More general families of infinite graphs

Antoine Meyer

Formal Methods Update 2006
IIT Guwahati

Prefix-recognizable graphs

Theorem

Let G be a graph, the following statements are equivalent:

- *G is defined by relations of the form $(U \xrightarrow{a} V) \cdot \mathrm{Id}_W$*
- *G is the prefix rewriting graph of a recognizable rewriting system (+ regular restriction)*
- *G is the transition graph of a pushdown automaton with ε-transitions*
- *G is the result of unfolding a finite graph and applying a regular substitution*

Prefix-recognizable graphs (2)

Properties

- Reachability relations and sets of reachable vertices are effectively computable
- The languages of prefix recognizable graphs are the context-free languages
- The monadic second order theory of any prefix-recognizable graph is decidable

Extensions and variants

- More general rewriting systems
  (term rewriting systems)
- More general computation models
  (higher-order pushdown automata)
- More powerful or iterated transformations
- More general finitely presented binary relations
  (automatic or rational relations)

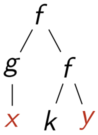
Extensions and variants

- More general rewriting systems
  (term rewriting systems)
- More general computation models
  (higher-order pushdown automata)
- More powerful or iterated transformations
- More general finitely presented binary relations
  (automatic or rational relations)

Term rewriting

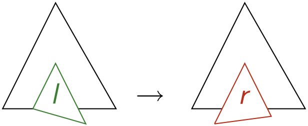
- Term: expression over function symbols and variables

$f(g(x), f(k, y))$



$a(b(c(x)))$

- Term rewriting system: set R of pairs of terms (l, r)
- *Ground* rewriting:



for all $(l, r) \in R$

Ground term rewriting graphs

Definition

A graph G is a (recognizable) ground rewriting graph if each of its set of edges is the ground rewriting relation of a (recognizable) ground term rewriting system R.

Example

The two-dimensional grid

Properties

- Reachability relations and sets of reachable vertices are effectively computable
- The first order theory with reachability of any ground rewriting graph is decidable
- The languages of ground rewriting graphs are ...?

Extensions and variants

- More general rewriting systems
  (term rewriting systems)
- More general computation models
  (higher-order pushdown automata)
- More powerful or iterated transformations
- More general finitely presented binary relations
  (automatic or rational relations)

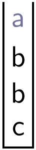
Higher-order pushdown stacks

Definition

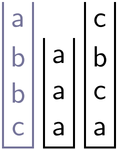
Higher-order stack:

- level 1: sequence of stack symbols (ordinary stack)
- level n: (non-empty) sequence of level $n-1$ stacks

Example



Level 1 store s
with a on top



Level 2 store
with s on top

Higher-order pushdown automata

Definition

Higher-order (level n) pushdown automaton:
pushdown automaton + higher order operations

- push_k: duplicate top-most level k stack
- pop_k: destroy top-most level k stack

Example

Automaton accepting the language $\{ww \mid w \in N^*\}$

Interesting abstract model for higher-order recursive sequential programs (e.g. ML, Scheme, ...)

Higher-order prefix-recognizable graphs

Theorem

Let G be a graph, the following statements are equivalent:

- *G is the transition graph of a level n pushdown automaton*
- *G is the result of applying (unfolding + substitution) n times to a finite graph*

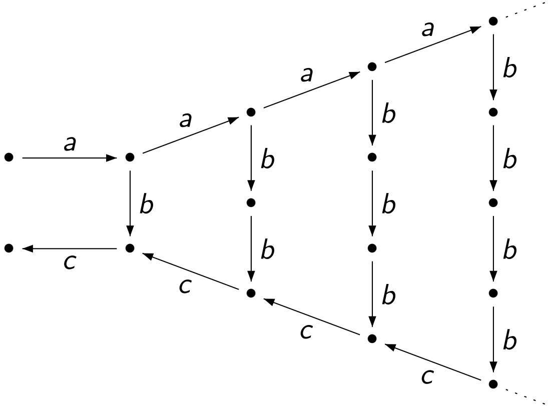
Properties

- Reachability relations and sets of reachable vertices are effectively computable
- The languages of these graphs are the IO languages
- The monadic second order theory of any such graph is decidable

Extensions and variants

- More general rewriting systems
  (term rewriting systems)
- More general computation models
  (higher-order pushdown automata)
- More powerful or iterated transformations
- More general finitely presented binary relations
  (automatic or rational relations)

A graph

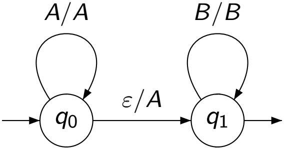


Rational relations

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

Properties

- Reachability relations and sets of reachable vertices are non-recursive
- The languages of rational graphs are the context-sensitive languages
- There exist some rational graphs whose first order theory is *un*decidable

A hierarchy of infinite automata

regular	finite
context-free	pushdown, prefix-recognizable . . .
context-sensitive	rational
recursively enumerable	computable

A hierarchy of infinite automata

regular	finite
context-free	pushdown, prefix-recognizable . . .
context-sensitive	rational
recursively enumerable	computable

Appendix:
languages of rational graphs (proof)

Subfamilies of rational graphs

- Synchronized transducer: all runs of the form

$$q_0 \xrightarrow{a_1/b_1} \dots \xrightarrow{a_n/b_n} \xrightarrow{\varepsilon/b_{n+1}} \dots \xrightarrow{\varepsilon/b_{n+k}} q_f$$

$$\text{or} \quad q_0 \xrightarrow{a_1/b_1} \dots \xrightarrow{a_n/b_n} \xrightarrow{a_{n+1}/\varepsilon} \dots \xrightarrow{a_{n+k}/\varepsilon} q_f$$

- Automatic graphs: rational graphs defined by synchronized transducers

- Synchronous transducer: no ε appearing on any transition
- Synchronous graphs: rational graphs defined by letter-to-letter transducers

Languages of rational graphs

- Existing proof uses the Penttonen normal form for context-sensitive grammars
  - Technically non-trivial
  - No link to complexity
  - No notion of determinism
- Our contributions:
  - New syntactical 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 Δ is a finite set of 2×2 pictures (tiles) with a border symbol $\#$

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

Proposition [Latteux&Simplot97]

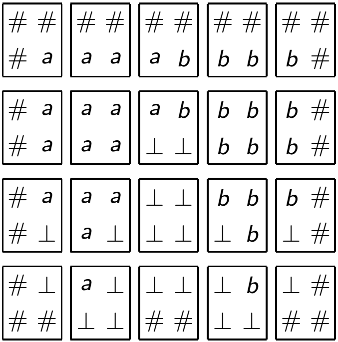
The languages of tiling systems are precisely the context-sensitive languages

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

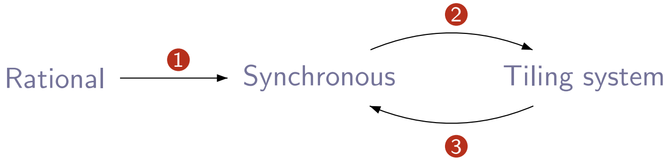


#	#	#	#	#	#	#	#	#	#	#	#
#	a	a	a	a	a	b	b	b	b	b	#
#	a	a	a	a	$\perp$	$\perp$	b	b	b	b	#
#	a	a	a	$\perp$	$\perp$	$\perp$	$\perp$	b	b	b	#
#	a	a	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	b	b	#
#	a	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	b	#
#	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	#
#	#	#	#	#	#	#	#	#	#	#	#

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 ε in transducers

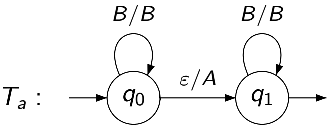
2 and 3 rely on identifying sequences of vertices with pictures

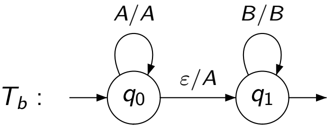
Rational $\longrightarrow$ synchronous graph

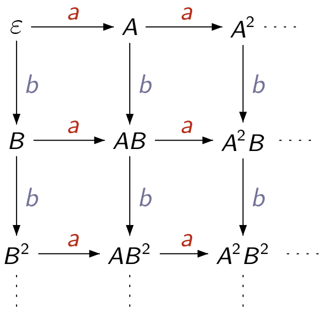
Proof idea

- Allow all transducers states to idle (ε/ε loops)
- Materialize ε as a fresh symbol $\#$ ($\longrightarrow$ synchronous graph)
- Define I' and F' as the shuffle of i and F by $\#^*$

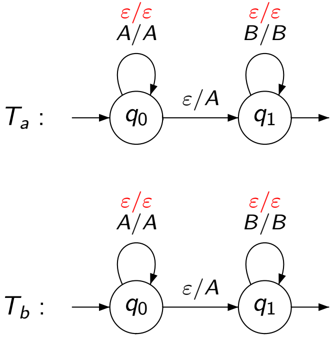
Rational $\rightarrow$ synchronous graph





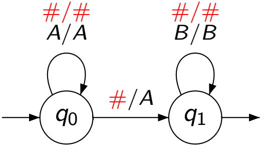


Rational → synchronous graph

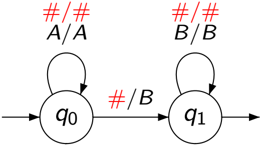


Rational → synchronous graph

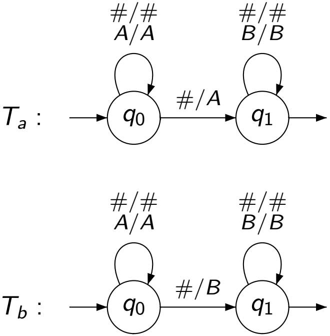
T_a :

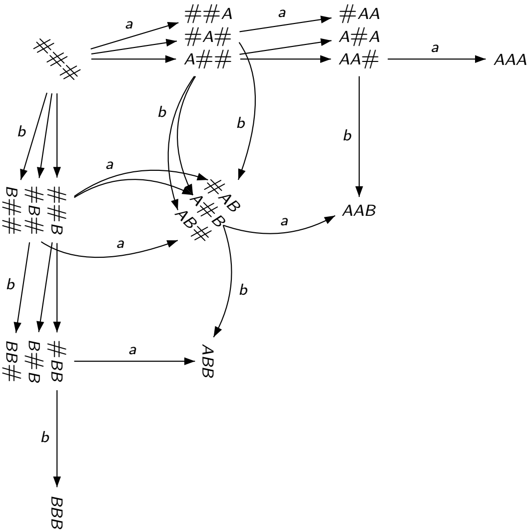


T_b :



Rational → synchronous graph





Synchronous graph $\leftrightarrow$ tiling system

Proof idea

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