View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

ELSEVIER

Theoretical Computer Science 296 (2003) 195–223

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the transition graphs of turing machines

Didier Caucal

IRISA-CNRS, Campus de Beaulieu, F-35042 Rennes, France

Abstract

As for pushdown automata, we consider labelled Turing machines with ε-rules. With any Turing machine M and with a rational set C of configurations, we associate the restriction to C of the ε-closure of the transition set of M. We get the same family of graphs by using the labelled word rewriting systems. We show that this family is the set of graphs obtained from the binary tree by applying an inverse mapping into F followed by a rational restriction, where F is any family of recursively enumerable languages containing the rational closure of all linear languages. We show also that this family is obtained from the rational graphs by inverse rational mappings. Finally we show that this family is also the set of graphs recognized by (unlabelled) Turing machines with labelled final states, and even if we restrict to deterministic Turing machines.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Formal languages; Infinite graphs

1. Introduction

The transition graphs of some classes of machines have already been investigated. First, Muller and Schupp have considered rational restrictions of the transition graphs of pushdown automata: these graphs are the graphs of bounded degree having a finite number of nonisomorphic connected components when decomposed by distance from any vertex; these graphs have a decidable monadic theory [22]. This graph family is also the set of rational restrictions of the prefix transition graphs of finite labelled word rewriting systems [9]. Extending to recognizable labelled rewriting systems, the rational restrictions of their prefix transition graphs define a larger family of graphs having a decidable monadic theory [11]. To extend this last family, Morvan has defined the family of rational graphs, which are the graphs recognized by transducers with labelled

E-mail address: caucal@irisa.fr (D. Caucal).
URL: http://www.irisa.fr/galion/caucal/

0304-3975/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(02)00655-2

final states [20]. This family is general: it contains for instance the transition graphs of Petri nets, the transition graphs of congruential systems [25], and the transition graphs of labelled word rewriting systems. Although the rational graphs are recursive, there exists a rational graph with an undecidable first order theory.

There is a simple and uniform way to present all the previous families of graphs from families of languages. Take a family F of languages, a set T of labels and a mapping $h : T \to F$ associating to each label a language in F. The inverse image $h^{-1}(G)$ of a graph G by h has an arc $s \xrightarrow{a} t$ when there is a path $s \overset{u}{\Rightarrow} t$ in G for some word u in $h(a)$. A family F of languages induces a family REC_F of graphs obtained from the infinite binary tree by marking rationally some vertices (with a special letter) and then applying an inverse mapping. Then the family of rational restrictions of the transition graphs of pushdown automata is the family REC_{Fin} of graphs induced by the family *Fin* of finite languages. Furthermore, the family of rational restrictions of the prefix transition graphs of recognizable rewriting systems is the family REC_{Rat} of graphs induced by the family *Rat* of rational languages [11]. Finally, the family of rational graphs is the family $REC_{\bar{\mathrm{L}}\mathrm{in}}$ of graphs induced by a subfamily $\bar{L}in$ of linear languages [20].

Thus small families of languages induce large families of graphs. Conversely, a family of graphs yields a family of languages. A trace of a graph is the language of path labels from and to given finite vertex sets. The traces of finite graphs are the rational languages. The traces of graphs in REC_{Fin} are the context-free languages which are also the traces of graphs in REC_{Rat}. Finally, the traces of rational graphs are the context-sensitive languages [21].

Following the Chomsky hierarchy, we present a general family of graphs, the traces of which are the recursively enumerable languages. We consider the off-line Turing machines [19] with a read only one way input tape and a unique two ways working tape. These machines are particular labelled word rewriting systems allowing rules labelled by ε. Following [23] and as for prefix transition graphs of word rewriting systems, we consider rational restrictions of the ε-closure of transition graphs of these off-line Turing machines. We show that this family of graphs coincides with the family of rational restrictions of the ε-closure of the transition graphs of word rewriting systems. We also show that this graph family is equal to $REC_{\mathrm{Rat}(\bar{\mathrm{L}}\mathrm{in})} = REC_{\mathrm{RE}}$ meaning that it is induced by any language family between the rational closure of $\bar{L}in$ and the family *RE* of recursively enumerable languages. Furthermore, we show that this graph family is obtained by inverse rational mappings of rational graphs. Finally and as for the rational graphs recognized by transducers with labelled final states, we show that our family is the set of graphs recognized by (usual) nondeterministic Turing machines with labelled final states, and this result remains true if we restrict to deterministic Turing machines.

2. Preliminaries on graphs

Let P be a subset of a monoid M, and $Id_P = \{(u,u) \mid u \in P\}$ the *identity* relation on P. A (simple oriented labelled) *P-graph* G is a subset of $V \times P \times V$ where V

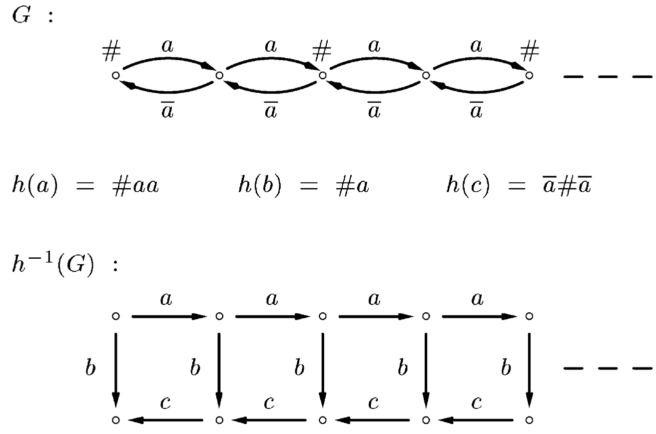


Fig. 1. An inverse mapping of a marked graph.

is an arbitrary set. Any (s,a,t) of G is a *labelled arc* of *source*1 s, of *target* t, with *label* a, and is identified with the labelled transition $s \underset{G}{\overset{a}{\longrightarrow}} t$ or directly $s \overset{a}{\longrightarrow} t$ if G is understood. We denote by $V_G := \{s \mid \exists a\ \exists t,\ s \overset{a}{\longrightarrow} t \vee t \overset{a}{\longrightarrow} s\}$ the *vertex* set of G. The set $2^{V \times P^* \times V}$ of P^*-graphs with vertices in V is a monoid for the *composition* $G_\circ H := \{r \overset{a \cdot b}{\longrightarrow} t \mid \exists s,\ r \underset{G}{\overset{a}{\longrightarrow}} s \wedge s \underset{H}{\overset{b}{\longrightarrow}} t\}$ for any $G, H \subseteq V \times P^* \times V$, where $\{s \overset{1}{\longrightarrow} s \mid s \in V\}$ is its neutral element. The submonoid $\{G\}^*$ of $2^{V_G \times P^* \times V_G}$ generated by any graph G gives by union the graph $G^* := \bigcup \{G\}^*$. The relation $\underset{G^*}{\overset{u}{\longrightarrow}}$ denoted by $\underset{G}{\overset{u}{\Longrightarrow}}$ or simply by $\overset{u}{\Longrightarrow}$ if G is understood, is the existence of a *path* in G labelled $u \in P^*$. For every $Q \subseteq P^*$, we write $s \overset{Q}{\Longrightarrow} t$ if there is some $u \in Q$ such that $s \overset{u}{\Longrightarrow} t$. The *restriction* $G_{|C}$ of a P-graph G to an arbitrary set C is $G_{|C} := G \cap (C \times P \times C)$. The labels $\mathscr{L}(G,E,F)$ of paths of G from a set E to a set F is the set $\mathscr{L}(G,E,F) := \{u \in M \mid \exists s \in E,\ \exists t \in F,\ s \underset{G}{\overset{u}{\Longrightarrow}} t\}$. A *trace* of a graph G is the language $\mathscr{L}(G,E,F)$ of path labels from a finite set E to a finite set F.

Given an alphabet T and a relation $h \subseteq T \times P$, the *inverse* $h^{-1}(G)$ by h of any P-graph G is the following T-graph:

$$h^{-1}(G) := \{s \overset{a}{\longrightarrow} t \mid a \in T \wedge \exists u \in h(a),\ s \underset{G}{\overset{u}{\Longrightarrow}} t\}$$

An example is given Fig. 1.

A relation $h \subseteq T \times P$ can be seen as a mapping from T into 2^P associating the image $h(a)$ of any $a \in T$. When $P = S^*$ for some alphabet S, such a mapping is extended by morphism to a *substitution* from T^* into S^* i.e. a mapping h from T^* into 2^{S^*} such that $h(\varepsilon) = \{\varepsilon\}$ and $h(uv) = h(u)h(v)$ for every $u, v \in T^*$. The composition of functions is the composition of their relations: $(g \circ h)(a) = h(g(a))$. Let us give basic properties of inverse mappings.

Lemma 2.1. *Let O, P, Q be alphabets and G be a Q-graph.*
Let $h \subseteq P \times Q^*$ and $g \subseteq O \times P^*$ extended by substitution. We have

(a) $s \underset{h^{-1}(G)}{\overset{u}{\Rightarrow}} t \Leftrightarrow s \underset{G}{\overset{h(u)}{\Rightarrow}} t$ *for any* $u \in P^+$ *and* $s, t \in V_G$

(b) $\mathscr{L}(h^{-1}(G), E, F) = h^{-1}(\mathscr{L}(G, E, F))$ *for any* $E, F \subseteq V_{h^{-1}(G)}$

(c)
$$g^{-1}(h^{-1}(G)) = ((g \circ h)^{-1}(G))_{|V_{h^{-1}(G)}}$$
$$\text{and } g^{-1}(h^{-1}(G)) = (g \circ h)^{-1}(G) \qquad if \ \varepsilon \notin g(O).$$

Proof. (i) We prove (a) by induction on the length of any word $u \in P^+$.
$u = a \in P$: $\forall s, t \in V_G$ $s \underset{h^{-1}(G)}{\overset{a}{\Rightarrow}} t \Leftrightarrow s \underset{h^{-1}(G)}{\overset{a}{\rightarrow}} t \Leftrightarrow s \underset{G}{\overset{h(a)}{\Rightarrow}} t$.
$u = vw$ with $v, w \in P^+$: for any $s, t \in V_G$, we have

$$\begin{aligned} s \underset{h^{-1}(G)}{\overset{vw}{\Rightarrow}} t &\Leftrightarrow \exists r,\ s \underset{h^{-1}(G)}{\overset{v}{\Rightarrow}} r \underset{h^{-1}(G)}{\overset{w}{\Rightarrow}} t \\ &\Leftrightarrow \exists r,\ s \underset{G}{\overset{h(v)}{\Rightarrow}} r \underset{G}{\overset{h(w)}{\Rightarrow}} t && \text{by induction hypothesis} \\ &\Leftrightarrow s \underset{G}{\overset{h(v)h(w)}{\Rightarrow}} t && G \text{ is a } Q\text{-graph} \\ &\Leftrightarrow s \underset{G}{\overset{h(vw)}{\Rightarrow}} t && h \text{ is a substitution.} \end{aligned}$$

(ii) Let us prove (b). By restricting (a) to vertices of $h^{-1}(G)$, we can extend it for $u = \varepsilon$:

$$s \underset{h^{-1}(G)}{\overset{u}{\Rightarrow}} t \Leftrightarrow s \underset{G}{\overset{h(u)}{\Rightarrow}} t \quad \text{for any } u \in P^* \text{ and } s, t \in V_{h^{-1}(G)}. \tag{1}$$

Taking vertex sets E and F of $h^{-1}(G)$, we have

$$\begin{aligned} \mathscr{L}(h^{-1}(G), E, F) &= \{u \in P^* \mid E \underset{h^{-1}(G)}{\overset{u}{\Rightarrow}} F\} \\ &= \{u \in P^* \mid E \underset{G}{\overset{h(u)}{\Rightarrow}} F\} && \text{by (1)} \\ &= \{u \in P^* \mid h(u) \in \mathscr{L}(G, E, F)\} \\ &= h^{-1}(\mathscr{L}(G, E, F)). \end{aligned}$$

(iii) Let us prove (c). Let $a \in O$ and $s, t \in V_G$.
Assume that $\varepsilon \notin g(O) \vee s, t \in V_{h^{-1}(G)}$. We have

$$\begin{aligned} s \underset{g^{-1}(h^{-1}(G))}{\overset{a}{\rightarrow}} t \Leftrightarrow &\quad s \underset{h^{-1}(G)}{\overset{g(a)}{\Rightarrow}} t \text{ by definition} \\ \Leftrightarrow &\quad s \underset{G}{\overset{h(g(a))}{\Rightarrow}} t \text{ by (i) if } \varepsilon \notin g(a) \text{ or by (1) if } s, t \in V_{h^{-1}(G)} \end{aligned}$$

$$\Leftrightarrow \quad s \overset{(g \circ h)(a)}{\underset{G}{\Rightarrow}} t$$

$$\Leftrightarrow \quad s \underset{(g \circ h)^{-1}(G)}{\overset{a}{\longrightarrow}} t \text{ by definition.} \qquad \square$$

Note that the vertex restriction of Lemma 2.1 (c) is necessary. It is sufficient to take $G = \{1 \xrightarrow{a} 1, 2 \xrightarrow{b} 2\}$ and the partial morphisms $h(a) = a$ and $g(a) = \varepsilon$ in such a way that $h^{-1}(G) = g^{-1}(h^{-1}(G)) = \{1 \xrightarrow{a} 1\}$ and $(g \circ h)^{-1}(G) = \{1 \xrightarrow{a} 1, 2 \xrightarrow{a} 2\}$.

Another basic property is the commutation between the inverse mapping and the restriction to particular sets. A set C is *stable* in a graph G when any path between vertices in C contains only vertices in C:

$$s_0 \underset{G}{\longrightarrow} s_1 \ldots s_{n-1} \underset{G}{\longrightarrow} s_n \wedge s_0, s_n \in C \Rightarrow s_1, \ldots, s_{n-1} \in C.$$

The restriction to any stable set commutes with any inverse mapping.

Lemma 2.2. *For any stable set C in a P-graph and any mapping h into 2^P, we have $h^{-1}(G_{|C}) = (h^{-1}(G))_{|C}$.*

Proof. We have $h^{-1}(G_{|C}) = (h^{-1}(G_{|C}))_{|C} \subseteq (h^{-1}(G))_{|C}$.

Let us verify the inverse inclusion.

Let $s \underset{(h^{-1}(G))_{|C}}{\overset{a}{\longrightarrow}} t$. So $s \underset{h^{-1}(G)}{\overset{a}{\longrightarrow}} t$ with $s, t \in C$.

By definition of $h^{-1}(G)$, there is $u \in h(a)$ such that $s \overset{u}{\underset{G}{\Rightarrow}} t$.

As $s, t \in C$ and C is stable in G, we have $s \overset{u}{\underset{G_{|C}}{\Rightarrow}} t$. Hence $s \underset{h^{-1}(G_{|C})}{\overset{a}{\longrightarrow}} t$. $\square$

Another way to express a restriction of an inverse mapping of a graph is to use a marking of the graph. The *marking* $\#_C(G)$ on a vertex set C of a graph G by a symbol $\#$ is the graph:

$$\#_C(G) := G \cup \{s \xrightarrow{\#} s \mid s \in C\}$$

obtained from G by adding $\#$ to any vertex in C. Let us give an example.

Any restriction of an inverse of a graph is an inverse of a marking of the graph.

Lemma 2.3. *We have $(h^{-1}(G))_{|C} = g^{-1}(\#_C(G))$ with $g(a) = \#h(a)\#$.*

Proof. Assuming $\#$ is a new symbol and by definition of g^{-1}, we have

$$\begin{aligned} g^{-1}(\#_C(G)) &= \{s \xrightarrow{a} t \mid \exists u \in h(a),\ s \overset{\#u\#}{\underset{\#_C(G)}{\Rightarrow}} t\} \\ &= \{s \xrightarrow{a} t \mid \exists u \in h(a),\ s \overset{u}{\underset{G}{\Rightarrow}} t \wedge s,\ t \in C\} \\ &= (h^{-1}(G))_{|C}. \qquad \square \end{aligned}$$

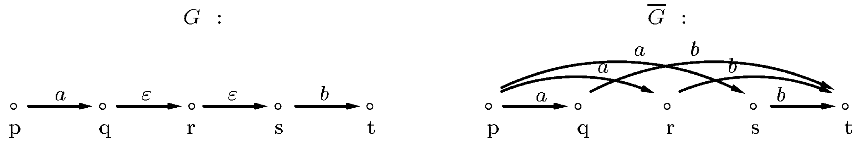


Fig. 2. ε-closure of a graph.

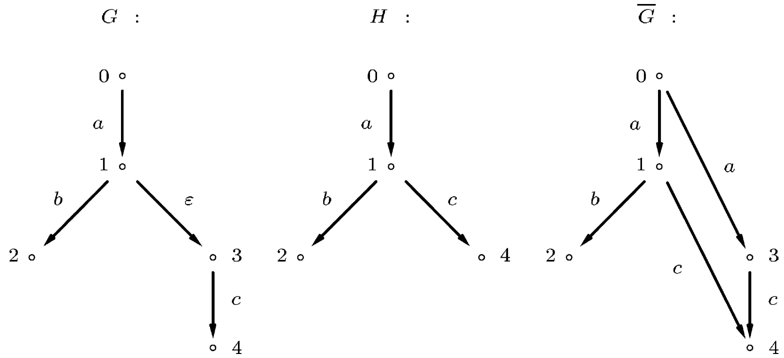


Fig. 3. Id_{V_H} is a partial weak isomorphism between G and H which are not (total) weak isomorphic.

For ε the neutral element of the free monoid T^* generated by an alphabet T, the *ε-closure* $\bar{G}$ of any $T \cup \{\varepsilon\}$-graph G is obtained by removing the ε-transitions and by adding T-transitions as follows:

$$\bar{G} := (Id_T)^{-1}(G) = \{s \xrightarrow{a} t \mid a \in T \wedge s \underset{G}{\overset{a}{\Rightarrow}} t\}.$$

We give the ε-closure of a graph in Fig. 2.

We compare graphs by isomorphism. A *partial isomorphism* from a graph G into a graph H is an injective function such that $s \underset{G}{\overset{a}{\rightarrow}} t \Leftrightarrow h(s) \underset{H}{\overset{a}{\rightarrow}} h(t)$. An *isomorphism* is a partial isomorphism such that $V_G \subseteq Dom(h)$ and $V_H \subseteq Im(h)$. A partial isomorphism on $T \cup \{\varepsilon\}$-graphs considers the ε label as a new letter. To take an ε-transition as an internal (silent) move, we compare $T \cup \{\varepsilon\}$-graphs by partial weak isomorphism. A *partial weak isomorphism* from a graph G into a graph H is an injective function such that $s \underset{G}{\overset{a}{\Rightarrow}} t \Leftrightarrow h(s) \underset{H}{\overset{a}{\Rightarrow}} h(t)$. A *weak isomorphism* is a partial weak isomorphism such that $V_G \subseteq Dom(h)$ and $V_H \subseteq Im(h)$. Note that h is a (resp. partial) weak isomorphism from G into H if and only if h is a (resp. partial) isomorphism from $\bar{G}$ into $\bar{H}$. In particular, the identity Id_{V_G} is a weak isomorphism between a graph G and $\bar{G}$. The notion of weak isomorphism is illustrated in Fig. 3.

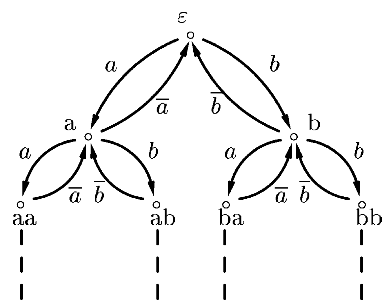
3. Classes of graphs

A general way to define a family REC_F of graphs from a family F of languages is to take the set of graphs obtained from the binary tree (with inverse transitions) by applying an inverse F-mapping followed by a rational restriction [11]. An equivalent way is to take the set of graphs obtained from the binary tree by marking a rational vertex set followed by an inverse F-mapping (Proposition 3.5). We deduce known results on the family REC_{Fin} (Theorem 3.6), on the family REC_{Rat} (Theorem 3.7), on the family $REC_{\bar{\mathrm{L}}\mathrm{in}}$ where $\bar{L}in$ is a subfamily of linear languages (Theorem 3.10). We deduce also closure properties by inverse mappings.

Let N be an alphabet containing at least two letters. We take a new alphabet $\bar{N} := \{\bar{a} \mid a \in N\}$ in bijection with N. We define the following *Dyck graph*:

$$\Lambda_N := \{u \xrightarrow{a} ua \mid u \in N^* \wedge a \in N\} \cup \{ua \xrightarrow{\bar{a}} u \mid u \in N^* \wedge a \in N\},$$

where a representation for $N = \{a, b\}$ is the following:



When L is rational, we say that $\#_L(\Lambda_N)$ is a *rational marking* of Λ_N. Note that

$$x \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{u}{\Rightarrow}} y \quad \Rightarrow \quad wx \underset{\#_L(\Lambda_N)}{\Rightarrow} wy.$$

We denote $\mathscr{L}(\#_L(\Lambda_N)) := \mathscr{L}(\#_L(\Lambda_N), \varepsilon, N^*)$ the language of path labels of Λ_N from ε to all the vertices. When L is rational, $\mathscr{L}(\#_L(\Lambda_N))$ is a context-free language. Precisely and from a deterministic and complete minimal automaton (G, ι, F) recognizing L, the language $\mathscr{L}(\#_L(\Lambda_N))$ is generated from axiom ι by the following context-free grammar:

$$\{(p, aq\bar{a}p) \mid p \xrightarrow[G]{a} q\} \cup \{(p, aq) \mid p \xrightarrow[G]{a} q\} \cup \{(p, \varepsilon) \mid p \in V_G\} \cup \{(p, \#p) \mid p \in F\}.$$

The binary relation

$$J := \{(a\bar{a}, \varepsilon) \mid a \in N\} \cup \{(\#, \varepsilon)\}$$

has a canonical rewriting, and we write $u{\downarrow}J$ the irreducible word (normal form) that derives from u according to J. We have for any word

$u \in \mathscr{L}(\#_L(\Lambda_N))$,

$$u \downarrow J \in N^* \wedge \varepsilon \underset{\#_L(\Lambda_N)}{\overset{u}{\Rightarrow}} u \downarrow J.$$

The reduction to the set of normal forms preserves the rationality.

Lemma 3.1. *Let* $L \in Rat(N^*)$ *and* $M \in Rat((N \cup \bar{N} \cup \{\#\})^*)$.
We have in an effective way $(\mathscr{L}(\#_L(\Lambda_N)) \cap M){\downarrow}J \in Rat(N^*)$.

Proof. Let $\bar{M} = (\mathscr{L}(\#_L(\Lambda_N)) \cap M){\downarrow}J$.
Let (G, i, F) be a finite $(N \cup \bar{N} \cup \{\#\})$-automaton recognizing M.
We color any vertex $u \in N^*$ of Λ_N by the set $c(u)$ of states p such that (p, u) is a vertex of the product $G \times \#_L(\Lambda_N)$ accessible from (i, ε):

$$c(u) := \{ p \mid \mathscr{L}(G, i, p) \cap \mathscr{L}(\#_L(\Lambda_N), \varepsilon, u) \neq \emptyset \}.$$

So $\bar{M} = \{u \in N^* \mid c(u) \cap F \neq \emptyset\}$.
We have to show that $\bar{M}$ is rational by proving that c is a regular coloring of $\#_L(\Lambda_N)$.
We consider the following equivalence $\equiv$ on N^*:

$$u \equiv v \quad \text{if} \quad c(u) = c(v) \wedge u^{-1}L = v^{-1}L.$$

As $Im(c)$ is finite and L is rational, the equivalence $\equiv$ is of finite index.
This equivalence is right regular: let $u \equiv v$ and $a \in N$. We have to show that $ua \equiv va$.
As usual $(ua)^{-1}L = a^{-1}(u^{-1}L) = a^{-1}(v^{-1}L) = (va)^{-1}L$.
By symmetry of u and v, it remains to verify that $c(ua) \subseteq c(va)$.
Let $p \in c(ua)$. There is $w \in (N \cup \bar{N} \cup \{\#\})^*$ such that

$$i \underset{G}{\overset{w}{\Rightarrow}} p \wedge \varepsilon \underset{\#_L(\Lambda_N)}{\overset{w}{\Rightarrow}} ua.$$

Let $xy = w$ such that $\varepsilon \underset{\#_L(\Lambda_N)}{\overset{x}{\Rightarrow}} u \underset{\#_L(\Lambda_N)}{\overset{y}{\Rightarrow}} ua$ and $|x|$ is maximal.
By maximality of $|x|$, we have $\varepsilon \underset{\#_{u^{-1}L}(\Lambda_N)}{\overset{y}{\Rightarrow}} a$.
As $u^{-1}L = v^{-1}L$, we have $\varepsilon \underset{\#_{v^{-1}L}(\Lambda_N)}{\overset{y}{\Rightarrow}} a$ hence $v \underset{\#_L(\Lambda_N)}{\overset{y}{\Rightarrow}} va$.
There is q such that $i \underset{G}{\overset{x}{\Rightarrow}} q \underset{G}{\overset{y}{\Rightarrow}} p$. So $q \in c(u) = c(v)$.
By definition of $c(v)$, there is $z \in (N \cup \bar{N} \cup \{\#\})^*$ such that

$$i \underset{G}{\overset{z}{\Rightarrow}} q \wedge \varepsilon \underset{\#_L(\Lambda_N)}{\overset{z}{\Rightarrow}} v$$

So $i \underset{G}{\overset{zy}{\Rightarrow}} p \wedge \varepsilon \underset{\#_L(\Lambda_N)}{\overset{zy}{\Rightarrow}} va$. Hence $p \in c(va)$.
So $H = \{[u] \xrightarrow{a} [ua] \mid u \in N^* \wedge a \in N\}$ is finite.

Furthermore $\bar{M} = \mathscr{L}(H, [\varepsilon], \{[u] \mid c(u) \cap F \neq \emptyset\})$ is rational.

Let us see that H can be constructed from (G, i, F) and a finite N-automaton $(\bar{G}, \bar{\imath}, \bar{F})$ recognizing L. We may assume that $(\bar{G}, \bar{\imath}, \bar{F})$ is deterministic, complete and minimal. This automaton is isomorphic to the complete (left) residual automaton of L:

$$(\{u^{-1}L \xrightarrow{a} (ua)^{-1}L \mid u \in N^* \wedge a \in N\}, L, \{u^{-1}L \mid \varepsilon \in u^{-1}L\}).$$

We denote by $\bar{\imath}{\cdot}u$ the unique state accessible from $\bar{\imath}$ by the path labelled by u i.e. $\bar{\imath} \underset{\bar{G}}{\overset{u}{\Rightarrow}} \bar{\imath}{\cdot}u$. So $u^{-1}L = v^{-1}L \Leftrightarrow \bar{\imath}{\cdot}u = \bar{\imath}{\cdot}v$.

It remains to show that $c(u)$ can be constructed for any $u \in N^*$.
Note that $\mathscr{L}(\#_\emptyset(\Lambda_N), \varepsilon, \varepsilon)$ is the context-free language $D_N'^*$. More generally, we will verify that $\mathscr{L}(\#_L(\Lambda_N), \varepsilon, u)$ is an effective context-free language, hence $c(u)$ is computable because the intersection of a rational language with a context-free language is (in an effective way) a context-free language, and the emptyness of a context-free language is decidable (see for instance [4]).

We define the following context-free grammar:

$$K := \{(p, aq\bar{a}p) \mid p \underset{\bar{G}}{\overset{a}{\rightarrow}} q\} \cup \{(p, \varepsilon) \mid p \in V_{\bar{G}}\} \cup \{(p, \#p) \mid p \in \bar{F}\}.$$

Then $\mathscr{L}(\#_L(\Lambda_N), \varepsilon, u) = \mathscr{L}(K, \bar{\imath}u(1)[\bar{\imath}{\cdot}u(1)] \ldots u(|u|)[\bar{\imath}{\cdot}u])$. $\square$

Lemma 3.1 with $L = \emptyset$ means that any rational language $M \in Rat((N \cup \bar{N})^*)$ to mark Λ_N, can be transformed into the following rational language over N:

$$(\mathscr{L}(\#_\emptyset(\Lambda_N)) \cap M) \downarrow J = M \downarrow I \cap N^*,$$

where $I = \{(a\bar{a}, \varepsilon) \mid a \in N\}$. It is a 'half form' of the standard Benois' lemma [3].

We denote $\tilde{u}$ the *mirror* of any word $u : \tilde{\varepsilon} = \varepsilon$ and $\widetilde{au} = \tilde{u}a$ for any letter a. The mapping $\bar{\ }$ associating to any $a \in N$ its barred letter $\bar{a} \in \bar{N}$ is extended by morphism to any word in $(N \cup \bar{N} \cup \{\#\})^*$ by defining $\bar{\#} = \#$ and $\bar{\bar{a}} = a$ for any $a \in N$. In this way, we have

$$s \underset{\#_L(\Lambda_N)}{\overset{u}{\Rightarrow}} t \quad \Leftrightarrow \quad t \underset{\#_L(\Lambda_N)}{\overset{\bar{\tilde{u}}}{\Rightarrow}} s.$$

Note also that $\bar{\tilde{u}} = \tilde{\bar{u}}$ and $(\bar{\tilde{u}}){\downarrow}J = \widetilde{\overline{u{\downarrow}J}}$ for any $u \in (N \cup \bar{N} \cup \{\#\})^*$.

Any inverse mapping of any marked Λ_N can be expressed in a suffix way.

Proposition 3.2. *For any mapping $h \subseteq T \times (N \cup \bar{N} \cup \{\#\})^*$ and any language $L \subseteq N^*$, the T-graph $h^{-1}(\#_L(\Lambda_N))$ is equal to*

$$\{w(\widetilde{\overline{u{\downarrow}J}}) \xrightarrow{a} w(v \downarrow J) \mid uv \in h(a) \wedge \bar{\tilde{u}},\ v \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N)) \wedge w \in N^*\}.$$

Proof. Let G be $\{w(\widetilde{\overline{u{\downarrow}J}}) \xrightarrow{a} w(v{\downarrow}J) \mid uv \in h(a) \wedge \bar{\tilde{u}}, v \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N)) \wedge w \in N^*\}$.

We have to show that $h^{-1}(\#_L(\Lambda_N)) = G$.

(i) Proof of $h^{-1}(\#_L(\Lambda_N)) \subseteq G$. Let $s \underset{h^{-1}(\#_L(\Lambda_N))}{\overset{a}{\longrightarrow}} t$.

There is $z \in h(a)$ such that $s \underset{\#_L(\Lambda_N)}{\overset{z}{\Longrightarrow}} t$.

Let $w \in N^*$ of minimal length such that $s \underset{\#_L(\Lambda_N)}{\overset{u}{\Longrightarrow}} w \underset{\#_L(\Lambda_N)}{\overset{v}{\Longrightarrow}} t$ with $uv = z$ meaning that w is the vertex of the path $s \underset{\#_L(\Lambda_N)}{\overset{z}{\Longrightarrow}} t$ closest to ε. Note that we can have several choices of (u, v). We have $uv \in h(a)$.

There are $x, y \in N^*$ such that $s = wx$ and $t = wy$ with $x \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{u}{\Longrightarrow}} \varepsilon \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{v}{\Longrightarrow}} y$.

So $\varepsilon \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{\tilde{\bar{u}}}{\Longrightarrow}} x$ hence $\tilde{\bar{u}} \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N))$ and $x = (\tilde{\bar{u}}){\downarrow}J = \widetilde{u{\downarrow}J}$.

Similarly $v \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N))$ and $y = v{\downarrow}J$.

Thus $s = wx = w\widetilde{u{\downarrow}J}$ and $t = wy = w(v{\downarrow}J)$. Finally $s \underset{G}{\overset{a}{\longrightarrow}} t$.

(ii) Proof of $G \subseteq h^{-1}(\#_L(\Lambda_N))$. Let $s \underset{G}{\overset{a}{\longrightarrow}} t$. There are $uv \in h(a)$ and $w \in N^*$ such that $\tilde{\bar{u}}, v \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N))$, $s = w(\widetilde{u{\downarrow}J})$, $t = w(v{\downarrow}J)$.

We have to show that $s \underset{h^{-1}(\#_L(\Lambda_N))}{\overset{a}{\longrightarrow}} t$.

As $\tilde{\bar{u}} \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N))$, we have $\varepsilon \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{\tilde{\bar{u}}}{\Longrightarrow}} \tilde{\bar{u}}{\downarrow}J = \widetilde{u{\downarrow}J}$.

So $\widetilde{u{\downarrow}J} \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{u}{\Longrightarrow}} \varepsilon$ hence $s = w(\widetilde{u{\downarrow}J}) \underset{\#_L(\Lambda_N)}{\overset{u}{\Longrightarrow}} w$.

As $v \in \mathscr{L}(\#_{w^{-1}L}(\Lambda_N))$, we have $\varepsilon \underset{\#_{w^{-1}L}(\Lambda_N)}{\overset{v}{\Longrightarrow}} v{\downarrow}J$ hence $w \underset{\#_L(\Lambda_N)}{\overset{v}{\Longrightarrow}} w(v{\downarrow}J) = t$.

Thus $s \underset{\#_L(\Lambda_N)}{\overset{uv}{\Longrightarrow}} t$ hence $s \underset{\#_L(\Lambda_N)}{\overset{h(a)}{\Longrightarrow}} t$ i.e. $s \underset{h^{-1}(\#_L(\Lambda_N))}{\overset{a}{\longrightarrow}} t$. $\square$

To give a simple form of Proposition 3.2 for any rational marking, we introduce some notations. The right concatenation of a graph $G \subseteq N^* \times T \times N^*$ by a language $L \subseteq N^*$ is the following graph:

$$G.L := \{uw \overset{a}{\longrightarrow} vw \mid u \underset{G}{\overset{a}{\longrightarrow}} v \wedge w \in L\}.$$

Similarly, we define the left concatenation $L.G$ of a graph G by a language L. A usual and simple fact is that for any $L \in Rat(N^*)$ and any $u \in N^*$, the language $[u]_L := \{v \mid v^{-1}L = u^{-1}L\} \in Rat(N^*)$ and the family $[L] := \{[u]_L \mid u \in N^*\}$ is finite. Note that for any $W \in [L]$ and any $w \in W$, $W^{-1}L = w^{-1}L$.

From Proposition 3.2, it follows that any inverse mapping of any rational marked Λ_N is a finite union of graphs $W.H$ with $W \in Rat(N^*)$.

Corollary 3.3. *For any $h \subseteq T \times (N \cup \bar{N} \cup \{\#\})^*$ and $L \in Rat(N^*)$,*

$$h^{-1}(\#_L(\Lambda_N)) = \bigcup_{W \in [L]} W.\{\widetilde{u \downarrow J} \overset{a}{\longrightarrow} v \downarrow J | uv \in h(a) \wedge \tilde{\bar{u}}, v \in \mathscr{L}(\#_{W^{-1}L}(\Lambda_N))\}.$$

To get a prefix form of Corollary 3.3, we take the mirror $\tilde{\Lambda}_N$ of Λ_N where

$$\tilde{G} = \{\tilde{u} \xrightarrow{a} \tilde{v} \mid u \xrightarrow[G]{a} v\} \text{ is the } \textit{mirror} \text{ of graph } G \subseteq N^* \times T \times N^*.$$

By applying Corollary 3.3 to $\#_L(\tilde{\Lambda}_N) = (\widetilde{\#_{\tilde{L}}(\Lambda_N)})$, we have for every mapping $h \subseteq T \times (N \cup \bar{N} \cup \{\#\})^*$ and for every $L \in Rat(N^*)$,

$$h^{-1}(\#_L(\tilde{\Lambda}_N)) \\ = \bigcup_{W \in [L]} \{\widetilde{u \downarrow J} \xrightarrow{a} \widetilde{v \downarrow J} \mid uv \in h(a) \wedge \tilde{\bar{u}},\ v \in \mathscr{L}(\#_{W^{-1}\tilde{L}}(\Lambda_N))\}.\tilde{W}. \tag{2}$$

Proposition 3.2 has also a simple form when we do not use marking.

Corollary 3.4. *For any mapping $h \subseteq T \times (N \cup \bar{N})^*$, we have the T-graph*

$$h^{-1}(\Lambda_N) = N^*.\{u \xrightarrow{a} v \mid \tilde{\bar{u}}v \in h(a) \downarrow I \wedge u,\ v \in N^* \wedge a \in T\}.$$

Let $Dyck_N = [\Lambda_N]$ where $[G]$ is the set of all graphs isomorphic to a graph G, that we extend by union to any class of graphs. We restrict here a *language family* F to be a subset of $2^{(N \cup \bar{N} \cup \{\#\})^*}$. A family of languages defines a set of mappings: a mapping h is rational (resp linear, ...) if for any letter a, the language $h(a)$ is rational (resp linear, ...). Precisely, a language family F and an alphabet T produce the set F_T of mappings defined for every $a \in T$ by a language $h(a) \in F$. By inverse of a class Φ of $(N \cup \bar{N} \cup \{\#\})$-graphs, we get the following class of T-graphs:

$$F_T^{-1}(\Phi) := \{h^{-1}(G) \mid G \in \Phi \wedge h \in F_T\}.$$

Starting from $Dyck_N$, we have two ways to get classes of graphs. Either we apply inverse F-mappings followed by rational restrictions [11]:

$$F_T^{-1}(Dyck_N)_| := [\{h^{-1}(\Lambda_N)_{|L} \mid h \in F_T \wedge L \in Rat(N^*)\}]$$

or we apply rational markings followed by inverse F-mappings:

$$F_T^{-1}(\#(Dyck_N)) := [\{h^{-1}(\#_L(\Lambda_N)) \mid h \in F_T \wedge L \in Rat(N^*)\}].$$

Henceforth F will be one of the following language families: the family *Fin* of finite languages; the family *Rat* of rational languages; the family *Lin* of linear languages; the family *RE* of recursively enumerable languages; the subfamily $\bar{Lin}$ of linear languages generated by linear grammars such that each right hand side is ε or of the form uBv where B is a nonterminal with $u \in \bar{N}^*$ and $v \in N^*$; and the rational closure $\bar{Lin}(Rat)$ of $\bar{Lin}$. Each of these families is an *internal family* meaning that it satisfies the two following conditions:

(i) $L \in F \Rightarrow \#L\# \in F$,

(ii) $L \in F \Rightarrow h(L){\downarrow}J \cap \bar{N}^*N^* \in F$
for any finite substitution h from $(N \cup \bar{N} \cup \{\#\})^*$ into itself such that $h(\#){\downarrow}\tilde{J} \subseteq \{\varepsilon\}$ and for any $a \in N$, $h(\bar{a}) = \widetilde{\overline{h(a)}}$ and $h(a){\downarrow}\tilde{J} \subseteq N^*$.

Note that any family closed by every rational binary relation and called a *rational cone* [4], is an internal family. For these families, the two previous classes of graphs coincide and we can also restrict N to have only two letters.

Proposition 3.5. *For any distinct letters $a,b\in N$ and for any internal family F, we have*

$$F_T^{-1}(Dyck_N)_| = F_T^{-1}(Dyck_{\{a,b\}})_| = F_T^{-1}(\#(Dyck_{\{a,b\}})) = F_T^{-1}(\#(Dyck_N))$$

This class is denoted REC_{F_T} or REC_F when T is undertood.

Proof. (i) Let us show that $F_T^{-1}(Dyck_{\{a,b\}})_| \subseteq F_T^{-1}(Dyck_N)_|$. Let $G\in F_T^{-1}(Dyck_{\{a,b\}})_|$. So G is isomorphic to $h^{-1}(\Lambda_{\{a,b\}})_{|L}$ with $h\in F_T$ and $L\in Rat(N^*)$.

We have $\Lambda_{\{a,b\}} = \imath^{-1}(\Lambda_N)_{|\{a,b\}^*}$ where $\imath = Id_{\{a,b,\bar{a},\bar{b}\}}$.

Note that $\{a,b\}^*$ is stable for $\imath^{-1}(\Lambda_N)$. Henceforth G is isomorphic to

$$\begin{aligned} h^{-1}(\imath^{-1}(\Lambda_N)_{|\{a,b\}^*})_{|L} &= (h^{-1}(\imath^{-1}(\Lambda_N))_{|\{a,b\}^*})_{|L} && \text{by Lemma 2.2} \\ &= (h\circ\imath)^{-1}(\Lambda_N)_{|V_{\imath^{-1}(\Lambda_N)}\cap\{a,b\}^*\cap L} && \text{by Lemma 2.1 (c)} \\ &= (h\circ\imath)^{-1}(\Lambda_N)_{|\{a,b\}^*\cap L}. \end{aligned}$$

By condition (ii) of an internal family F (restricted to partial morphism), we have $h\circ\imath\in F_T$ hence $G\in F_T^{-1}(Dyck_N)_|$.

(ii) Let us show that $F_T^{-1}(Dyck_N)_| \subseteq F_T^{-1}(\#(Dyck_N))$. Let $G\in F_T^{-1}(Dyck_N)_|$.

So G is isomorphic to $h^{-1}(\Lambda_N)_{|L}$ with $h\in F_T$ and $L\in Rat(N^*)$.

By Lemma 2.3, $h^{-1}(\Lambda_N)_{|L} = g^{-1}(\#_L(\Lambda_N))$ with $g(a) = \#h(a)\#$ for any $a\in T$.

By condition (i) of an internal family F, we have $g\in F_T$ hence $G\in F_T^{-1}(\#(Dyck_N))$.

(iii) Let us show that $F_T^{-1}(\#(Dyck_N)) \subseteq F_T^{-1}(Dyck_{\{a,b\}})_|$.

Let $G\in F_T^{-1}(\#(Dyck_N))$.

So G is isomorphic to $h^{-1}(\#_L(\Lambda_N))$ with $h\in F_T$ and $L\in Rat(N^*)$.

Let (A,i,F) be a finite deterministic and complete automaton recognizing L and such that $p\xrightarrow[A]{c}q \wedge p\xrightarrow[A]{d}q \Rightarrow c=d$. By duplication of states, it is easy to satisfy this condition. However this condition is not necessary but it permits to simplify the notations.

Let $Q = V_A$ be the state set of A and let P be the language of words obtained from i by suffix derivation according to the relation $\{(p,pq) \mid \exists a\, p\xrightarrow[A]{a}q\}$. From [7], $P\in Rat(N^*)$.

So $\#_L(\Lambda_N)$ is isomorphic to $f^{-1}(\Lambda_Q)_{|P}$ where f is the following finite mapping:

$$f(a) = \{\bar{p}pq \mid p\xrightarrow[A]{a}q\};\quad f(\bar{a}) = \{\bar{q}\,\bar{p}p \mid p\xrightarrow[A]{a}q\};\quad f(\#) = \{\bar{p}p \mid p\in F\}.$$

By definition, P is the vertex set of the connected component of $f^{-1}(\Lambda_Q)$ containing i, hence P is stable for $f^{-1}(\Lambda_Q)$. Note that Q may have more than two letters. As for (i), we denote $Q = \{a_1, \ldots, a_n\}$ and we have

$$g\lfloor \Lambda_Q \rfloor = g^{-1}(\Lambda_{\{a,b\}})_{|M} \text{ with } \begin{array}{l} g(a_i) = ab^{i-1} \text{ and } g(\bar{a}_i) = \bar{b}^{i-1}\bar{a} \text{ for } i \in [n] \\ M = g(\{a_1, \ldots, a_n\}^*) = \{a, \ldots, ab^{n-1}\}^*. \end{array}$$

Note that M is stable for $g^{-1}(\Lambda_{\{a,b\}})$. Henceforth $\#_L(\Lambda_N)$ is isomorphic to

$$\begin{aligned} g\lfloor f^{-1}(\Lambda_Q)_{|P} \rfloor &= g\lfloor f^{-1}(\Lambda_Q) \rfloor_{|g(P)} \\ &= f^{-1}(g\lfloor \Lambda_Q \rfloor)_{|g(P)} \\ &= f^{-1}(g^{-1}(\Lambda_{\{a,b\}})_{|M})_{|g(P)} \\ &= f^{-1}(g^{-1}(\Lambda_{\{a,b\}}))_{|M \cap g(P)} && \text{by Lemma 2.2} \\ &= (f \circ g)^{-1}(\Lambda_{\{a,b\}})_{|V_{g^{-1}(\Lambda_{\{a,b\}})} \cap M \cap g(P)} && \text{by Lemma 2.1 (c)} \\ &= (f \circ g)^{-1}(\Lambda_{\{a,b\}})_{|g(P)}. \end{aligned}$$

Note that $g(P)$ is stable for $(f \circ g)^{-1}(\Lambda_{\{a,b\}})$. By Lemma 2.2, G is isomorphic to $h^{-1}((f \circ g)^{-1}(\Lambda_{\{a,b\}})_{|g(P)}) = (h \circ f \circ g)^{-1}(\Lambda_{\{a,b\}})_{|g(P)} = ((h \circ f \circ g){\downarrow}J)^{-1}(\Lambda_{\{a,b\}})_{|g(P)}$ where for any $x \in T, (h \circ f \circ g){\downarrow}J(x) = ((f \circ g)(h(x))){\downarrow}J$.

As $f \circ g : (N \cup \bar{N} \cup \{\#\})^* \to 2^{\{a,b,\bar{a},\bar{b}\}^*}$ is a finite substitution, and by condition (ii) of an internal family F, we have $(h \circ f \circ g){\downarrow}J \in F_T$ hence G belongs to $F_T^{-1}(Dyck_{\{a,b\}})_{|}$.

(iv) By (i), (ii), (iii), we have

$$F_T^{-1}(Dyck_{\{a,b\}})_{|} \subseteq F_T^{-1}(Dyck_N)_{|} \subseteq F_T^{-1}(\#(Dyck_N)) \subseteq F_T^{-1}(Dyck_{\{a,b\}})_{|}$$

Hence $F_T^{-1}(Dyck_{\{a,b\}})_{|} = F_T^{-1}(Dyck_N)_{|} = F_T^{-1}(\#(Dyck_N))$.

For $N = \{a,b\}$, the last equation is $F_T^{-1}(Dyck_{\{a,b\}})_{|} = F_T^{-1}(\#(Dyck_{\{a,b\}}))$. $\square$

The class REC_{Fin} is the set of *regular graphs* (see [22,15,9]) of bounded degree, and we present again two sets of representatives.

Theorem 3.6 (Caucal [10,11]). *Given an alphabet N of at least two letters, the following properties are equivalent*:

(a) $G \in REC_{\text{Fin}_T}$,

(b) *G is isomorphic to $(H.N^*)_{|L}$ for some finite $H \subseteq N^* \times T \times N^*$ and $L \in Rat(N^*)$*,

(c) *G is isomorphic to $\bigcup_{i=1}^n (u_i \xrightarrow{a_i} v_i).W_i$ for some $n \geqslant 0$, $a_1, \ldots, a_n \in T$, $u_1, v_1, \ldots, u_n, v_n \in N^*$, $W_1, \ldots, W_n \in Rat(N^*)$*,

(d) *G is a regular T-graph of bounded degree.*

The traces of the graphs in REC_{Fin} are all the context-free languages.

Proof. (i) $(a) \Rightarrow (b)$: Let $G \in REC_{\text{Fin}_T}$.

We have G isomorphic to $h^{-1}(\tilde{\Lambda}_N)_{|L}$ for some $h \in Fin_T$ and $L \in Rat(N^*)$.

Taking the finite graph $H = \{u \xrightarrow{a} v \mid a \in T \wedge \bar{u}\tilde{v} \in h(a){\downarrow}I\}$ and by Corollary 3.4, we have $h^{-1}(\tilde{\Lambda}_N) = H.N^*$.

(ii) $(b) \Rightarrow (a)$: Let a finite graph $H \subseteq N^* \times T \times N^*$ and $L \in Rat(N^*)$.

By Corollary 3.4, $H = h^{-1}(\tilde{\Lambda}_N)$ such that $h(a) = \{\bar{u}\tilde{v} \mid u \underset{H}{\xrightarrow{a}} v\}\ \forall a \in T$.

(iii) $(a) \Rightarrow (c)$: Let $G \in REC_{\mathrm{Fin}_T}$.

So G is isomorphic to $h^{-1}(\#_L(\tilde{\Lambda}_N))$ with $h \in Fin_T$ and $L \in Rat(N^*)$.

It remains to apply Equation (2) which is the prefix form of Corollary 3.3.

(iv) $(c) \Rightarrow (a)$: Let $n \geqslant 0$, $a_1, \ldots, a_n \in T$, $u_1, v_1, \ldots, u_n, v_n \in N^*$.

Let $W_1, \ldots, W_n \in Rat(N^*)$.

Let us show that $G = \bigcup_{i=1}^{n}(u_i \xrightarrow{a_i} v_i).W_i$ is in REC_{Fin_T}.

Taking the following rational language L and the following finite mapping h:

$$L = \bigcup_{i=1}^{n} a_i W_i \quad \text{and} \quad h(a) = \{\bar{u_i} a_i \# \bar{a}_i \tilde{v_i} \mid a_i = a\} \quad \text{for every } a \in T$$

we have $G = h^{-1}(\#_L(\tilde{\Lambda}_{N \cup T}))$ in REC_{Fin}. $\square$

Recall that a graph $G \subseteq N^* \times T \times N^*$ is *recognizable* if G is a finite union of elementary graphs of the form $U \xrightarrow{a} V$ where $a \in T$ and $U, V \in Rat(N^*)$. The class REC_{Rat} has been studied in [11] and we present again two sets of representatives.

Theorem 3.7 (Caucal [11]). *Given an alphabet N of at least two letters, the following properties are equivalent*:

(a) $G \in REC_{\mathrm{Rat}_T}$.

(b) *G is isomorphic to $(H.N^*)_{|L}$ for some recognizable $H \subseteq N^* \times T \times N^*$ and $L \in Rat(N^*)$.*

(c) *G is isomorphic to $\bigcup_{i=1}^{n}(U_i \xrightarrow{a_i} V_i).W_i$ for some $n \geqslant 0$, $a_1, \ldots, a_n \in T$, $U_1, V_1, W_1, \ldots, U_n, V_n, W_n \in Rat(N^*)$.*

The traces of the graphs in REC_{Rat} are all the context-free languages.

Proof. The implications $(b) \Rightarrow (a)$ and $(c) \Rightarrow (a)$ are as in the proof of Theorem 3.6.

(i) $(a) \Rightarrow (b)$: as in the proof of Theorem 3.6, it remains to verify that for any $L \in Rat((N \cup \bar{N})^*), L{\downarrow}I \cap \bar{N}^*N^*$ is a finite union of sets of the form $A.B$ where $A \in Rat(\bar{N}^*)$ and $B \in Rat(N^*)$.

Let (A, i, F) be a finite automaton recognizing L and let $Q = V_A$ be the state set of A. We have

$$L \downarrow I \cap \bar{N}^*N^* = \bigcup_{q \in Q} (\overline{\mathscr{L}(G, i, q)} \cap \mathscr{L}(\Lambda_N)) \downarrow I.(\widetilde{\mathscr{L}(G, q, F)} \cap \mathscr{L}(\Lambda_N)) \downarrow I$$

and the rationality follows from Lemma 3.1.

(ii) $(a) \Rightarrow (c)$: Let $G \in REC_{\mathrm{Rat}_T}$.

So G is isomorphic to $h^{-1}(\#_L(\tilde{\Lambda}_N))$ with $h \in Rat_T$ and $L \in Rat(N^*)$.

For every $a \in T$, let (A_a, i_a, F_a) be a finite automaton recognizing $h(a)$ and let $Q_a = V_{A_a}$ be the state set of A_a.

By applying Eq. (2) which is the prefix form of Corollary 3.3, $h^{-1}(\#_L(\tilde{\Lambda}_N))$ is equal to

$$\bigcup_{\substack{W\in[L]\\ a\in T\\ q\in Q_a}} ((\mathscr{L}\overline{(G_a,i_a,q)} \cap \mathscr{L}(\#_{W^{-1}\tilde{L}}(\Lambda_N))) \downarrow J \xrightarrow{a} (\mathscr{L}(\widetilde{G_a,q},F_a) \cap \mathscr{L}(\#_{W^{-1}\tilde{L}}(\Lambda_N))) \downarrow J).\tilde{W}$$

and the rationality follows from Lemma 3.1. □

Several characterizations of REC_{Fin} inside REC_{Rat} have been given [2,13]. A major question is the closure of REC_F by inverse F-mappings. We denote by

$$F(E) := \{h(L) \mid L \in E \wedge h \in F_{N\cup\bar{N}\cup\{\#\}}\}$$

the family obtained by applying F substitutions to a family E.

In particular $Fin(Fin) = Fin$ and $Fin(Rat) = Rat(Fin) = Rat(Rat) = Rat$.

Lemma 3.8. *Let F be the internal family Fin or Rat. Let E be any family such that $F(E)$ is internal. We have $E_T^{-1}(REC_{F_N}) \subseteq REC_{F(E)_T}$.*

Proof. (i) Let $G \in E_T^{-1}(REC_{F_N}) : G = g^{-1}(H)$ for some $g \in E_T$ and $H \in REC_{F_N}$.

So H is isomorphic to $h^{-1}(\#_L(\Lambda_N))$ with $h \in F_N$ and $L \in Rat(N^*)$.

Hence G is isomorphic to $g^{-1}(h^{-1}(\#_L(\Lambda_N)))$.

By Lemma 2.1 (c), G is isomorphic to $(g\circ h)^{-1}(\#_L(\Lambda_N))_{|V_{h^{-1}(\#_L(\Lambda_N))}}$.

For any $a \in T$, $(g\circ h)(a) = h(g(a)) \in F(E)$, hence $(g\circ h)^{-1}(\#_L(\Lambda_N)) \in REC_{F(E)_T}$.

So $(g\circ h)^{-1}(\#_L(\Lambda_N))$ is isomorphic to $k^{-1}(\Lambda_N)_{|M}$ with $k \in F(E)_T$ and with $M \in Rat(N^*)$. Finally G is isomorphic to $k^{-1}(\Lambda_N)_{|M\cap V_{h^{-1}(\#_L(\Lambda_N))}}$.

(ii) It remains to show that $V_{h^{-1}(\#_L(\Lambda_N))}$ is rational. By union, it is sufficient to assume that $h(a) \in Rat$ for some $a \in T$ and $h(b) = \emptyset$ for any $b \in T - \{a\}$.

So $h(a) = \mathscr{L}(A,i,Q_f)$ is recognized by some finite automaton (A,i,Q_f). Let $Q = V_A$ be the state set of the automaton. By Corollary 3.3, we have

$$V_{h^{-1}(\#_L(\Lambda_N))} = \bigcup_{\substack{W\in[L]\\ q\in C(Q)}} W.([\mathscr{L}\overline{\widetilde{(G,i,q)}} \cup \mathscr{L}(G,q,Q_f)] \cap \mathscr{L}(\#_{W^{-1}L}(\Lambda_N))) \downarrow J,$$

where $C(Q)$ is the set $q \in Q$ such that the languages $\mathscr{L}\overline{\widetilde{(G,i,q)}} \cap \mathscr{L}(\#_{W^{-1}L}(\Lambda_N))$ and $\mathscr{L}(G,q,Q_f) \cap \mathscr{L}(\#_{W^{-1}L}(\Lambda_N))$ are nonempty. The rationality follows from Lemma 3.1. □

We deduce closure properties for REC_{Fin} and REC_{Rat}.

Proposition 3.9. *We have $Fin_T^{-1}(REC_{\mathrm{Fin}_N}) = REC_{\mathrm{Fin}_T}$ and $Rat_T^{-1}(REC_{\mathrm{Fin}_N}) = Rat_T^{-1}(REC_{\mathrm{Rat}_N}) = REC_{\mathrm{Rat}_T}$.*

Proof. As $\#(Dyck_{\{a,b\}}) \in REC_{\mathrm{Fin}_N}$, we have

$$REC_{\mathrm{Fin}_T} = Fin_T^{-1}(\#(Dyck_{\{a,b\}})) \subseteq Fin_T^{-1}(REC_{\mathrm{Fin}_N}).$$

As $Fin(Fin) = Fin$ and by Lemma 3.8, we have $Fin_T^{-1}(REC_{\mathrm{Fin}_N}) \subseteq REC_{\mathrm{Fin}_T}$.
Similarly, we deduce the two other equalities. □

Note that the closure of REC_{Rat} by any inverse rational mapping has been obtained in [11] with a long proof.

It remains to recall the family of rational graphs [20]. We consider a graph as a subset of $N^* \times T \times N^*$ i.e. a T-graph with vertices in N^*. We extend the monoid $N^* \times N^*$ to the partial semigroup $N^* \times T \times N^*$ defined by $(u,a,v).(x,a,y) = (ux,a,vy)$ for every $u,v,x,y \in N^*$ and $a \in T$. The extension by union of . to subsets is the usual *synchronization product* for graphs [1]:

$$G.H = \{ux \xrightarrow{a} vy \mid u \underset{G}{\xrightarrow{a}} v \wedge x \underset{H}{\xrightarrow{a}} y\} \quad \text{for any } G,H \subseteq N^* \times T \times N^*.$$

To this operation is associated the rational family $Rat(N^* \times T \times N^*)$ of graphs: it is the smallest subset of $2^{N^* \times T \times N^*}$ containing the finite graphs and closed by $\cup, ., +$. A *rational graph* is a graph isomorphic to a graph in $Rat(N^* \times T \times N^*)$; we denote by RAT_T the family of rational T-graphs. The rational graphs are the graphs recognized by the labelled transducers. Precisely, a *T-labelled transducer* is a finite $(N^* \times N^*)$-automaton $A = (G, i, (F_a)_{a \in T})$ with a set F_a of final states for each $a \in T$; such an automaton recognizes the graph:

$$\mathscr{L}(A) := \{u \xrightarrow{a} v \mid \exists s \in F_a, i \underset{G}{\overset{u/v}{\Longrightarrow}} s\}.$$

The family $\bar{L}in$ defines by inverse mappings the class of rational graphs.

Theorem 3.10 (Morvan [20], Morvan and Stirling [21]). We have

$$RAT_T = REC_{\bar{\mathrm{L}}\mathrm{in}_T} \subset REC_{\mathrm{Lin}_T}.$$

The traces of the graphs in RAT are the context-sensitive languages.

A particular rational graph is an *automatic graph* [5] which is a graph isomorphic to a graph recognized by a labelled left-synchronized (or by a labelled right-synchronized) transducer [17,18]. The traces of the automatic graphs remain the context-sensitive languages [24]. Note that we can have nonrecursive traces for graphs in REC_{Lin}. From the closure by composition of rational relations, the rational graphs are closed by inverse finite mappings.

Proposition 3.11. *We have* $Fin_T^{-1}(RAT_N) = RAT_T$.

We will now use Turing machines to define a general class of graphs whose the traces are the recursively enumerable languages.

4. Graphs of rewriting systems and of Turing machines

We consider the rational restrictions of the ε-closure for the set of transitions of the labelled Turing machines. We show that this family is the same that for the labelled word rewriting systems (Theorem 4.5). We show also that this family is REC_F for any family F of recursively enumerable languages containing the rational closure of the linear languages (Theorem 4.6). Furthermore, we show that this family is the set of the inverse rational mappings of the rational graphs (Theorem 4.7). Finally, we show that this family is also the set of graphs recognized by (unlabelled) Turing machines with labelled final states (Theorem 4.8), and even if we restrict to deterministic Turing machines (Theorem 4.9).

The notion of a word-rewriting system is well known (see for instance the survey [16,6]): it is just a finite set of rules between words. As for the transitions of a push-down automaton, we allow labelled rules, and to any system, we associate a rational language of admissible words, usually called configurations, which are the words where the rules can be applied. The words are over an alphabet (finite set of symbols) N of *nonterminals*, and the rules are labelled by symbols in an alphabet T of *terminals*, plus the empty word ε.

Definition 4.1. A finite *labelled rewriting system* (R, C) over words is a couple of a finite relation $R \subseteq N^* \times (T \cup \{\varepsilon\}) \times N^*$ and a rational language $C \subseteq N^*$ of *configurations*. We write shortly R instead of (R, N^*).

The set of transitions of R is the following $(T \cup \{\varepsilon\})$-graph:

$$T(R) := \{xuy \xrightarrow{a} xvy \mid (u, a, v) \in R \wedge x, y \in N^*\}.$$

The unlabelled transitions of $T(R)$ form the usual *rewriting* $\underset{R}{\longrightarrow}$ of R:

$$xuy \underset{R}{\longrightarrow} xvy \quad \text{for some } (u, a, v) \in R \text{ with } x, y \in N^*.$$

Its reflexive and transitive closure $\underset{R}{\longrightarrow}^*$ by composition is the *derivation* of R. To any system (R, C), we associate its *transition graph*:

$$G(R, C) := \overline{T(R)}_{|C} = \{u \xrightarrow{a} v \mid u \underset{T(R)}{\overset{a}{\Longrightarrow}} v \wedge u,\ v \in C \wedge a \in T\},$$

which is the restriction to C of the ε-closure of $T(R)$. In particular $G(R) = \overline{T(R)}$.

For instance, the transition relation of a *pushdown automaton* over a set Q of states and over a disjoint set P of stack letters, can be seen as a labelled rewriting system (R, C) over $N = P \cup Q$ where R is a finite subset of $Q.P \times (T \cup \{\varepsilon\}) \times Q.P^*$ and C is a rational subset of $Q.P^*$. The closure by isomorphism $[G(R, C)]$ of their transition graphs form the family REC_{Rat}.

Proposition 4.2. *We have*

$$G \in REC_{\mathrm{Rat}} \;\Leftrightarrow\; G \text{ isomorphic to } \overline{R.N^*}_{|C} \text{ for some system } (R,C).$$

Proof. $\Leftarrow$: by Theorem 3.6 and Proposition 3.9.

$\Rightarrow$: Let $G \in REC_{\mathrm{Rat}}$. By Theorem 3.7, G is isomorphic to the following graph:

$$H := \left(\bigcup_{i=1}^{n} (U_i \xrightarrow{a_i} V_i).N^* \right)_{|L} \quad \text{with } L, U_1, V_1, \ldots, U_n, V_n \in Rat(N^*).$$

For every $1 \leqslant i \leqslant n$, let (G_i, r_i, E_i) and (H_i, s_i, F_i) be finite N-automata recognizing, respectively, U_i and V_i. We may assume that $V_{G_1}, V_{H_1}, \ldots, V_{G_n}, V_{H_n}$ are pairwise disjoint. Let \$ be a new symbol. We define the following rewriting system (R,C):

$$R \left| \begin{array}{ll} \$ \xrightarrow{a_i} r_i & \text{for any } 1 \leqslant i \leqslant n, \\ pA \xrightarrow{\varepsilon} q & \text{for any } p \underset{G_i}{\xrightarrow{A}} q \text{ with } 1 \leqslant i \leqslant n, \\ p \xrightarrow{\varepsilon} t & \text{for any } p \in E_i,\ t \in F_i \text{ with } 1 \leqslant i \leqslant n, \\ t \xrightarrow{\varepsilon} sA & \text{for any } s \underset{H_i}{\xrightarrow{A}} t \text{ with } 1 \leqslant i \leqslant n, \\ s_i \xrightarrow{\varepsilon} \$ & \text{for any } 1 \leqslant i \leqslant n \end{array} \right.$$

and $C = \$L$. So $\overline{R.N^*}_{|C} = \$.H$. $\square$

Another particular labelled rewriting systems are the Turing machines with a read only input tape and a working tape [19,23]. More exactly and given an alphabet Q of states, a disjoint alphabet T of input tape letters, and a disjoint alphabet $P_\square = P \cup \{\square\}$ of working tape letters, a (nondeterministic) *labelled Turing machine* (M,C) is a finite set M of rules of the form:

$$pA \xrightarrow{a} qB\delta \quad \text{where } p,q \in Q,\ a \in T \cup \{\varepsilon\},\ A,B \in P_\square,\ \delta \in \{+,-\}$$

with a rational set $C \in Rat((Q \cup P_\square)^*)$ of configurations.

However we are only interested to configurations upv where $p \in Q$ and $u,v \in P_\square^*$ with $u(1), v(|v|) \neq \square$. Precisely a configuration is of the form $]u]p[v[$ where for any word $u \in P_\square^*$, $[u[$ (resp. $]u]$) is the greatest prefix (resp. suffix) of u having its last (resp. first) letter distinct of $\square$ i.e. by induction,

$$[u\square[= [u[\ \wedge\ [u[= u \text{ if } u(|u|) \neq \square \quad \text{and} \quad]\square u] =]u] \ \wedge\ ]u] = u \text{ if } u(1) \neq \square.$$

The set of transitions of M is the following $(T \cup \{\varepsilon\})$-graph:

$$\begin{aligned} T(M) := & \{]u]p[Av[\xrightarrow{a}]uB]q[v[\mid pA \underset{M}{\xrightarrow{a}} qB+ \wedge u,v \in P_\square^* \} \\ & \cup \{]uC]p[Av[\xrightarrow{a}]u]q[CBv[\mid pA \underset{M}{\xrightarrow{a}} qB- \wedge C \in P_\square \wedge u,v \in P_\square^* \}. \end{aligned}$$

Hence the *transition graph* of any labelled Turing machine (M,C) is the T-graph:

$$G(M,C) := \overline{T(M)}_{|C}.$$

The transition graph of any labelled Turing machine is the transition graph of a *stable* labelled rewriting system (R, C) meaning that C is stable in $T(R)$:

$$s \underset{R}{\longrightarrow}^* r \underset{R}{\longrightarrow}^* t \wedge s,\ t \in C \quad \Rightarrow \quad r \in C.$$

Lemma 4.3. *We can transform any labelled Turing machine M into a stable labelled rewriting system (R, C) such that $T(R)_{|C}$ is isomorphic to $T(M)$.*

Proof. We take a new symbol \$ and the following rational language:

$$C = \{\$upv\$ \mid p \in Q \wedge u, v \in P_\Box^* \wedge u(1), v(|v|) \neq \Box\}.$$

We transform any rule $pA \xrightarrow{a} qB+$ of M into the following rules:

$$\begin{array}{ll} CpA \xrightarrow{a} CBq & \text{if } CB \neq \$\,\Box \\ Cp\$ \xrightarrow{a} CBq\$ & \text{if } A = \Box \wedge CB \neq \$\,\Box \\ \$pA \xrightarrow{a} \$q & \text{if } B = \Box \\ \$p\$ \xrightarrow{a} \$q\$ & \text{if } A = B = \Box \end{array}$$

We transform any rule $pA \xrightarrow{a} qB-$ of M into the following rules:

$$\begin{array}{ll} CpAD \xrightarrow{a} qCBD & \text{if } C \neq \$ \wedge BD \neq \Box\,\$ \\ CpA\$ \xrightarrow{a} qC\$ & \text{if } B = \Box \wedge C \neq \Box\,\$ \\ \Box pA\$ \xrightarrow{a} q\$ & \text{if } B = \Box \\ Cp\$ \xrightarrow{a} qCB\$ & \text{if } A = \Box \neq B \wedge C \neq \$ \\ Cp\$ \xrightarrow{a} qC\$ & \text{if } A = B = \Box \wedge C \neq \Box, \$ \\ \Box p\$ \xrightarrow{a} q\$ & \text{if } A = B = \Box \\ \$pAD \xrightarrow{a} \$q\Box BD & \text{if } BD \neq \Box\,\$ \\ \$pA\$ \xrightarrow{a} \$q\$ & \text{if } B = \Box \\ \$p\$ \xrightarrow{a} \$q\Box B\$ & \text{if } A = \Box \neq B \\ \$p\$ \xrightarrow{a} \$q\$ & \text{if } A = B = \Box \end{array}$$

In this way, we obtain a labelled rewriting system R such that

C is closed by $\underset{R}{\longrightarrow}$ hence (R, C) is stable

$$U \underset{T(M)}{\xrightarrow{a}} V \quad \Leftrightarrow \quad \$U\$ \underset{T(R)}{\xrightarrow{a}} \$V\$ \wedge \$U\$, \$V\$ \in C.$$

Thus $\$T(M)\$ = T(R)_{|C}$. $\Box$

Conversely and up to the ε-transitions, any labelled rewriting system can be simulated by a labelled Turing machine.

Lemma 4.4. *We can transform any labelled rewriting system R into a labelled Turing machine (M, C) such that $\overline{T(M)}_{|C}$ is isomorphic to $\overline{T(R)}$.*

Proof. We denote by m_1 (resp. m_2) the maximum length of the left (resp. right)-hand sides of the rules of R i.e.

$$m_1 = \max\{|U| \mid \exists a, V, (U, a, V) \in R\}$$

and

$$m_2 = \max\{|V| \mid \exists U, a, (U, a, V) \in R\}.$$

We take two new symbols $\bullet$ and \$, and we define the following state set Q of the labelled Turing machine to be constructed:

$$Q = \{\bullet\} \cup N^{\leqslant m_1} \times (N^{\leqslant m_2} \cup \{\$\}) \times (T \cup \{\varepsilon\}).$$

We take the following set M' of Turing rules:

$$\bullet A \xrightarrow{\varepsilon} \bullet A+ \qquad \text{for } A \in N_\Box$$

$$\bullet A \xrightarrow{\varepsilon} \bullet A- \qquad \text{for } A \in N_\Box$$

$$\bullet A \xrightarrow{\varepsilon} (U, V, a)A+ \quad \text{for } A \in N_\Box \text{ and } (U, a, V) \in R$$

$$\begin{aligned} &(AU, BV, a)A \xrightarrow{\varepsilon} (U, V, a)B+ \\ &\quad (\varepsilon, BV, a)A \xrightarrow{\varepsilon} (\varepsilon, VA, a)B+ \quad \text{for } A \in N \\ &\quad (\varepsilon, BV, a)\Box \xrightarrow{\varepsilon} (\varepsilon, V, a)B+ \end{aligned}$$

$$\begin{aligned} &(AU, \varepsilon, a)A \xrightarrow{\varepsilon} (UB, \varepsilon, a)B+ \text{ for } B \in N \\ &(AU, \varepsilon, a)A \xrightarrow{\varepsilon} (U, \$, a)\Box + \\ &(AU, \$, a)A \xrightarrow{\varepsilon} (U, \$, a)\Box + \\ &(\varepsilon, \$, a)\ \Box \xrightarrow{\varepsilon} (\varepsilon, \varepsilon, a)\Box + \end{aligned}$$

In this way, we obtain a labelled Turing machine M' such that for every $a \in T \cup \{\varepsilon\}$ and $U, V \in N^*$,

$$U \underset{T(R)}{\xrightarrow{a}} V \quad \Leftrightarrow \quad \bullet U \underset{T(M')}{\overset{\varepsilon}{\Rightarrow}} X(\varepsilon, \varepsilon, a)Y \wedge [XY[= V.$$

We complete M' to M by adding the following rules:

$$(\varepsilon, \varepsilon, a)A \xrightarrow{a} \bullet A + \quad \text{for } A \in N_\Box$$

The relation $H = \{(u, \bullet U) \mid U \in N^*\}$ is a partial weak isomorphism from $T(R)$ into $T(M)$. More precisely and for every $a \in T \cup \{\varepsilon\}$, we have

$$U \underset{T(R)}{\overset{a}{\longrightarrow}} V \Rightarrow \bullet U \underset{T(M)}{\overset{a}{\Longrightarrow}} \bullet V,$$

$$\bullet U \underset{T(M)}{\overset{a}{\Longrightarrow}} \bullet V \Rightarrow U \underset{T(R)}{\overset{a}{\Longrightarrow}} V.$$

So h is a partial isomorphism from $\overline{T(R)}$ into $\overline{T(M)}$.

Thus h is a partial isomorphism from $\overline{T(R)}$ into $\overline{T(M)}_{|C}$ with the rational language $C = Im(h) = \bullet N^*$.

As $V_{\overline{T(R)}} \subseteq N^* = Dom(h)$, the graphs $\overline{T(R)}$ and $\overline{T(M)}_{|C}$ are isomorphic. $\square$

The rewriting systems and the Turing machines have the same transition graphs.

Theorem 4.5. *The labelled Turing machines and the labelled rewriting systems define up to isomorphism, the same family of transition graphs, and their traces are the recursively enumerable languages.*

Proof. (i) Let (M, D) be a labelled Turing machine.

By Lemma 4.3, we can construct a stable labelled rewriting system (R, C) and an isomorphism h from $T(M)$ to $T(R)_{|C}$.

By restriction, h defines an isomorphism from $\overline{T(M)}$ to $\overline{T(R)_{|C}}$.

By Eq. (2.2), $\overline{T(R)_{|C}} = \overline{T(R)}_{|C} = G(R, C)$. Thus $G(M, D) = \overline{T(M)}_{|D}$ is isomorphic (by a restriction of h) to $\overline{T(R)}_{|C \cap h(D)} = G(R, C \cap h(D))$.

(ii) Let (R, C) be a labelled rewriting system.

By Lemma 4.4, we can construct a labelled Turing machine (M, D) and an isomorphism h from $\overline{T(R)}$ to $\overline{T(M)}_{|D}$. Thus $G(R, C) = \overline{T(R)}_{|C}$ is isomorphic (by a restriction of h) to $\overline{T(M)}_{|h(C) \cap D} = G(M, h(C) \cap D)$. $\square$

We denote by $TURING_T$ the family of T-graphs isomorphic to the transition graphs of labelled Turing machines (or of labelled rewriting systems). As for the previous graph families (investigated in the previous section), we characterize the family $TURING_T$ by inverse mappings of the binary tree. The images of these mappings can be the class of recursively enumerable languages, or can be only the class of the rational closure $\bar{L}in(Rat)$ of $\bar{L}in$.

Theorem 4.6. *We have* $TURING_T = REC_{\bar{\mathrm{L}}\mathrm{in}(\mathrm{Rat})_T} = REC_{\mathrm{RE}_T}$.

Proof. (i) Let us show that $TURING_T \subseteq REC_{\bar{\mathrm{L}}\mathrm{in}(\mathrm{Rat})_T}$.

Let (R, C) be a labelled rewriting system: R is a finite subset of $N^* \times (T \cup \{\varepsilon\}) \times N^*$ and C is a rational subset of N^*.

We replace in R the label ε by a new letter \$:

$$S := \{(u, a, v) \in R \mid a \in T\} \cup \{(u, \$, v) \mid (u, \varepsilon, v) \in R\}.$$

So $T(S) = h^{-1}(\Lambda_N)$ where h is the following linear mapping:

$$h(a) = \{\tilde{\bar{x}}\,\tilde{\bar{u}}vx \mid (u,a,v) \in S \wedge x \in N^*\} \quad \text{for every } a \in T \cup \{\$\}.$$

Furthermore $\overline{T(R)} = g^{-1}(T(S))$ where g is the following rational mapping:

$$g(a) = \$^* a \$^* \text{ for every } a \in T.$$

By (2.1), we have $\overline{T(R)} = (g \circ h)^{-1}(\Lambda_N)$ where $g \circ h$ is the following mapping:

$$(g \circ h)(a) = h(g(a)) = h(\$^* a \$^*) = h(\$)^* h(a) h(\$)^* \in \bar{L}in(Rat).$$

Finally and by Proposition 3.5, $G(R,C) = \overline{T(R)}_{|C} \in REC_{\bar{\mathrm{L}}\mathrm{in}(\mathrm{Rat})_T}$.

(ii) $REC_{\bar{\mathrm{L}}\mathrm{in}(\mathrm{Rat})} \subseteq REC_{\mathrm{RE}}$ because $\bar{L}in(Rat) \subseteq RE$.

(iii) Let us show that $REC_{\mathrm{RE}_T} \subseteq TURING_T$.

Let a mapping $h : T \to RE(\{a,b,\bar{a},\bar{b}\}^*)$. By Proposition 3.5, it is sufficient to construct a rewriting system (R,C) such that $h^{-1}(\Lambda_{\{a,b\}})$ is isomorphic to $\overline{T(R)}_{|C}$.

For every $c \in T$, there is a Turing machine M_c: a finite set of rules of the form:

$$pA \xrightarrow{x} qB\delta \quad \text{where } p,q \in Q_c,\ x \in \{\varepsilon,a,b,\bar{a},\bar{b}\},\ A,B \in P_c \cup \{\Box\},\ \delta \in \{+,-\}$$

plus an initial configuration i_c and a set $F_c \subseteq Q_c$ of final states recognizing:

$$h(c) = \mathscr{L}(T(M_c), i_c, \{]u]q[v[\mid q \in F_c \wedge u,v \in (P_c \cup \Box)^*\}).$$

Up to renaming, we may assume that the sets $(P_c)_{c\in T}, (Q_c)_{c\in T}$ are pairwise disjoint, and we define the following Turing machine:

$$\begin{aligned} M = {} & \{\, pA \xrightarrow{\varepsilon} qB\delta \in M_c \mid c \in T \} \\ & \cup \{\, pA \xrightarrow{\varepsilon} q_x B\delta \mid \exists c \in T,\ \ pA \xrightarrow{x} qB\delta \in M_c \wedge x \neq \varepsilon \}. \end{aligned}$$

We take three new symbols $\$, \&, \bullet$ and we construct a rewriting system R. First, we take the following rules:

$$|\$\$ \xrightarrow{\varepsilon} \$ i_c \$ \quad \text{for every } c \in T$$

to describe the moves between two \$ of the Turing machines defining h. We transform (as in Lemma 4.3) any rule $pA \xrightarrow{\varepsilon} qB+$ of M into the following rules:

$$\begin{array}{lll} CpA \xrightarrow{\varepsilon} & CBq & \text{if } CB \neq \$\,\Box \\ Cp\$ \xrightarrow{\varepsilon} & CBq\$ & \text{if } A = \Box \wedge CB \neq \$\,\Box \\ \$pA \xrightarrow{\varepsilon} & \$q & \text{if } B = \Box \\ \$p\$ \xrightarrow{\varepsilon} & \$q\$ & \text{if } A = B = \Box \end{array}$$

In a same way (and as in the proof of Lemma 4.3), we transform any rule $pA \xrightarrow{\varepsilon} qB-$ of M into new rules of R.

For every $c \in T$, $q \in Q_c$, $A \in P_c \cup \{\Box\}$, $y \in \{a, b, \bar{a}, \bar{b}\}$, $x \in \{a, b\}$, we take the rules:

$$\begin{aligned} q_y &\xrightarrow{\varepsilon} q'_y \& \\ Aq'_y &\xrightarrow{\varepsilon} q'_y A \\ x\$q'_{\bar{x}} &\xrightarrow{\varepsilon} \$\bar{q} \\ \$q'_x &\xrightarrow{\varepsilon} x\$\bar{q} \\ \bar{q}A &\xrightarrow{\varepsilon} A\bar{q} \\ \bar{q}\& &\xrightarrow{\varepsilon} q \end{aligned}$$

For the acceptance and for every $c \in T$ and $A \in (\bigcup_c P_c) \cup \{\Box\}$, we define

$$\begin{aligned} q &\xrightarrow{c} \bullet \quad \text{if } q \in F_c \\ \bullet A &\xrightarrow{\varepsilon} \bullet \\ A\bullet &\xrightarrow{\varepsilon} \bullet \\ \$ \bullet \$ &\xrightarrow{\varepsilon} \$\$ \end{aligned}$$

For every $c \in T$ and $u, v \in \{a, b\}^*$, we have

$$u \underset{h^{-1}(\Lambda_{\{a,b\}})}{\xrightarrow{c}} v \iff u\$\$ \underset{T(R)}{\overset{c}{\Rightarrow}} v\$\$.$$

So $h = \{(u, u\$\$) \mid u \in \{a, b\}^*\}$ is a partial weak isomorphism from $h^{-1}(\Lambda_{\{a,b\}})$ into $T(R)$.

Thus, h is a partial isomorphism from $\overline{h^{-1}(\Lambda_{\{a,b\}})} = h^{-1}(\Lambda_{\{a,b\}})$ into $\overline{T(R)}$. Hence h is an isomorphism from $h^{-1}(\Lambda_{\{a,b\}})$ into $\overline{T(R)}_{|C}$ where $C = Im(h) = \{a, b\}^*\$\$$. $\Box$

The class *TURING* is the closure of *RAT* by inverse rational mapping.

Theorem 4.7. *We have* $Rat_T^{-1}(RAT_N) = TURING_T$.

Proof. (i) $TURING_T \subseteq Rat_T^{-1}(RAT_N)$.

Let $G \in TURING_T : G$ is isomorphic to $\overline{T(R)}_{|C}$ for some labelled rewriting system (R, C). Let #, \$ be two new symbols. We have

$$\overline{T(R)} = h^{-1}(T(S)),$$

where h is the rational mapping defined by $h(a) = \$^* a \$^*$ for every $a \in T$, and S is the system obtained from R by replacing the label ε by \$:

$$S := \{(u, a, v) \in R \mid a \in T\} \cup \{(u, \$, v) \mid (u, \varepsilon, v) \in R\}.$$

By Eqs. (2.4) and (2.1), we have

$$\overline{T(R)}_{|C} = h_\#^{-1}(T(S) \cup \{u \xrightarrow{\#} u \mid u \in C\}),$$

where $h_\#(a) = \#\$^* a \$^* \#$ for every $a \in T$.

Obviously $T(S)$ is a rational graph and $\{u \xrightarrow{\#} u \mid u \in C\}$ is also a rational graph because C is a rational language. Hence $G \in Rat_T^{-1}(RAT_N)$.

(ii) $Rat_T^{-1}(RAT_N) \subseteq TURING_T$.

Let $G \in Rat_T^{-1}(RAT_N)$: so $G = h^{-1}(H)$ for some rational N-graph H and some mapping $h : T \to Rat(N^*)$.

By definition of a rational graph, there is an alphabet X and an N*-labelled transducer* $A = (K, i, (E_x)_{x \in N})$ where K is a finite $(X^* \times X^*)$-automaton, i is the initial state, and for each $x \in N$, E_x is a set of final states, and such that the automaton A recognizes the graph $\mathscr{L}(A) = \{u \xrightarrow{x} v \mid \exists s \in E_x,\ i \underset{K}{\overset{u/v}{\Rightarrow}} s\}$ which is isomorphic to H. Furthermore and for each $a \in T$, there is a finite N-automaton (K_a, i_a, F_a) recognizing the rational language $h(a)$.

We may assume that the automata $(K_a)_{a \in T}$ have pairwise disjoint state sets: for any $a \neq b$, $V_{K_a} \cap V_{K_b} = \emptyset$. We denote by $\bar{K} = \bigcup_{a \in T} K_a$ and we take a new state $\bar{\imath} \notin V_{\bar{K}}$.

We take a new symbol \$ and we denote by $C = \$\bar{\imath}X^*\$$ the (rational) configuration set of the following rewriting system R:

$$
R \left|
\begin{array}{ll}
\bar{\imath} \xrightarrow{\varepsilon} i_a & \text{for each } a \in T, \\
\$s \xrightarrow{\varepsilon} \$(i,s) & \text{for each } s \in V_{\bar{K}}, \\
\$s \xrightarrow{a} \$\bar{\imath} & \text{if } s \in F_a, \\
(p,s)u \xrightarrow{\varepsilon} v(q,s) & \text{if } p \underset{K}{\overset{u/v}{\longrightarrow}} q \text{ and } s \in V_{\bar{K}}, \\
(p,s)\$ \xrightarrow{\varepsilon} t\$ & \text{if } p \in E_x \text{ and } s \underset{\bar{K}}{\overset{x}{\longrightarrow}} t \ n \text{ for some } x \in N, \\
As \xrightarrow{\varepsilon} sA & \text{for each } A \in X \text{ and } s \in V_{\bar{K}}.
\end{array}
\right.
$$

Thus,

$$
\begin{aligned}
\overline{T(R)}_{|C} &= \{\$\bar{\imath}u\$ \xrightarrow{a} \$\bar{\imath}v\$ \mid \$\bar{\imath}u\$ \underset{T(R)}{\overset{a}{\Rightarrow}} \$\bar{\imath}v\$\} \\
&= \{\$\bar{\imath}u\$ \xrightarrow{a} \$\bar{\imath}v\$ \mid \exists s \in F_a \$i_a u\$ \underset{T(R)}{\overset{\varepsilon}{\Rightarrow}} \$sv\$\} \\
&= \{\$\bar{\imath}u\$ \xrightarrow{a} \$\bar{\imath}v\$ \mid \exists w \in h(a) u \underset{L(A)}{\overset{w(1)}{\longrightarrow}} \dots \underset{L(A)}{\overset{w(|w|)}{\longrightarrow}} v\} \\
&= \{\$\bar{\imath}u\$ \xrightarrow{a} \$\bar{\imath}v\$ \mid u \underset{L(A)}{\overset{h(a)}{\Rightarrow}} v\} \\
&= \$\bar{\imath}h^{-1}(L(A))\$ \text{ isomorphic to } h^{-1}(H) = G. \qquad \square
\end{aligned}
$$

In particular RAT is not closed by inverse rational mapping. We also deduce that the transition graphs of labelled Turing machines are the rational restrictions of the ε-closure of rational graphs (with ε-arcs).

An equivalent way to get the family $TURING_T$ is to consider the computable relations of single tape nondeterministic Turing machines. Precisely and given an alphabet Q of states and a disjoint alphabet $P_\square = P \cup \{\square\}$ of working tape letters, a (single tape non deterministic) *Turing machine* M is a finite set of rules of the form:

$$pA \longrightarrow qB\delta \quad \text{where } p,q \in Q,\ A,B \in P_\square,\ \delta \in \{+,-\}.$$

The set of transitions of M is the previous graph $T(M)$ which is unlabelled. For a T-labelling and as for the labelled transducers recognizing the rational graphs, we take a subset $F_a \subseteq Q$ of final states for each letter $a \in T$. Furthermore and as usual, we take an initial state $q_0 \in Q$. In this way, a Turing machine M defines the following *T-computation graph*:

$$R(M) := \{u \xrightarrow{a} \overleftarrow{v}\,\overrightarrow{w} \mid u \in P^* \wedge a \in T \wedge \exists q \in F_a\, p_0 u \underset{T(M)}{\Rightarrow} vqw\},$$

where $\overleftarrow{v}$ is the greatest suffix in P^* of v, and $\overrightarrow{w}$ is the greatest prefix in P^* of w. The transition graphs of labelled Turing machines are the computation graphs of unlabelled Turing machines.

Theorem 4.8. *The family $TURING_T$ is the set of T-graphs isomorphic to the computation graphs of Turing machines.*

Proof. $\subseteq$: Let (M,C) be a labelled Turing machine.

Let T be its label alphabet, Q be its state alphabet and P be its tape set.

We have to construct a (unlabelled non deterministic) Turing machine N such that its computable graph $R(N)$ is isomorphic to $G(M,C)$.

Such an isomorphism is given by the mapping which associates to any configuration upv where $p \in Q$ and $u,v \in P_\square^*$ with $u(1), u(|u|) \neq \square$, the word $\&\bar{u}\bar{p}\bar{v}\$$ with $\bar{p}$ in a new alphabet $\bar{Q}$ in bijection with Q, $\bar{u}$ (resp. $\bar{v}$) are obtained from u (resp. v) by replacing $\square$ by a new symbol $\bar{\square}$, and $\&, \$$ are also new symbols.

So we have to construct a Turing machine N such that

$$u \underset{G(M,C)}{\overset{a}{\longrightarrow}} v \iff \&\bar{u}\$ \underset{R(N)}{\overset{a}{\longrightarrow}} \&\bar{v}\$.$$

For p_0 the initial state of N and f_a the unique final state for each label $a \in T$, we will construct N in such a way that

$$u \underset{T(M)}{\overset{\varepsilon}{\Rightarrow}} \underset{T(M)}{\overset{a}{\longrightarrow}} \underset{T(M)}{\overset{\varepsilon}{\Rightarrow}} v \iff p_0\&\bar{u}\$ \underset{T(N)}{\Rightarrow} \&\bar{v} f_a \$ \quad \text{for any } u,v \in C.$$

As C is a rational set of configurations, there is a finite $(Q \cup P_\square)$-automaton (G,i,F) recognizing $C : \mathscr{L}(G,i,F) = C$.

First, the Turing machine N checks that the input word (between & and \$) belongs to C:

$$\begin{aligned}
&p_0\& \longrightarrow i\&+ \\
&\quad sA \longrightarrow tA+ \quad \text{if } s \underset{G}{\overset{A}{\longrightarrow}} t \text{ and } A \in P \\
&\quad s\bar{\square} \longrightarrow t\square+ \quad \text{if } s \underset{G}{\overset{\square}{\longrightarrow}} t
\end{aligned}$$

$$
\begin{aligned}
s\bar{p} &\to t\bar{p}+ \quad \text{if } s \underset{G}{\overset{p}{\to}} t \text{ and } p \in Q \\
s\$ &\to \hat{\$}\Box- \quad \text{if } s \in F \\
\hat{A}B &\to \hat{B}A- \quad \text{if } A \in P_\Box \cup \{\$\} \text{ and } B \in P_\Box \\
\hat{A}\bar{p} &\to pA \quad \text{if } A \in P_\Box \cup \{\$\} \text{ and } p \in Q
\end{aligned}
$$

The last rule without $+$ and $-$ means that we do not move the tape head.

Now the machine N simulates any path $\overset{a}{\Rightarrow}$ of M:

$$
\begin{aligned}
pA &\to qB\delta \quad \text{if } pA \overset{\varepsilon}{\to} qB\delta \text{ is a rule of } M, \\
pA &\to q_aB\delta \quad \text{if } pA \overset{a}{\to} qB\delta \text{ is a rule of } M, \\
p_aA &\to q_aB\delta \quad \text{if } pA \overset{\varepsilon}{\to} qB\delta \text{ is a rule of } M.
\end{aligned}
$$

Furthermore N must push the endmarkers & and \$ when they are accessible: for any $p \in Q \cup \bigcup_{a \in T} Q_a$

$$
\begin{aligned}
p\$ &\to p'\Box+ \\
p'\Box &\to p\$- \\
p\& &\to p''\Box- \\
p''\Box &\to p\&+
\end{aligned}
$$

Then N removes the useless $\Box$ (on the right of & and on the left of \$) and add the label $a \in T$ after the right endmarker \$:

$$
\begin{aligned}
p_aA &\to A_a\bar{p}+ \quad \text{for any } p \in Q \text{ and } A \in P_\Box \cup \{\$\} \\
A_aB &\to B_aA+ \quad \text{for any } A, B \in P_\Box \cup \{\$\} \\
\$_a\Box &\to \$_a\Box- \\
\$_aA &\to \$'_aA+ \quad \text{for any } A \in P \cup \{\&\} \cup \bar{Q} \\
\$'_a\Box &\to a'\$+ \\
a'\Box &\to \ell a- \\
\ell A &\to \ell A- \quad \text{for any } A \in P_\Box \cup \bar{Q} \\
\ell\& &\to \ell'\Box+ \\
\ell'\Box &\to \ell'\Box+ \\
\ell' A &\to \ell'' A- \quad \text{for any } A \in P \cup \{\$\} \cup \bar{Q} \\
\ell''\Box &\to i'\&+
\end{aligned}
$$

At this step, N reaches a configuration of the form $\&v\bar{p}w\$a$ with $v(1), w(|w|) \neq \Box$ with a state i' (i is the initial state of the finite automaton G recognizing C) reading the first letter of $v\bar{p}$. It remains to test whether $vw \in C$, to replace $\square$ by $\bar{\square}$, to remove a and to reach f_a:

$$
\begin{aligned}
s'A &\rightarrow t'A+ && \text{if } s \underset{G}{\overset{A}{\rightarrow}} t \text{ and } A \in P \cup \bar{Q},\\
s'\square &\rightarrow t'\bar{\square}+ && \text{if } s \underset{G}{\overset{\square}{\rightarrow}} t,\\
s'\$ &\rightarrow s'\$+ && \text{if } s \in F,\\
s'a &\rightarrow f_a\square- && \text{if } s \in F \text{ and } a \in T.
\end{aligned}
$$

$\supseteq$: Let N be a (unlabelled nondeterministic) Turing machine.

Let P be its tape alphabet and p_0 be its initial state.

We have to construct a labelled Turing machine (M, C) such that its transition graph $G(M, C)$ is isomorphic to the computable graph $R(N)$ of N.

We take the rational language $C = i\&P^*\$$ where $i, \&, \$$ are new symbols. For the isomorphism, we take the bijection which associates to any $u \in P^*$ the word $i\&u\$ \in C$. So we have to construct a labelled Turing machine M such that

$$
u \underset{R(N)}{\overset{a}{\rightarrow}} v \quad \Leftrightarrow \quad i\&u\$ \underset{G(M,C)}{\overset{a}{\rightarrow}} i\&v\$
$$

or equivalently for any $u, v, w \in P^*$,

$$
\begin{aligned}
& p_0 u \underset{T(N)}{\Rightarrow} xvqwy \text{ for some } q \in F_a,\ x \in (P^*\square)^*\ \ y \in (\square P^*)^*\\
\Leftrightarrow\ \ & i\&u\$ \underset{T(M)}{\overset{\varepsilon}{\Rightarrow}} \underset{T(M)}{\overset{a}{\rightarrow}} \underset{T(M)}{\overset{\varepsilon}{\Rightarrow}} i\&vw\$.
\end{aligned}
$$

First, the machine M simulates N:

$$
\begin{aligned}
i\& &\overset{\varepsilon}{\rightarrow} p_0\&+\\
pA &\overset{\varepsilon}{\rightarrow} qB\delta && \text{if } pA \rightarrow qB\delta \text{ is a rule of } N\\
p\$ &\overset{\varepsilon}{\rightarrow} p'\square+\\
p'\square &\overset{\varepsilon}{\rightarrow} p\$-\\
p\& &\overset{\varepsilon}{\rightarrow} p''\square-\\
p''\square &\overset{\varepsilon}{\rightarrow} p\&+
\end{aligned}
$$

Then M does a transition by a when N has reached a final state for a. And M removes the useless right part (beginning by a $\square$) of its configuration:

$$
\begin{aligned}
pA &\overset{a}{\rightarrow} \ell A+ && \text{if } p \in F_a \text{ and } A \in P\\
p\$ &\overset{a}{\rightarrow} m\$- && \text{if } p \in F_a\\
p\,\square &\overset{a}{\rightarrow} j\,\square+ && \text{if } p \in F_a
\end{aligned}
$$

$$\begin{aligned}
\ell A &\xrightarrow{\varepsilon} \ell A+ &&\text{if } A \in P\\
\ell \$ &\xrightarrow{\varepsilon} m\$- &&\\
\ell\ \square &\xrightarrow{\varepsilon} j\ \square+ &&\\
jA &\xrightarrow{\varepsilon} j\square+ &&\text{if } A \in P_\square\\
j\$ &\xrightarrow{\varepsilon} k\square- &&\\
k\ \square &\xrightarrow{\varepsilon} k\square- &&\\
kA &\xrightarrow{\varepsilon} m'A+ &&\text{if } A \in P \cup \{\&\}\\
m'\square &\xrightarrow{\varepsilon} m\$- &&
\end{aligned}$$

Finally M removes the useless left part (ending by a $\square$) of its configuration, and reads the marker & at state i. We do not give this simple part which is similar to the previous one. $\square$

As for languages, the family of computation graphs does not change if we restrict to *deterministic Turing machines* (the set of rules is functional: there is no two rules with the same left hand side).

Theorem 4.9. *The family $TURING_T$ is the set of T-graphs isomorphic to the computation graphs of deterministic Turing machines.*

The transformation of a (nondeterministic) Turing machine to a deterministic Turing machine with the same computation graph is similar to the usual transformation preserving the recognized language.

5. Conclusion

We have presented a hierarchy of graph families and essentially the family *TURING* of transition graphs of labelled Turing machines with ε-rules. In particular REC_{Rat} is the family of transition graphs of pushdown automata with ε-rules. Between the lowest family *FIN* of finite graphs and the greatest family *TURING*, we can show that the two families REC_{Rat} and *RAT* are natural, by considering the Cayley graphs of the word rewriting systems [8,12,14]. Finally and using traces, the hierarchy *FIN*, REC_{Rat}, *RAT*, *TURING* yields a Chomsky hierarchy. Another point is to find a subclass of word rewriting systems such that their transition graphs are the graphs of REC_{Lin}.

References

[1] A. Arnold, M. Nivat, Comportements de processus, Colloque AFCET ‘Les mathématiques de l’informatique’, 1982, pp. 35–68.

[2] K. Barthelmann, When can an equational simple graph be generated by hyperedge replacement, in: L. Brim, J. Gruska, J. Zlatuska (Eds.), 23rd MFCS, Lecture Notes in Computer Science, Vol. 1450, Springer, Berlin, 1998, pp. 543–552.

[3] M. Benois, Parties rationnelles du groupe libre, C.R. Académie Sci. Paris, Sér. A 269 (1969) 1188–1190.

[4] J. Berstel, in: Teubner (Ed.), Transductions and Context-Free Languages, Stuttgart, 1979.

[5] A. Blumensath, E. Grädel, Automatic structures, 15th LICS, IEEE, 2000, pp. 51–62.

[6] R. Book, F. Otto, String-rewriting systems, in: P. Gries (Ed.), Texts and Monographs in Computer Science, Springer, Berlin, 1993.

[7] R. Büchi, Regular canonical systems, Arch. Math. Logik Grundlag. 6 (1964) 91–111 [in: S. Mac Lane, D. Siefkes (Eds.), The Collected Works of J. Richard Büchi, Springer, New York, 1990, pp. 317–337].

[8] H. Calbrix, T. Knapik, A string-rewriting characterization of Muller and Schupp's context-free graphs, in: V. Arvind, R. Ramanujam (Eds.), 18th FSTTCS, Lecture Notes in Computer Science, Vol. 1530, Springer, Berlin, 1998, pp. 331–342.

[9] D. Caucal, On the regular structure of prefix rewriting, in: A. Arnold (Ed.), 15th CAAP, Lecture Notes in Computer Science, Vol. 431, Springer, Berlin, 1990, pp. 87–102 [a full version is in Theoretical Computer Science 106 (1992) 61–86].

[10] D. Caucal, Bisimulation of context-free grammars and of pushdown automata, CSLI, Vol. 53, in: A. Ponse, M. de Rijke, Y. Venema (Eds.), Modal Logic and Process Algebra, Stanford, 1995, pp. 85–106.

[11] D. Caucal, On infinite transition graphs having a decidable monadic theory, in: F. Meyer auf der Heide, B. Monien (Eds.), 23rd ICALP, Lecture Notes in Computer Science, Vol. 1099, Springer, Berlin, 1996, pp. 194–205, [Theoret. Comput. Sci. 290 (2003) 79–115].

[12] D. Caucal, On word rewriting systems having a rational derivation, in: J. Tiuryn (Ed.), 3rd FOSSACS, Lecture Notes in Computer Science, Vol. 1784, Springer, Berlin, 2000, pp. 48–62.

[13] D. Caucal, T. Knapik, An internal presentation of regular graphs by prefix-recognizable ones, Theory Comput. Systems 34 (4) (2001) 299–336.

[14] D. Caucal, T. Knapik, On a Chomsky-like hierarchy of infinite graphs, in: K. Diks, W. Rytter (Eds.), 27th MFCS, Lecture Notes in Computer Science, Lecture Notes in Computer Science, Vol. 2420, Springer, Berlin, 2002, pp. 177–187.

[15] B. Courcelle, Graph rewriting: an algebraic and logic approach, in: J. Leeuwen (Ed.), Handbook of Theoretical Computer Science, Vol. B, Elsevier, Amsterdam, 1990, pp. 193–242.

[16] N. Dershowitz, J.-P. Jouannaud, Rewrite systems, in: J. Leeuwen (Ed.), Handbook of Theoretical Computer Science, Vol. B, Elsevier, Amsterdam, 1990, pp. 243–320.

[17] C. Elgot, J. Mezei, On relations defined by generalized finite automata, IBM J. Res. Dev. 9 (1965) 47–68.

[18] C. Frougny, J. Sakarovitch, Synchronized rational relations of finite and infinite words, Theoret. Comput. Sci. 108 (1993) 45–82.

[19] A. Mateescu, A. Salomaa, Aspects of classical language theory, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer, Berlin, 1997, pp. 175–251.

[20] C. Morvan, On rational graphs, in: J. Tiuryn (Ed.), 3rd FOSSACS, Lecture Notes in Computer Science, Vol. 1784, Springer, Berlin, 2000, pp. 252–266.

[21] C. Morvan, C. Stirling, Rational graphs trace context-sensitive languages, in: P. Kolman, A. Pultr, J. Sgall (Eds.), 26th MFCS, Lecture Notes in Computer Science, Vol. 2136, Springer, Berlin, 2001, pp. 548–559.

[22] D. Muller, P. Schupp, The theory of ends, pushdown automata, and second-order logic, Theoret. Comput. Sci. 37 (1985) 51–75.

[23] E. Payet, Produit synchronisé pour quelques classes de graphes infinis, Ph.D. Thesis, University of La Réunion, 2000.

[24] C. Rispal, The synchronized graphs trace the context-sensitive languages, DEA Rep. University of Rennes, 2001.

[25] T. Urvoy, Regularity of congruential graphs, in: M. Nielsen, B. Rovan (Eds.), 25th MFCS, Lecture Notes in Computer Science, Vol. 1893, Springer, Berlin, 2000, pp. 680–689.