

# Languages of tree-automatic graphs

Antoine Meyer

Institute of Mathematical Sciences,  
Chennai, India

*Journées Montoises 2006, Irisa, Rennes*

# Outline

- ① Graphs and languages
- ② Languages of rational graphs
- ③ Languages of term-automatic graphs
- ④ Future work

# Outline

- ① Graphs and languages
- ② Languages of rational graphs
- ③ Languages of term-automatic graphs
- ④ Future work

# Graphs and languages

- Graph: countable set of edges  $u \xrightarrow{a} v$   
(up to isomorphism)

- Language of a graph  $G$  between two sets  $I$  and  $F$ :

$$L(G, I, F) = \{w \mid \exists i \in I, f \in F, i \xrightarrow{w} f\}$$

- Parallel between classes of languages and classes of (infinite) graphs

## A hierarchy of infinite automata

Graphs	Languages
Finite	<i>Regular</i>
Pushdown, Regular Prefix-recognizable	<i>Context-free</i>
Pushdown( $n$ ) Prefix-recognizable( $n$ )	<i>OI-languages(<math>n</math>)</i>
Automatic / Rational Linearly bounded	<i>Context-sensitive</i>

# A hierarchy of infinite automata

Classes of graphs defined by ...

<b>Relations on words</b>	<b>Relations on terms</b>
Prefix rewriting	Ground term rewriting
Automatic relations	Term-automatic relations
Rational relations	?

This work: languages of term-automatic graphs

# Outline

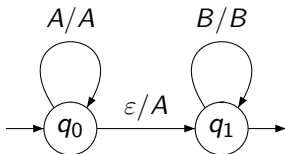
- ① Graphs and languages
- ② Languages of rational graphs
- ③ Languages of term-automatic graphs
- ④ Future work

# Rational relations

## Definition

A binary relation over words is called **rational** if it is the set of pairs accepted by a **finite transducer**

Example:



accepts the relation  $\{(A^n B^m, A^{n+1} B^m) \mid m, n \geq 0\}$



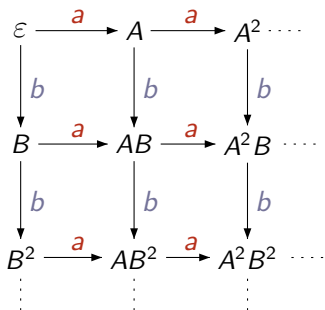
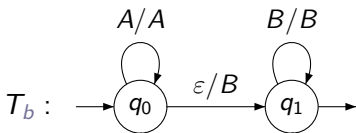
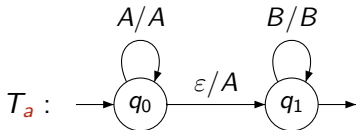
# Rational graphs

## Definition

A rational graph is a graph whose edge relations are rational

- Domain of vertices = words
- Edge relation for each label accepted by a transducer

Example:



## Subclasses of rational graphs

- Synchronized transducer: all runs of one the forms

$$\left\{ \begin{array}{l} q_0 \xrightarrow{a_1/b_1} \dots \xrightarrow{a_n/b_n} q_n \xrightarrow{\varepsilon/b_{n+1}} \dots \xrightarrow{\varepsilon/b_{n+k}} q_f \\ q_0 \xrightarrow{a_1/b_1} \dots \xrightarrow{a_n/b_n} q_n \xrightarrow{a_{n+1}/\varepsilon} \dots \xrightarrow{a_{n+k}/\varepsilon} q_f \end{array} \right.$$

- Automatic graph: defined by synchronized transducers (\*)
- Synchronous transducer: no  $\varepsilon$  appearing on any transition
- Synchronous graph: defined by synchronous transducers

# Languages of rational graphs

**Theorem** (Morvan, Stirling, Rispal)

*Rational and automatic graphs accept precisely the context-sensitive languages*

*Synchronous graphs accept precisely the context-sensitive languages (between regular sets of vertices)*

# Languages of rational graphs

- Initial proofs use the Penttonen normal form
  - Technically non-trivial
  - No link to complexity
  - No notion of determinism
- Recent contributions: (Carayol, M.)
  - Self-contained proof using tiling systems
  - Characterization of languages for sub-families of graphs
  - Characterization of graphs for sub-families of languages

# Tiling systems

## Definition

A framed tiling system  $\Delta$  is a finite set of  $2 \times 2$  pictures (tiles) with a border symbol  $\#$

- Picture: rectangular array of symbols
- Picture language of  $\Delta$ : set of all framed pictures with only tiles in  $\Delta$
- Word language of  $\Delta$ : set of all *first row contents* in the picture language of  $\Delta$

## Proposition (Lattaux, Simplot)

The languages of tiling systems are precisely the context-sensitive languages

# A tiling system

# # # a	# # a a	# # a b	# # b b	# # b #
# a # a	a a a a	a b ⊥ ⊥	b b b b	b # b #
# a # ⊥	a a a ⊥	⊥ ⊥ ⊥ ⊥	b b ⊥ b	b # ⊥ #
# ⊥ # #	a ⊥ ⊥ ⊥	⊥ ⊥ # #	⊥ b ⊥ ⊥	⊥ # # #

#####  
# a a a a a b b b b b #  
# a a a a ⊥ ⊥ b b b b #  
# a a a ⊥ ⊥ ⊥ ⊥ b b b #  
# a a ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ b b #  
# a ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ b #  
# ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ #  
#####

# A tiling system

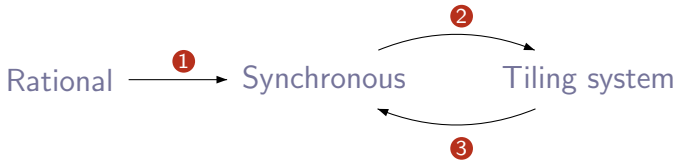
# # # a	# # a a	# # a b	# # b b	# # b #
# a # a	a a a a	a b ⊥ ⊥	b b b b	b # b #
# a # ⊥	a a a ⊥	⊥ ⊥ ⊥ ⊥	b b ⊥ b	b # ⊥ #
# ⊥ # #	a ⊥ ⊥ ⊥	⊥ ⊥ # #	⊥ b ⊥ ⊥	⊥ # # #

#####  
# a a a a b b b b #  
# a a a a ⊥ ⊥ b b b b #  
# a a a ⊥ ⊥ ⊥ ⊥ b b b #  
# a a ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ b b #  
# a ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ b #  
# ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ ⊥ #  
#####

# Proof technique

Proof in three steps:

- ① Trace-equivalence of rational and synchronous graphs
- ② Simulation of a synchronous graph by a tiling systems
- ③ Simulation of a tiling system by a synchronous graph



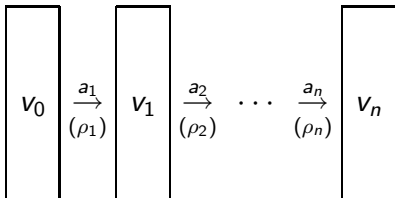
- ① relies on elimination of  $\varepsilon$  in transducers
- ② and ③ rely on identifying graph paths with pictures



# Synchronous graph $\leftrightarrow$ tiling system

## Proof idea

- Identify transducer runs and picture columns
- Establish a bijection between accepting paths and pictures
- Deduce a bijection between synchronous graphs and tiling systems



$\leftrightarrow$

$a_1$	$\dots$	$a_n$
$\rho_1$	$\dots$	$\rho_n$

# Outline

- ① Graphs and languages
- ② Languages of rational graphs
- ③ Languages of term-automatic graphs
- ④ Future work

# Languages of term-automatic graphs

## Theorem

*The following statements are equivalent:*

- ①  $L = L(G, I, F)$  for some term-automatic graph  $G$  and finite sets  $I$  and  $F$
- ②  $L = L(G, I, F)$  for some term-synchronous graph  $G$  and regular sets  $I$  and  $F$
- ③  $L$  is accepted by an arborescent tiling system
- ④  $L$  is accepted by an alternating linearly bounded machine
- ⑤  $L$  is in  $ETIME (= DTIME(2^{O(n)}) )$

# Term-automatic relations

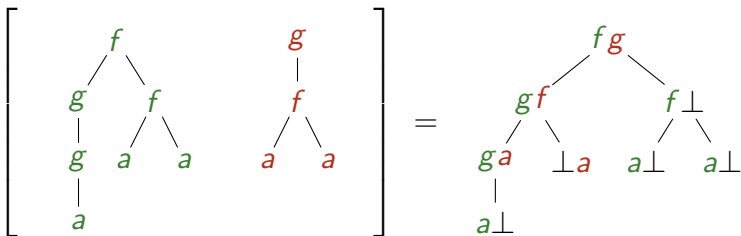
## Definition

Let  $s$ ,  $t$  be two terms,  $[st]$  denotes the term such that

- $\text{dom}([st]) = \text{dom}(s) \cup \text{dom}(t)$
- $[st](x) = fg$  with
$$\begin{cases} f = s(x) & \text{if } x \in \text{dom}(s), \perp \text{ otherwise} \\ g = t(x) & \text{if } x \in \text{dom}(t), \perp \text{ otherwise} \end{cases}$$

# Term-automatic relations

Example:



# Term-automatic relations

## Definition

- A binary relation  $R$  over terms is **automatic** if  $\{[st] \mid (s, t) \in R\}$  is regular  
(i.e. accepted by a finite tree automaton)
- A binary relation  $R$  over terms is **synchronous** if it is automatic and  $\forall (s, t) \in R, \text{dom}(s) = \text{dom}(t)$
- A graph is term-automatic (resp. synchronous) if its edge relations are automatic (resp. synchronous)

# Arborescent pictures

## Definition

An arborescent picture is a mapping  $P : X \times [1, n] \rightarrow \Gamma$  where

- $X \subseteq \mathbb{N}^*$  is a prefix- and left-closed set of positions
- $n$  is the width of  $P$
- $\Gamma$  is a finite alphabet

Remark:

$P$  isomorphic to a finite tree of domain  $X$  with labels in  $\Gamma^n$

# Arborescent tiling systems

## Definition

An arborescent tiling system  $\Delta$  is a set of arborescent pictures of height and width 2 (tiles) over  $\Gamma \cup \{\#\}$  (with  $\# \notin \Gamma$ )

- Picture language of  $\Delta$ : set of all framed arborescent pictures with tiles only in  $\Delta$
- Word language of  $\Delta$ : set of all *first row contents* in the picture language of  $\Delta$



# Linearly Bounded Machines

## Definition

Linearly Bounded Machine (LBM): Turing machine working in linear space

- Finite set of control states
- Fixed-size tape containing the input word
- Transitions: cell rewriting + left/right movement

$$pA \rightarrow qB+$$

$$pA \rightarrow qB-$$

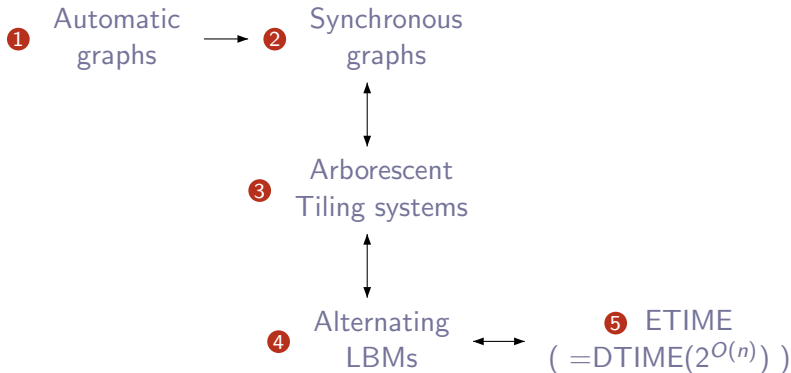
$$p[ \rightarrow q[+$$

$$p] \rightarrow q]-$$

- Alternation: combination of right-hand sides:

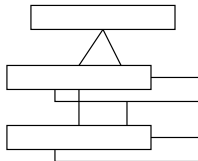
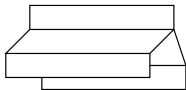
$$pA \rightarrow (qB+ \wedge q'C- \wedge q''D+)$$

# Equivalence proof



# Equivalence proof

- Term-automatic  $\rightarrow$  term-synchronous graphs:
  - Consider the padding symbol  $\perp$  as a new symbol
- Term-synchronous graphs  $\leftrightarrow$  alternating LBMs:
  - Same mathematical description for paths and LBM runs (arborescent pictures)
  - **Local** integrity constraints
  - Equivalence between alternation and branching



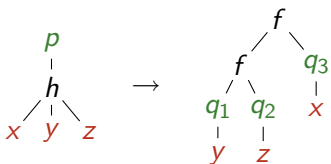
# Outline

- ① Graphs and languages
- ② Languages of rational graphs
- ③ Languages of term-automatic graphs
- ④ Future work

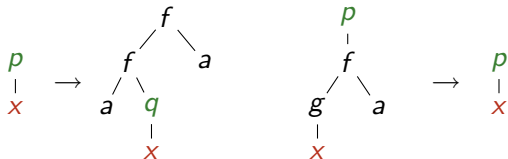
# Term-rational graphs?

Possible extension: graphs defined by tree transducers

- Problem 1: permutations between sub-terms



- Problem 2: arbitrary  $\varepsilon$ -transitions along runs



## Other questions

- Term-automatic graphs
  - Structural restrictions (in particular the degree)
  - Comparison with other classes (up to isomorphism)
- Arborescent tiling systems
  - Yields of regular sets of terms: context-free languages
  - ↳ Yields of arborescent pictures?