4.2, we have $X_{1} \underset{\Gamma_{2}}{\longrightarrow} X_{2} \underset{\Gamma_{2}}{\longrightarrow} \underset{\Gamma_{2}}{\longrightarrow} X_{m}$.
By hypothesis $\left[X_{1}(1) \ldots X_{m}(1)\right] X_{1}(2) W_{1} \quad \in L_{\text {Rat }} \quad$ and $\quad X_{1}(1)=\ldots=X_{m}(1)$. So $S \underset{G r}{2} X_{1}(1) X_{1}(2) W_{1}$. Thus $S \underset{G r}{*} X_{1}(1) \ldots X_{1}(n)=X_{1}$ hence $X_{1} \in L_{L i n}$.

The transducer $T_{2}^{\prime}$ successively generates letters of $X_{1}$. It remains to label arcs of the recognized graph to get a left-synchronized graph such that the language of path labels leading from the rational vertex set $L_{\text {Rat }}$ to the final vertex set $\{\varepsilon\}$ is the context-sensitive language defined by $\Gamma$. As in Section 4.1, any arc of the form $[U A] B W \rightarrow[B V] W$ is labelled $a \in \mathcal{A}$ if $A \underset{\Gamma_{1}}{\longrightarrow} a$.
Proposition 4.4. Any context-sensitive language is trace of a synchronized graph.
Proof. Let $L$ be a context-sensitive language. There exists a 2 -system $\Gamma$ such that

$$
L=\left\{v \in \mathcal{A}^{*} \mid \exists u \in L_{\text {Lin }}, u \underset{\Gamma}{*} v\right\} .
$$

For all letter $a \in \mathcal{A}$, we denote by

$$
\Sigma_{a}:=\left\{A \in \Sigma \mid A \underset{\Gamma_{1}}{\longrightarrow} a\right\}
$$

the set of non-terminals generating the terminal $a$ in $\Gamma$.
We define the graph $G_{0}$ such that for any $a \in \mathcal{A}$,

$$
\underset{G_{0}}{a}:=R_{2}^{\prime} \cap\left[\Sigma^{*} \Sigma_{a}\right] \Sigma \Sigma_{r}^{*} \times\left(\left[\Sigma^{+}\right] \Sigma \Sigma_{r}^{*} \cup\left[\Sigma^{+}\right] \Sigma_{r}^{*}\right) .
$$

Since $R_{2}^{\prime}$ is a bounded length difference relation, so $G_{0}$ is by Proposition 2.8 and Theorem 2.7.
In particular, $G_{0}$ is left-synchronized and the following graph:

$$
G:=G_{0} \cup \bigcup_{a \in \mathcal{A}}\left[\Sigma^{*} \Sigma_{a}\right] \times\{a\} \times \varepsilon
$$

is also left-synchronized since $\left[\Sigma^{*} \Sigma_{a}\right] \times\{a\} \times \varepsilon$ is recognizable ([10]) for all $a \in \mathcal{A}$.
We recall that

$$
L_{\text {Rat }}:=\left\{\left[A^{m}\right] B W \mid S \underset{G r}{*} A B W \wedge m \geq 1\right\}
$$

where $S$ is the axiom of $G r$. We have

$$
\begin{aligned}
& u \in L \text { with }|u|=n>1 \\
& \Longleftrightarrow(\text { By definition) } \\
& \text { there exists } X_{1}, \ldots, X_{m} \in \Sigma^{*} \text { of length } n \text { such that } \\
& X_{1} \in L_{L i n} \text { and } X_{1} \xrightarrow[r_{2}]{\longrightarrow} X_{2} \underset{r_{2}}{\longrightarrow} \cdots X_{m} \text { and } X_{m}(i) \xrightarrow[r_{1}]{\longrightarrow} u(i) \text { for all } 1 \leq i \leq n \\
& \Longleftrightarrow \text { (by Lemma 4.3) }
\end{aligned}
$$

there exists non-terminal words $W_{1}, \ldots, W_{n-1}$ of $G r$ such that $\left[X_{1}(1) \ldots X_{m}(1)\right] X_{1}(2) W_{1} \in L_{\text {Rat }}$ and $W_{n-1}=\varepsilon$ and such that

$$
\begin{aligned}
{\left[X_{1}(1) \ldots X_{m}(1)\right] X_{1}(2) W_{1} \xrightarrow[G_{0}]{u(1)}[ } & \left.X_{1}(2) \ldots X_{m}(2)\right] X_{1}(3) W_{2} \underset{G_{0}}{u(2)} \ldots \underset{G_{0}}{u(n-1)}\left[X_{1}(n) \ldots X_{m}(n)\right] \\
& \text { and } X_{m}(n) \in \Sigma_{u(n)} \\
& \Longleftrightarrow(\text { By definition }) \\
& u \in L\left(G, L_{\text {Rat }},\{\varepsilon\}\right)
\end{aligned}
$$

Thus

$$
L=L\left(G, L_{\text {Rat }},\{\varepsilon\}\right) \cup\{u \in L| | u \mid \leq 1\}
$$

and it remains to apply Lemma 2.9.

This leads to the main result of this paper:
Theorem 4.5. Context-sensitive languages are exactly the traces of synchronized graphs between finite sets.

Proof. Any synchronized graph is a rational graph, hence any trace of a synchronized graph is a context-sensitive language by Proposition 3.4. Proposition 4.4 ensures the converse.

### 4.3 Letter to letter graphs

Using Lemma 2.9, the previous section defined Csl as traces of synchronized graphs from and to finite sets of vertices. In this section, we study traces with initial and final rational sets. Provided this extension, the traces of letter to letter graphs are Csl.
Indeed, the synchronized relation of bounded length difference $R_{2}^{\prime}$, we have used in the proof of Proposition 4.4, can be completed into a letter-to-letter relation.

Lemma 4.6. Let $R \subseteq \Sigma^{*} \times \Sigma^{*}$ be a left-synchronized relation and let $\diamond$ be a symbol such that $\diamond \notin \Sigma$. We can transform $R$ into a letter-to-letter relation $R_{l}$ such that $\forall(U, V) \in \Sigma^{*} \times \Sigma^{*}, \forall n \geq 0$,

$$
(U \underset{R}{\stackrel{n}{\longrightarrow}} V) \Longleftrightarrow\left(\exists k \geq 0, \exists k^{\prime} \geq 0 \text { such that } U \diamond^{k} \underset{R_{l}}{n} V \diamond^{k^{\prime}}\right)
$$

Proof. Let $T$ be a left-synchronized transducer recognizing $R$. The transducer $T_{l}$ is built from $T$ by replacing each arc of the form $p \xrightarrow{\epsilon / A} q$ (respectively $p \xrightarrow{A / \epsilon} q$ ) with $A \in \Sigma$ by the $\operatorname{arc} p \xrightarrow{\diamond / A} q($ respectively $p \xrightarrow{A / \diamond} q)$. Then for each final vertex $f$ of $T$, we create a new final state $f^{\prime}$ of $T_{l}$ and add the $\operatorname{arcs} f \xrightarrow{\diamond / \diamond} f^{\prime}$ and $f^{\prime} \xrightarrow{\diamond / \diamond} f^{\prime}$.

Let us reformulate Proposition 4.4.
Proposition 4.7. Any context-sensitive language is the language $L\left(G, L_{\text {Rat }}, F_{\text {Rat }}\right)$ of path labels leading from a rational set of vertices $L_{R a t}$ to another $F_{R a t}$ and where $G$ is a letter-to-letter rational graph.

Proof. Using Proposition 2.8 we get that $R_{2}^{\prime}$ is a left-synchronized relation. Let $\diamond$ be a symbol such that $\diamond \notin \Sigma \cup \Sigma_{r}$. Using Lemma 4.6, $R_{2}^{\prime}$ is completed into a letter-to-letter relation $R_{l}$. The result is obtained by adapting the proof of Proposition 4.4 with

$$
\begin{gathered}
\stackrel{a}{G_{0}}:=R_{l} \cap\left[\Sigma^{*} \Sigma_{a}\right] \Sigma \Sigma_{r}^{*} \diamond^{*} \times\left(\left[\Sigma^{+}\right] \Sigma \Sigma_{r}^{*} \diamond^{*} \cup\left[\Sigma^{+}\right] \diamond^{*}\right) \\
G:=G_{0} \cup \bigcup_{a \in \mathcal{A}}\left\{[U A] \diamond^{k} \xrightarrow{a} \$|[U A]|+k \mid U \in \Sigma^{*} \wedge A \in \Sigma_{a}\right\} \\
L_{R a t}:=\left\{\left[A^{m}\right] B W \diamond^{k} \mid S \xrightarrow[G r]{2} A B W \wedge m \geq 1 \wedge k \geq 0\right\}
\end{gathered}
$$

and

$$
F_{R a t}:=\$^{+}
$$

## 5 Conclusion

In this paper, we have established a connection between context-sensitive languages and rational graphs. We have been able to prove that the traces of these graphs are contextsensitive languages, and that the context-sensitive languages are traces of letter-to-letter rational graphs with initial and final rational sets. The proof of the latter result relies on the Penttonen normal form for context-sensitive languages, it is indeed possible to avoid the use of this form: this has been done by Carayol [2] and Meyer [13], those proofs adapt our construction to produce a rational graph from a linearly bounded Turing machine. Our result might give an interesting approach to Kuroda's conjecture [12]: do the deterministic context-sensitive languages (i.e., generated using a deterministic LBM) coincide with context-sensitive languages? An easier question would be to characterize the traces of deterministic rational graphs. This question is still unsolved.

## References

1. A. Blumensath and E. Grädel. Automatic Structures. In Proceedings of 15 th IEEE Symposium on Logic in Computer Science LICS 2000, pages 51-62, 2000.
2. A. Carayol. Notions of determinism for rational graphs. Private communication, 2001.
3. D. Caucal. On the transition graphs of Turing machines. Theoretical Computer Science, 296(2):195223, 2003.
4. D. Caucal. On transition graphs having a decidable monadic theory. Theoretical Computer Science, 290(1):79-115, 2003.
5. N. Chomsky. Three models for the description of language. IRE Transactions on Information Theory, 2(3):113-124, 1956.
6. T. Colcombet. On families of graphs having a decidable first order theory with reachability. In Proceedings of the 29th International Conference on Automata, Languages, and Programming, volume 2380 of Lecture Notes in Computer Science. Springer-Verlag, July 2002.
7. B. Courcelle. Handbook of Theoretical Computer Science, chapter Graph rewriting: an algebraic and logic approach. Elsevier, 1990.
8. C. Elgot and J. Mezei. On relations defined by generalized finite automata. IBM, 9:47-68, 1965.
9. D. Epstein, J.W. Cannon, D.F. Holt, S.V.F. Levy, M.S. Paterson, and Thurston. Word processing in groups. Jones and Barlett publishers, 1992.
10. C. Frougny and J. Sakarovitch. Synchronized rational relations of finite and infinite words. Theoretical Computer Science, 108:45-82, 1993.
11. S.C. Kleene. Representation of events in nerve nets and finite automata. In C.E. Shannon and J. McCarthy, editors, Automata Studies, pages 3-40, 1956. Princeton.
12. S.-Y. Kuroda. Classes of languages and linear-bounded automata. Information and Control, 7(2):207223, June 1964.
13. A. Meyer. Sur des sous-familles de graphes rationnels. Rapport de DEA, Université de Rennes $1,2002$.
14. C. Morvan. On rational graphs. In J. Tiuryn, editor, Fossacs 00, volume 1784 of $L N C S$, pages 252-266, 2000. ETAPS 2000 best theoretical paper Award.
15. C. Morvan and C. Stirling. Rational graphs trace context-sensitive languages. In A. Pultr and J. Sgall, editors, MFCS 01, volume 2136 of $L N C S$, pages 548-559, 2001.
16. D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logic. Theoretical Computer Science, 37:51-75, 1985.
17. M. Penttonen. One-sided and two-sided context in formal grammars. Information and Control, 25(4):371-392, 1974.
18. C. Rispal. Graphes rationnels synchronisés. Rapport de DEA, Université de Rennes 1, 2001.
19. C. Rispal. Synchronized graphs trace the context-sensitive languages. In R. Mayr A. Kucera, editor, Infinity 02, volume 68 of ENTCS, 2002.
20. S. Greibach S. Ginsburg. Abstract families of languages. Mem. Am. Math. Soc., 87, 1969.
21. G. Sénizergues. Definability in weak monadic second-order logic of some infinite graphs. In Dagstuhl seminar on Automata theory: Infinite computations, Warden, Germany, volume 28, page 16, 1992.
22. T. Urvoy. Abstract families of graphs. In M. Toyama M. Ito, editor, DLT 02, volume 2450 of LNCS, pages 381-392, 2002.
