Linearly Bounded Infinite Graphs

Arnaud Carayol[1] and Antoine Meyer[1,2]

[1] Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France,
{Arnaud.Carayol,Antoine.Meyer}@irisa.fr
[2] Liafa, Université Denis Diderot, Case 7014, 2 place Jussieu, F-75251 Paris Cedex 05, France, Antoine.Meyer@liafa.jussieu.fr

Abstract. Linearly bounded Turing machines have been mainly studied as acceptors for context-sensitive languages. We define a natural family of canonical infinite automata representing their observable computational behavior, called linearly bounded graphs. These automata naturally accept the same languages as the linearly bounded machines defining them. We present some of their structural properties as well as alternative characterizations in terms of rewriting systems and context-sensitive transductions. Finally, we compare these graphs to rational graphs, which are another family of automata accepting the context-sensitive languages, and prove that in the bounded-degree case, rational graphs are a strict sub-family of linearly bounded graphs.

1 Introduction

One of the cornerstones of formal language theory is the hierarchy of languages introduced by Chomsky in [Cho59]. It rests on the definition of four increasingly restricted families of grammars, which respectively generate the *recursively enumerable*, *context-sensitive*, *context-free* and *regular* languages. All were extensively studied, and were given several alternative characterizations using different kinds of formalisms (or *acceptors*). For instance, pushdown systems characterize context-free languages, and linearly bounded Turing machines (LBMs) characterize context-sensitive languages. More recently, several authors have related these four families of languages to families of infinite graphs (see for instance [Tho01]). Given a fixed initial vertex and a set of final vertices, one can associate a language to a graph by considering the set of all words labeling a path between the initial vertex and one of the final vertices. In [CK02], a summary of four families of graphs accepting the four families of the Chomsky hierarchy is presented. They are the Turing graphs [Cau03b], rational graphs [Mor00,MS01], prefix-recognizable graphs [Cau96,Cau03a] and finite graphs.

Several approaches exist to define families of infinite graphs, among which we will cite three. The first one is to consider the finite acceptor of a language, and to build a graph representing the structure of its computations: vertices represent configurations, and each edge reflects the observable effect of an input on the configuration. One speaks of the *transition graph* of the acceptor. An interesting consequence is that the language of the graph can be deduced from the language of the acceptor it was built from. A second method proposed in [CK02] is to consider the Cayley-type graphs of some families of word rewriting systems. Each vertex is a normal form for a given rewriting system, and an edge between two vertices represents the addition of a letter and re-normalization by the rewriting system. Finally, a third possibility is to directly define the edge relations in a graph using automata or other formalisms. One speaks of derivation, transduction or computation graphs. In this approach, a path no longer represents a run of an acceptor, but rather a composition of binary relations.

Both prefix-recognizable graphs and Turing graphs have alternative definitions along all three approaches. Prefix-recognizable graphs are defined as the graphs of recognizable prefix relations. In [Sti00], Stirling presented them as the transition graphs of pushdown systems. It was also proved that they coincide with the Cayley-type graphs of prefix rewriting systems. As for Turing graphs, Caucal showed that they can be seen indifferently as the transition and computation graphs of Turing machines [Cau03b]. They are also the Cayley-type graphs of unrestricted rewriting systems. Rational graphs, however, are only defined as transduction graphs (using rational transducers) and as the Cayley-type graphs of left-overlapping rewriting systems, and lack a characterization as transition graphs. In this paper, we are thus interested in defining a suitable notion of transition graphs of linearly bounded Turing machines, and to determine some of their structural properties as well as to compare them with rational graphs.

As in [Cau03b] for Turing machines, we define a labeled version of LBMs, called LLBMs. Their transition rules are labeled either by a symbol from the input alphabet or by a special symbol denoting an unobservable transition. Following an idea from [Sti00], we consider that in every configuration of a LLBM, either internal actions or inputs are allowed, but not both at a time. This way, we can distinguish between internal and external configurations. The transition graph of a LLBM is the graph whose vertices are external configurations, and whose edges represent an input followed by a finite number of silent transitions. This definition is purely structural and associates a unique graph to a given LLBM. For convenience, we call such graphs *linearly bounded graphs*. To our knowledge, the notion of transition graph of a LBM was never considered. A similar work was proposed in [KP99,Pay00], where the family of configuration graphs of LBMs up to weak bisimulation is studied. However, it provides no formal definition associating LBMs to a family of *real-time* graphs (without edges labeled by silent transitions) representing their observable computations.

To further illustrate the suitability of our notion, we provide two alternative definitions of linearly bounded graphs. First, we prove that they are isomorphic to the Cayley-type graphs of length-decreasing rewriting systems. The second alternative definition directly represents the edge relations of a linearly bounded graph as a certain kind of context-sensitive transductions. This allows us to straightforwardly deduce structural properties of linearly bounded graphs, like their closure under synchronized product (which was already known from [KP99]) and under restriction to a context-sensitive set of vertices. To conclude this study, we show that linearly bounded graphs and rational graphs form incomparable families, even in the finite degree case. However, bounded degree rational graphs are a strict sub-family of linearly bounded graphs.

A more complete study of this family of graphs, including the proofs of the results stated in this article can be found in [CM05b].

2 Preliminary Definitions

A labeled, directed and simple *graph* is a set $G \subseteq V \times \Sigma \times V$ with Σ is a finite set of labels and V a countable set of *vertices*. An element (s, a, t) of G is an *edge* of *source* s, *target* t and *label* a, and is written $s \xrightarrow[G]{a} t$ or simply $s \xrightarrow{a} t$ if G is understood. The set of all sources and targets of a graph is its *support* V_G. A sequence of edges $s_1 \xrightarrow{a_1} t_1, \ldots, s_k \xrightarrow{a_k} t_k$ with $\forall i \in [2, k],\ s_i = t_{i-1}$ is a *path*. It is written $s_1 \xrightarrow{u} t_k$, where $u = a_1 \ldots a_k$ is the corresponding *path label*. A graph is *deterministic* if it contains no pair of edges with the same source and label. One can relate a graph to a languages by considering its path language, defined as the set of all words labeling a path between two given sets of vertices.

Definition 2.1. *The (path) language of a graph G between two sets of vertices I and F is the set* $L(G, I, F) \ = \ \{\ w \mid s \xrightarrow[G]{w} t,\ s \in I,\ t \in F\}$.

Linearly bounded Turing machines. We now recall the definition of context-sensitive languages and linearly bounded Turing machines. A context-sensitive language is a set of words generated by a grammar whose production rules are of the form $\alpha \to \beta$ with $|\beta| \geq |\alpha|$. Such grammars are called *context-sensitive*.

A more operational definition of context-sensitive languages is as the family of languages accepted by *linearly bounded Turing machines* (LBMs). Informally, a LBM is a Turing machine accepting each word w of its language using at most $k.|w|$ tape cells, where k is a fixed constant. Without loss of generality, one usually considers k to be equal to 1. Note that, contrary to unbounded Turing machines, it is sufficient to only consider linearly bounded machines which always terminate, also called *quasi-real time* machines. An interesting open problem raised by Kuroda [Kur64] concerns deterministic context-sensitive languages, which are the languages accepted by deterministic LBMs. It is not known whether they coincide with non-deterministic context-sensitive languages, as is the case for recursively enumerable or rational languages.

Rational graphs. Consider the product monoid $\Sigma^* \times \Sigma^*$, whose elements are pairs of words (u, v) in Σ^*, and whose composition law is defined by $(u_1, v_1) \cdot (u_2, v_2) = (u_1 u_2, v_1 v_2)$. A finite transducer is an automaton over $\Sigma^* \times \Sigma^*$ with labels in $(\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\})$. Transducers accept the rational subsets of $\Sigma^* \times \Sigma^*$, which are seen as binary relations on words and called rational transductions. We do not distinguish a transducer from the relation it accepts and write $(w, w') \in T$ if the pair (w, w') is accepted by T. Graphs whose vertices are words and whose edge relations is defined by transducers (one per letter in the label alphabet) are called rational graphs.

Definition 2.2 ([Mor00]). *A rational graph labeled by Σ with vertices in Γ^* is given by a tuple of transducers $(T_a)_{a\in\Sigma}$ over Γ. For all $a \in \Sigma$, $(u, a, v) \in G$ if and only if $(u, v) \in T_a$.*

For $w \in \Sigma^+$ and $a \in \Sigma$, we write $T_{wa} = T_w \circ T_a$, and $u \xrightarrow{w} v$ if and only if $(u, v) \in T_w$. In general, there is no bound on the size difference between input and output in a transducer (and hence between the lengths of two adjacent vertices in a rational graph). Interesting subclasses are obtained by enforcing some form of synchronization. The most well-known was defined by Elgot and Mezei [EM65] as follows. A transducer over Σ with initial state q_0 is (left-)synchronized if for every path $q_0 \xrightarrow{x_0/y_0} q_1 \ldots q_{n-1} \xrightarrow{x_n/y_n} q_n$, there exists $k \in [0, n]$ such that for all $i \in [0, k-1]$, x_i and y_i belongs to Σ and either $x_k = \ldots = x_n = \varepsilon$ or $y_k = \ldots = y_n = \varepsilon$. A rational graph defined by synchronized transducers will simply be called a synchronized (rational) graph.

3 Linearly Bounded Graphs

3.1 LBM Transition Graphs

Following [Cau03b], we define the notion of labeled linearly bounded Turing machine (LLBM). As in standard definitions of LBMs, the transition rules can only move the head of the LLBM between the two end markers [and]. In addition, a silent step can decrease the size of the configuration (without removing the markers) and a Σ-transition can increase the size of the configuration by one cell. This ensures that while reading a word of length n, the labeled LBM uses at most n cells.

Definition 3.1. *A labeled linearly bounded Turing machine is a tuple $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$, where Γ is a finite set of* tape symbols*, $\Sigma \subseteq \Gamma$ is the* input alphabet*, [and] $\notin \Gamma$ are the* left *and* right end-marker*, Q is a finite set (disjoint from Γ) of* control states*, $q_0 \in Q$ is the unique* initial state*, $F \subseteq Q$ is a set of* final states *and δ is a finite set of labeled* transition rules *of one of the forms:*

$$pA \xrightarrow{\varepsilon} qB\pm \qquad p[\xrightarrow{\varepsilon} q[+ \qquad p] \xrightarrow{\varepsilon} q]-$$
$$pB \xrightarrow{a} qAB \qquad p] \xrightarrow{a} qA] \qquad pA \xrightarrow{\varepsilon} q$$

with $p, q \in Q$, $A, B \in \Gamma$, $\pm \in \{+, -\}$ and $a \in \Sigma$.

The set of configurations C_M of M is the set of words uqv such that $q \in Q$, $v \neq \varepsilon$ and $uv \in [\Gamma^*]$. For all $x \in \Sigma \cup \{\varepsilon\}$, the transition relation $\xrightarrow[M]{x}$ is a subset of $C_M \times C_M$ defined as

$$\begin{aligned} \xrightarrow[M]{x} = &\{\ (upAv, uBqv) \mid pA \xrightarrow{x} qB+ \in \delta\ \} \\ &\cup \{\ (uCpAv, uqCBv) \mid pA \xrightarrow{x} qB- \in \delta\} \\ &\cup \begin{cases} \{\ (upAv, uqv) \mid pA \xrightarrow{x} q \in \delta\} & \text{with } x = \varepsilon \\ \{\ (upAv, uqBAv) \mid pA \xrightarrow{x} qBA \in \delta\} & \text{with } x \in \Sigma. \end{cases} \end{aligned}$$

We will simply write $\xrightarrow{x}$ when M is understood. As usual, we define $\xrightarrow{wx}$ as $(\xrightarrow{w} \circ \xrightarrow{x})$ for all $w \in (\Sigma \cup \{\varepsilon\})^*$. The unique initial configuration is $[q_0]$ and a final configuration c_f is a configuration containing a terminal control state. A word w is accepted by M if $[q_0] \xrightarrow{w} c_f$ where c_f is a final configuration. Quite naturally, M is *deterministic* if, from any configuration, either all possible moves are labeled by *distinct* letters of Σ, or there is only one possible move. Formally, it means that for all configurations c, c_1, c_2 with $c_1 \neq c_2$, if $c \xrightarrow{a} c_1$ and $c \xrightarrow{b} c_2$ then $a \neq b$, $a \neq \varepsilon$ and $b \neq \varepsilon$.

Remark 3.1. For convenience, one may consider LBMs whose initial configuration is not of the form $[q_0]$ but is any fixed configuration c_0. This does not add any expressive power, as can be proved by a simple encoding of c_0 into the control state set of the machine.

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, F, [,])$ be a LLBM, we define its configuration graph

$$C_M \;=\; \big\{(c, a, c') \mid c \xrightarrow[M]{a} c' \quad \text{for } a \in \Sigma \cup \{\varepsilon\}\,\big\}.$$

The vertices of this graph are all configurations of M, and its edges denote the transitions between them, including ε-transitions. One may wish to only consider the behavior of M from an external point of view, i.e. only looking at the sequence of inputs. This means one has to find a way to conceal ε-transitions without changing the accepted language or destroying the structure. One speaks of the *transition graph* of an acceptor, as opposed to its configuration graph.

In [Sti00], Stirling mentions a normal form for pushdown automata which allows him to consider a structural notion of transition graphs, without relying on the naming of vertices. We first recall this notion of *normalized* systems adapted to labeled LBMs. A labeled LBM is *normalized* if its set of control states can be partitioned in two subsets: one set of *internal* states, noted Q_ε, which can always and only perform ε-rules, and a set of *external* states noted Q_Σ, which can only perform Σ-rules. More formally:

Definition 3.2. *A labeled LBM $M = (Q, \Sigma, \Gamma, \delta, q_0, F, [,])$ is* normalized *if there are disjoint sets Q_Σ and Q_ε such that $Q = Q_\varepsilon \cup Q_\Sigma$, $F \subseteq Q_\Sigma$, and*

$$\begin{aligned}
&pB \xrightarrow{a} qAB \in \delta \implies p \in Q_\Sigma,\\
&pA \xrightarrow{\varepsilon} qB\pm \in \delta \implies p \in Q_\varepsilon,\\
&p \in Q_\varepsilon \implies \exists\ pA \xrightarrow{\varepsilon} qB\pm \in \delta.
\end{aligned}$$

This definition implies in particular that a control states from which there exists no transition must belong to Q_Σ. A configuration is external if its control state is in Q_Σ, and internal otherwise. This makes it possible to *structurally* distinguish between internal vertices, which have one or more outgoing ε-edges, and external ones which only have outgoing Σ-edges or have no outgoing edges. Given any labeled LBM, it is always possible to normalize it without changing the accepted language.

From this point on, unless otherwise stated, we will only consider normalized LLBMs. We can now define our notion of LLBM transition graph as the ε-closure of its configuration graph, followed by a restriction to its set of external configurations (which happens to be a rational set).

Definition 3.3. *Let $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ be a (normalized) LLBM, and C_Σ be its set of external configurations. The transition graph of M is*

$$G_M \;=\; \big\{(c, a, c') \mid c, c' \in C_\Sigma,\; a \in \Sigma,\; \wedge\; c \xrightarrow[M]{a\varepsilon^*} c' \,\big\}.$$

We now define the family of linearly bounded graphs as the closure under isomorphism of transition graphs of labeled LBMs, i.e. as the set of all graphs which can be obtained by renaming the vertices of a LLBM transition graph.

3.2 Alternative definitions

This section provides two alternative definitions of linearly bounded graphs. In [CK02], it is shown that all previously mentioned families of graphs can be expressed in a uniform way in terms of Cayley-type graphs of certain families of rewriting systems. We show that it is also the case for linearly bounded graphs, which are the Cayley-type graphs of length-decreasing rewriting systems. The second alternative definition we present changes the perspective and directly defines the edges of linearly bounded graphs using incremental context-sensitive transductions. This variety of definitions will allow us to prove in a simpler way some of the properties of linearly bounded graphs.

Cayley-type graphs of decreasing rewriting systems. We first give the relevant definitions about rewriting systems and Cayley-type graphs. A *word rewriting system* R over alphabet Γ is a subset of $\Gamma^* \times \Gamma^*$. Each element $(l, r) \in R$ is called a *rewriting rule* and noted $l \rightarrow r$. The words l and r are respectively called the left-hand and right-hand side of the rule. The *rewriting relation* of R is the binary relation $\{(ulv, urv) \mid u, v \in \Gamma^*,\; l \rightarrow r \in R\}$ which we also denote by R, consisting of all pairs of words (w_1, w_2) such that w_2 can be obtained by replacing (*rewriting*) an instance of a left-hand side l in w_1 with the corresponding right-hand side r. The reflexive and transitive closure R^* of this relation is called the *derivation* of R. Whenever for some words u and v we have uR^*v, we say R rewrites u into v. A word which contains no left-hand side is called a *normal form*. The set of all normal forms of R is written $\mathrm{NF}(R)$.

One can associate a unique infinite graph to any rewriting system by considering its *Cayley-type graph* defined as follows:

Definition 3.4. *The Σ-labeled Cayley-type graph of a rewriting system R over Γ, with $\Sigma \subseteq \Gamma$, is the infinite graph*

$$G_R \;=\; \{(u, a, v) \mid a \in \Sigma,\; u, v \in \mathrm{NF}(R),\; uaR^*v\}.$$

The family of rewriting systems we consider is the family of *finite length-decreasing word rewriting systems*, i.e. rewriting systems with a finite set of rules of the form $l \to r$ with $|l| \geq |r|$, which can only preserve or decrease the length of the word to which they are applied. The reason for this choice is that the derivation relations of such systems coincide with arbitrary compositions of labeled LBM ε-rules.

Theorem 3.1. *The two families of linearly bounded graphs and of Cayley-type graphs of decreasing rewriting systems are equal up to isomorphism.*

Incremental context-sensitive transduction graphs. The notion of *computation graph* was first introduced in early versions of [Cau03b] and systematically used in [CK02]. It corresponds to the graphs defined by the *transductions* (i.e. binary relations on words) associated to a family of finite machines. These works prove that for pushdown automata and Turing machines, the classes of transition and computation graphs coincide. We show that it is also the possible to give a definition of linearly bounded graphs as the computation graphs of a certain family of LBMs, or equivalently as the graphs defined by a certain family of context-sensitive transductions.

A relation R is recognized by a LBM M if the language $\{u\#v \mid (u,v) \in R\}$ where # is a fresh symbol is accepted by M. However, this type of transductions generates more than linearly bounded graphs. Even if we only consider linear relations (i.e relations R such that there exists c and $k \in \mathbb{N}$ such that $(u,v) \in R$ implies $|v| \leq c \cdot |u| + k$), we obtain graphs accepting the languages recognizable in exponential space (EXPSPACE) which strictly contain the context-sensitive languages [Imm88]. We need to consider relations for which the length difference between a word and its image is bounded by a certain constant. Such relations can be associated to LBMs.

Definition 3.5. *A* k-incremental context-sensitive transduction T *over* Γ *is defined by a LBM recognizing a language* $L = \{u\#v \mid u, v \in \Gamma^* \text{ and } |v| \leq |u|+k\}$ *where* # *does not belong to* Γ. *The relation* T *is defined as* $\{(u,v) \mid u\#v \in L\}$.

The following proposition states that incremental context-sensitive transductions form a boolean algebra.

Proposition 3.1. *For all* k*-incremental context-sensitive transductions* T *and* T' *over* Γ^*, $T \cup T'$, $T \cap T'$ *and* $\overline{T} = E - T$ *(where* E *is* $\{(u,v) \mid 0 \leq |v| \leq |u|+k\}$*) are incremental context-sensitive transductions.*

The canonical graph associated to a finite set of transductions is called a *transduction graph*. Relating graphs to a family of binary relations on words was already used to define rational graphs and their sub-families.

Definition 3.6. *The* Σ*-labeled transduction graph* G_T *of a finite set of incremental context-sensitive transductions* $(T_a)_{a \in \Sigma}$ *is*

$$G_T \;=\; \{(u,a,v) \mid a \in \Sigma \text{ and } (u,v) \text{ is recognized by } T_a\}.$$

Length-preserving context-sensitive transductions have already been extensively studied in [LST98]. In the rest of this presentation, unless otherwise stated, we will only consider 1-incremental transductions without loss of generality regarding the obtained family of graphs.

Theorem 3.2. *The families of linearly bounded graphs and of incremental context-sensitive transduction graphs are equal up to isomorphism.*

3.3 Structural properties

Languages. It is quite obvious that the language of the transition graph of a LLBM M between the vertex representing its initial configuration and the set of vertices representing its final configurations is the language of M. In fact, the choice of initial and final vertices has no importance in terms of the family of languages one obtains.

Proposition 3.2. *The languages of linearly bounded graphs between an initial vertex i and a finite set F of final vertices are the context-sensitive languages.*

Remark 3.2. When a linearly bounded graph is explicitly seen as the transition graph of a LLBM, as a Cayley-type graph or as a transduction graph, i.e. when the naming of its vertices is fixed, considering context-sensitive sets of final vertices does not increase the accepted family of languages.

Closure properties. Linearly bounded graphs enjoy several good properties, which will be especially important when comparing this class to other families of graphs related to LBMs or context-sensitive languages (see Section 4).

Proposition 3.3. *The family of linearly bounded graphs is closed under restriction to reachable vertices from any vertex and under restriction to a context-sensitive set of vertices.*

Since all rational languages are context-sensitive, linearly bounded graphs are also closed under restriction to a rational set of vertices. This shows that it is not necessary to allow arbitrary rational restrictions in the definition of transition graphs of linearly bounded machine, since such a restriction can be directly applied to the set of external configurations of a machine. By a slight adaptation of the proofs used in [KP99], one also gets the result below.

Proposition 3.4 ([KP99]). *Linearly bounded graphs are closed under synchronized product.*

Deterministic linearly bounded graphs. It is straightforward to notice that there exist non-deterministic labeled LBMs whose transition graphs are deterministic, and we do not know whether, for all non-deterministic labeled LBM whose transition graph is deterministic, it is possible to build an equivalent deterministic labeled LBM, possibly having the same transition graph. In fact, we can show that, for any context-sensitive language, it is always possible to build a deterministic linearly bounded graph accepting it.

Proposition 3.5. *For all context-sensitive language L, there exists a* deterministic *linearly bounded graph G, a vertex i and a rational set of vertices F of G such that $L = L(G, \{i\}, F)$.*

We are of course not able to conclude that the languages of deterministic transition graphs of labeled LBMs are the deterministic context-sensitive languages, because it would imply that deterministic and non-deterministic context-sensitive languages coincide. However, if we only consider quasi real-time linearly bounded machines, which have no infinite run on any given input word, the family of transition graphs we obtain faithfully illustrates the determinism of the languages.

Proposition 3.6. *The languages of deterministic transition graphs of quasi real-time LBMs are the deterministic context-sensitive languages.*

4 Comparison with rational graphs

We will now give some remarks about the comparison between linearly bounded graphs and several different sub-families of rational graphs. First note that since linearly bounded graphs have by definition a finite degree, it is more relevant to only consider rational graphs of finite degree. However, even under this structural restriction, rational and linearly-bounded graphs are incomparable, due to the incompatibility in the growth rate of their vertices degrees.

In a rational graph the out-degree at distance n from any vertex can be c^{c^n}, whereas in a linearly bounded graph is at most c^n for some c.

Lemma 4.1. *For any linearly bounded graph L and any vertex x, there exists $c \in \mathbb{N}$ such that the out-degree of L at distance $n > 0$ of x is at most c^n.*

Figure 1 shows a rational graph whose vertices at distance n from the root A have out-degree $2^{2^{n+1}}$. This graph is thus not linearly bounded.

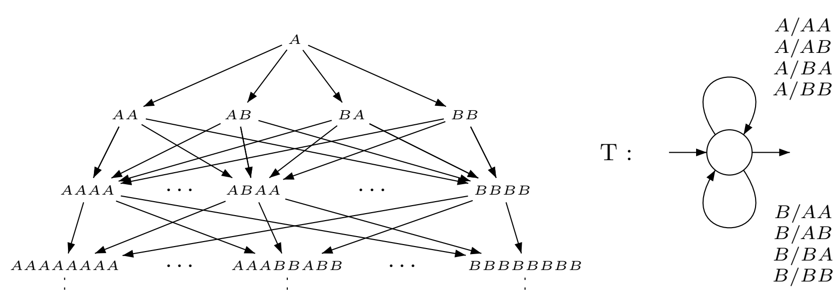

Fig. 1. A finite degree rational graph (together with its transducer) which is not isomorphic to any linearly bounded graph.

Conversely, in a rational graph of finite degree, the in-degree at distance n from any vertex is at most c^{c^n} for some $c \in \mathbb{N}$, in a linearly bounded graph it can be as large as $f(n)$ for any mapping f from $\mathbb{N}$ to $\mathbb{N}$ recognizable in linear space (i.e. such that the language $\{0^n 1^{f(n)} \mid n \in \mathbb{N}\}$ is context-sensitive).

Lemma 4.2. *For any mapping $f : \mathbb{N} \mapsto \mathbb{N}$ recognizable in linear space, there exists a linearly bounded graph L with a vertex x such that the in-degree at distance $n > 0$ of x is $f(n)$.*

An instance of such a mapping is $f : n \mapsto 2^{2^{2^n}}$, which is more than the in-degree at distance n of a vertex in any rational graph of finite degree. From these two observations, we get the result below.

Proposition 4.1. *The families of finite degree rational graphs and of linearly bounded graphs are incomparable.*

Since finite-degree rational graphs and linearly bounded graphs are incomparable, we investigate more restricted sub-families of rational graphs. For synchronized graphs of finite out-degree, we have the following result.

Proposition 4.2. *The synchronized graphs of finite degree form a strict sub-family of linearly bounded graphs (up to isomorphism).*

Proof (Sketch). Synchronized transducers of finite image can only map together words whose length difference is at most some constant k. It can thus very easily be seen that synchronized rational relations of finite image are incremental context-sensitive transductions.

For the even more restricted family of bounded-degree rational graphs, we show the following comparison.

Theorem 4.1. *The rational graphs of bounded degree form a strict sub-family of linearly bounded graphs of bounded degree (up to isomorphism).*

Proof (Sketch). The inclusion is based on a uniformization result for rational relations of bounded image due to Weber [Web96], which states that they can be decomposed into a finite union of functional transductions. This allows us to propose a coding of the rational graph's vertices such that the edge relation of the obtained graph is a 1-incremental context-sensitive transduction. The idea of this coding is to identify a vertex either by its name in the rational graph, or by a unique path from another vertex, whichever is shortest. This allows to express the edge relation of the graph as a 1-incremental context-sensitive transduction.

As rational graphs are closed under edge reversal, an equality between the two families would imply that linearly bounded graphs of bounded degree are also closed under edge reversal, which can be proved wrong. □

It may be interesting at this point to recall that all existing proofs that the rational graphs accept the context-sensitive languages break down when the out-degree is bounded. It is thus not at all clear whether rational graphs of bounded degree accept all context-sensitive languages. However, as noted in 3.5, it is still the case for bounded degree linearly bounded graphs, and in particular for deterministic linearly bounded graphs.

5 Conclusion

This paper gives a natural definition of a family of canonical graphs associated to the observable computations of labeled linearly bounded machines. It provides equivalent characterizations of this family as the Cayley-type graphs of length-decreasing term-rewriting systems, and as the graphs defined by a subfamily of context-sensitive transductions which can increase the length of their input by at most a constant number of letters. Although of a sensibly different nature from rational graphs, we showed that all rational graphs of bounded degree are linearly bounded graphs of bounded degree, and that this inclusion is strict. This leads us to consider a more restricted notion of infinite automata, closer to classical finite automata (as was already observed in [CM05a]), and to propose a hierarchy of families of infinite graphs of bounded degree accepting the families of languages of the Chomsky hierarchy (see Fig. 2).

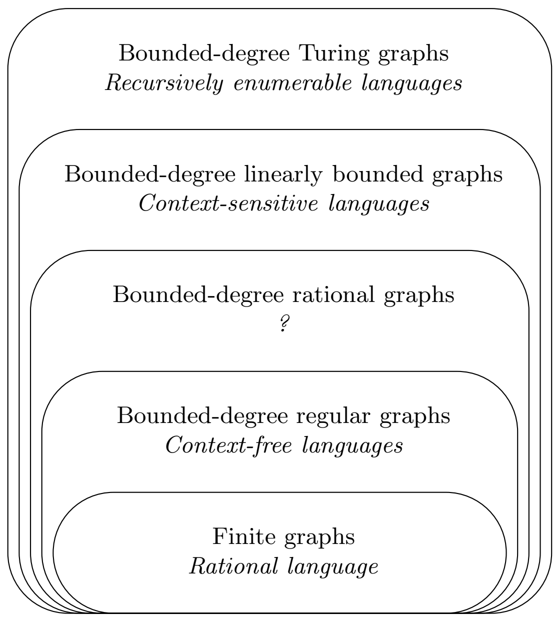

Fig. 2. A Chomsky-like hierarchy of bounded-degree infinite graphs.

Acknowledgment

The authors would like to thank Didier Caucal for his support.

References

[Cau96] D. Caucal. On infinite transition graphs having a decidable monadic theory. In *Proc. of ICALP '96*, volume 1099 of *LNCS*, pages 194–205. Springer Verlag, 1996.

[Cau03a] D. Caucal. On infinite transition graphs having a decidable monadic theory. *Theor. Comput. Sci.*, 290:79–115, 2003.

[Cau03b] D. Caucal. On the transition graphs of Turing machines. *Theor. Comput. Sci.*, 296:195–223, 2003.

[Cho59] N. Chomsky. On certain formal properties of grammars. *Inf. Control*, 2:137–167, 1959.

[CK02] D. Caucal and T. Knapik. A Chomsky-like hierarchy of infinite graphs. In *Proc. of MFCS '02*, volume 2420 of *LNCS*, pages 177–187. Springer Verlag, 2002.

[CM05a] A. Carayol and A. Meyer. Context-sensitive languages, rational graphs and determinism. Submitted to publication (a preliminary version of this article is available at `http://www.irisa.fr/galion/acarayol/`), 2005.

[CM05b] A. Carayol and A. Meyer. Linearly bounded infinite graphs (extended version). Submitted to publication (a preliminary version of this article is available at `http://www.irisa.fr/galion/acarayol/`), 2005.

[EM65] C. Elgot and J. Mezei. On relations defined by finite automata. *IBM J. Res. Develop.*, 9:47–68, 1965.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. *SIAM J. Comput.*, 17(5):935–938, 1988.

[KP99] T. Knapik and É. Payet. Synchronized product of linear bounded machines. In *Proc. of FCT '99*, volume 1684 of *LNCS*, pages 362–373. Springer Verlag, 1999.

[Kur64] S. Kuroda. Classes of languages and linear-bounded automata. *Inf. Control*, 7(2):207–223, 1964.

[LST98] M. Latteux, D. Simplot, and A. Terlutte. Iterated length-preserving rational transductions. In *Proc. of MFCS '98*, volume 1450 of *LNCS*, pages 286–295. Springer Verlag, 1998.

[Mor00] C. Morvan. On rational graphs. In *Proc. of FoSSaCS '00*, volume 1784 of *LNCS*, pages 252–266. Springer Verlag, 2000.

[MS01] C. Morvan and C. Stirling. Rational graphs trace context-sensitive languages. In *Proc. of MFCS '01*, volume 2136 of *LNCS*, pages 548–559. Springer Verlag, 2001.

[Pay00] É. Payet. *Thue Specifications, Infinite Graphs and Synchronized Product.* PhD thesis, Université de la Réunion, 2000.

[Sti00] C. Stirling. Decidability of bisimulation equivalence for pushdown processes. Technical Report EDI-INF-RR-0005, School of Informatics, University of Edinburgh, 2000.

[Tho01] W. Thomas. A short introduction to infinite automata. In *Proc. of DLT '01*, volume 2295 of *LNCS*, pages 130–144. Springer Verlag, 2001.

[Web96] A. Weber. Decomposing a k-valued transducer into k unambiguous ones. *Inf. Théor. Appl.*, 30(5):379–413, 1996.