Verification of infinite state systems

Angelo Montanari and Gabriele Puppis

Department of Mathematics and Computer Science
University of Udine, Italy
{montana,puppis}@dimi.uniud.it

Go

In this part

We present two interesting classes of transition systems:

- Context-free graphs
  these are (the connected components of) transition graphs of **pushdown systems**
- Prefix-recognizable graphs
  these are the transition graphs of **prefix rewriting systems**

We provide alternative representations of graphs in both classes and we show that their MSO-theories are decidable.

Definition (Pushdown system)

A **pushdown system** is a tuple $\mathcal{P} = (Q, \Gamma, A, \Delta)$, where:

- Q is a finite set of control states
- Γ is a finite set of stack symbols
- A is a finite set of transition labels
- $\Delta \subseteq Q \times \Gamma \times A \times Q \times \Gamma^*$ is a set of transition rules

Definition (Pushdown system)

A **pushdown system** is a tuple $\mathcal{P} = (Q, \Gamma, A, \Delta)$, where:

- Q is a finite set of control states
- Γ is a finite set of stack symbols
- A is a finite set of transition labels
- $\Delta \subseteq Q \times \Gamma \times A \times Q \times \Gamma^*$ is a set of transition rules

Configurations:
pairs in $Q \times \Gamma^*$ (state + stack content).

Transitions:
$(q, zw) \xrightarrow[\mathcal{P}]{a} (q', w'w)$ is a transition iff $(q, z, a, q', w') \in \Delta$.

Two main differences w.r.t. standard pushdown automata:

- no initial state and no initial stack symbol

  pushdown systems are not used as language acceptors ...
  ... we are interested in evaluating
  properties of their transition graphs

Two main differences w.r.t. standard pushdown automata:

- no initial state and no initial stack symbol

  pushdown systems are not used as language acceptors ...
  ... we are interested in evaluating
  properties of their transition graphs

- normalized forms of transition

  change: (q, z, a, q', z') with $q, q' \in Q$, $z, z' \in \Gamma$, $a \in A$
  push: $(q, z, a, q', z'z)$ with $q, q' \in Q$, $z, z' \in \Gamma$, $a \in A$
  pop: $(q, z, a, q', \varepsilon)$ with $q, q' \in Q$, $z \in \Gamma$, $a \in A$

  the length of the stack changes at most by one ...
  ... this is not a restriction: use *blocks of stack symbols*
  to put a generic pushdown system into normal form.

Definition (Pushdown transition graph)

The **transition graph** of a pushdown system $\mathcal{P} = (Q, \Gamma, A, \Delta)$ is the transition system $\mathcal{T} = \big(S, (\delta_a)_{a \in A}\big)$ where

- $S = Q \times \Gamma^*$
- $\big((q, w), (q', w')\big) \in \delta_a$ iff $(q, w) \xrightarrow[\mathcal{P}]{a} (q', w')$.

Note: pushdown graphs have bounded out-/in-degree.

Definition (Pushdown transition graph)

The **transition graph** of a pushdown system $\mathcal{P} = (Q, \Gamma, A, \Delta)$ is the transition system $\mathcal{T} = \big(S, (\delta_a)_{a \in A}\big)$ where

- $S = Q \times \Gamma^*$
- $\big((q, w), (q', w')\big) \in \delta_a$ iff $(q, w) \xrightarrow[\mathcal{P}]{a} (q', w')$.

Note: pushdown graphs have bounded out-/in-degree.

Definition (Connected component)

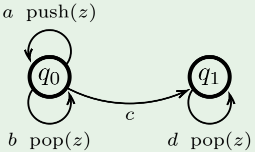
A (strongly) **connected component** of a graph $\mathcal{T}$ is a maximal subgraph of $\mathcal{T}$ such that, for every pair of vertices u, v, there exist a path π from u to v (π is allowed to traverse edges in both directions).

A **context-free graph** is a connected component of a pushdown transition graph.

Example

Consider the pushdown system $\mathcal{P} = (Q, \Gamma, A, \Delta)$, where

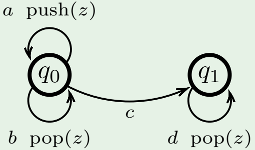
- $Q = \{q_0, q_1\}$
- $\Gamma = \{z\}$
- $A = \{a, b, c, d\}$
- Δ consists of the transitions (q_0, z, a, q_0, zz), $(q_0, z, b, q_0, \varepsilon)$, (q_0, z, c, q_1, z), $(q_1, z, d, q_1, \varepsilon)$



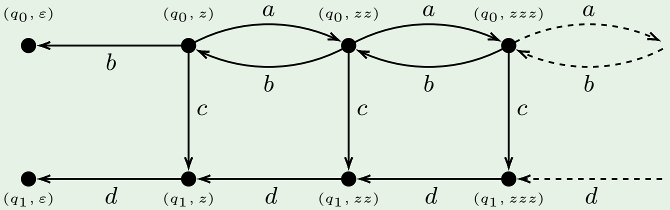
Example

Consider the pushdown system $\mathcal{P} = (Q, \Gamma, A, \Delta)$, where

- $Q = \{q_0, q_1\}$
- $\Gamma = \{z\}$
- $A = \{a, b, c, d\}$
- Δ consists of the transitions (q_0, z, a, q_0, zz), $(q_0, z, b, q_0, \varepsilon)$, (q_0, z, c, q_1, z), $(q_1, z, d, q_1, \varepsilon)$



The connected component of its transition graph is depicted below:



Theorem

Transition graphs of pushdown systems and context-free graphs can be defined inside the infinite binary tree via inverse rational mappings (in fact, inverse finite mappings) and rational restrictions.

Theorem

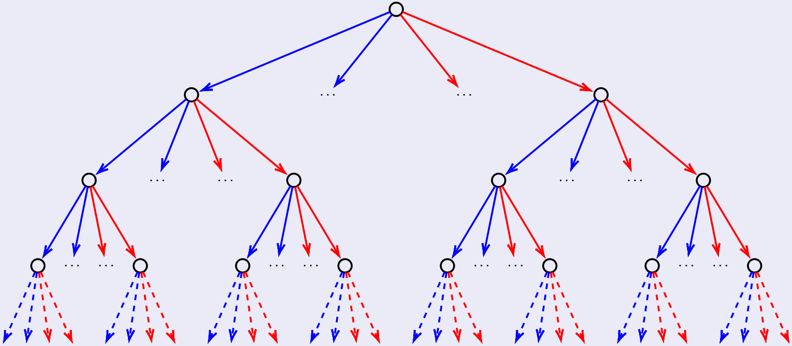
Transition graphs of pushdown systems and context-free graphs can be defined inside the infinite binary tree via inverse rational mappings (in fact, inverse finite mappings) and rational restrictions.

Corollary (Muller and Schupp '85)

The model checking problem for MSO logic over context-free graphs is decidable.

Proof of the theorem

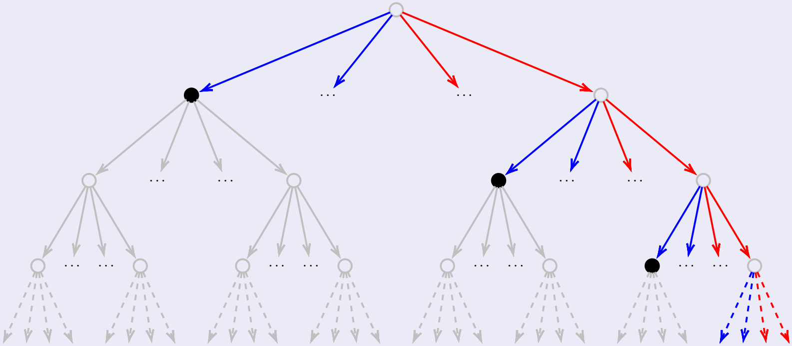
Let $\mathcal{P} = (Q, \Gamma, A, \Delta)$ be a pushdown system and let $\mathcal{T}$ be the **infinite** $Q \cup \Gamma$**-labeled tree**.



Proof of the theorem

We identify a $\mathcal{P}$-configuration (q, w) by the **reversed word** $\widetilde{w}q$ and we color the corresponding vertices of $\mathcal{T}$ by black:

$$k(black) := \Gamma^* Q$$

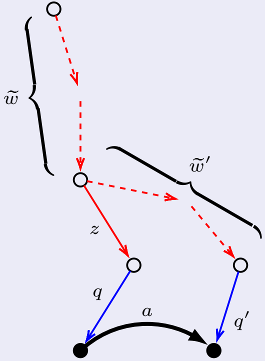


Proof of the theorem

We define a-labeled transitions via the inverse finite mapping

$$h(a) := \{\bar{q}\bar{z}\widetilde{w}'q' \ : \ (q, z, a, q', w') \in \Delta\}$$

$(q, zw) \xrightarrow[\mathcal{P}]{a} (q', w'w)$ iff $(\widetilde{w}zq, \widetilde{w}\widetilde{w}'q')$ is an a-labeled edge in $h^{-1}(\mathcal{T})$.



Proof of the theorem

Now, $h^{-1}(\mathcal{T})$, restricted to black-colored vertices, is the **transition graph** of $\mathcal{P}$.

To cope with context-free graphs, we must further restrict the domain of $h^{-1}(\mathcal{T})$.

Given a $\mathcal{P}$-configuration (q, w), we further restrict the domain of $h^{-1}(\mathcal{T})$ to the **regular** set

$$L := \widetilde{w}q \cdot \left(\bigcup_{a \in A} h(a) \cup \bigcup_{a \in A} \overline{\widetilde{h(a)}} \right)^*$$

thus obtaining the connected component of $h^{-1}(\mathcal{T})$ (i.e., the **context-free graph** of $\mathcal{P}$) that contains $\widetilde{w}q$.

Let thus now consider the class of prefix rewriting systems, which are a natural generalization of pushdown systems and, unlike them, may produce graphs with possibly infinite out-/in-degree.

Basic features:

- no more distinction between control states and stack letters (a single alphabet is used)
- less restricted forms of rewriting rules (more than one letter can be rewritten in a single transition)

Definition (Prefix rewriting system)

A **prefix rewriting system** is a tuple $\mathcal{P} = (\Gamma, L, A, \Delta)$, where:

- Γ is a finite set of symbols
- L is a regular language over Γ,
- A is a finite set of transition labels
- Δ is a finite set of rules of the form (U, a, V), where $a \in A$ and U, V *are regular languages over* Γ.

Definition (Prefix rewriting system)

A **prefix rewriting system** is a tuple $\mathcal{P} = (\Gamma, L, A, \Delta)$, where:

- Γ is a finite set of symbols
- L is a regular language over Γ,
- A is a finite set of transition labels
- Δ is a finite set of rules of the form (U, a, V), where $a \in A$ and U, V *are regular languages over* Γ.

Configurations:
all finite words in L.

Transitions:
$uw \xrightarrow[\mathcal{P}]{a} vw$ is a transition iff $\exists\, (U, a, V) \in \Delta.\ u \in U, v \in V$.

Definition (Prefix rewriting system)

A **prefix rewriting system** is a tuple $\mathcal{P} = (\Gamma, L, A, \Delta)$, where:

- Γ is a finite set of symbols
- L is a regular language over Γ,
- A is a finite set of transition labels
- Δ is a finite set of rules of the form (U, a, V), where $a \in A$ and U, V *are regular languages over* Γ.

Configurations:
all finite words in L.

Transitions:
$uw \xrightarrow[\mathcal{P}]{a} vw$ is a transition iff $\exists\,(U, a, V) \in \Delta.\ u \in U, v \in V$.

Note: pushdown systems are special forms of prefix rewriting ones.

Definition (Prefix-recognizable graph)

A **prefix-recognizable graph** is the transition graph of a prefix rewriting system.

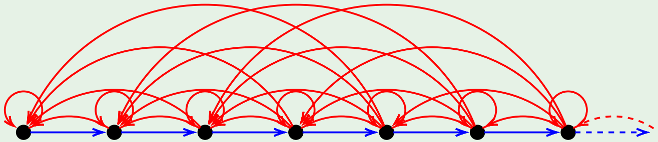
Definition (Prefix-recognizable graph)

A **prefix-recognizable graph** is the transition graph of a prefix rewriting system.

Example

Consider the prefix rewriting system $\mathcal{P} = (\Gamma, L, A, \Delta)$, where

- $\Gamma = \{z\}$
- $L = \{z\}^*$
- $A = \{succ, geq\}$
- Δ consists of the two rules $\big(\{\varepsilon\}, succ, \{z\}\big)$, $\big(\{z\}^*, geq, \{\varepsilon\}\big)$.



Theorem (Caucal '96)

Prefix-recognizable graphs are definable inside the infinite binary tree via inverse rational mappings and rational restrictions.

Theorem (Caucal '96)

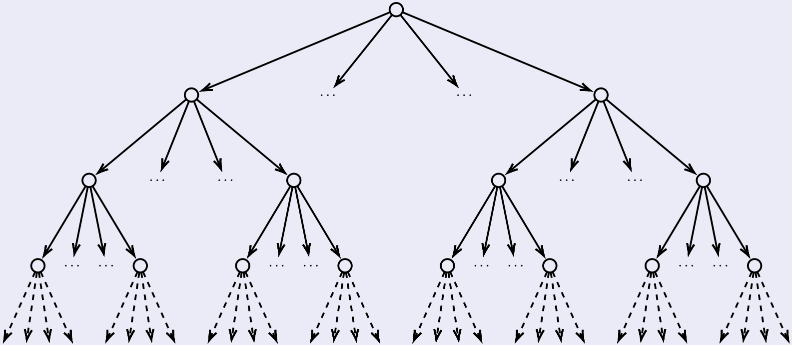
Prefix-recognizable graphs are definable inside the infinite binary tree via inverse rational mappings and rational restrictions.

Corollary (Caucal '96)

The model checking problem for MSO logic over prefix-recognizable graphs is decidable.

Proof of the theorem

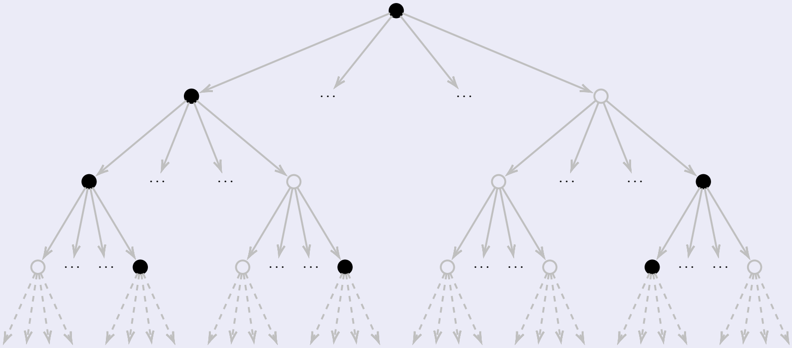
Let $\mathcal{P} = (\Gamma, L, A, \Delta)$ be a prefix rewriting system and let $\mathcal{T}$ be the **infinite Γ-labeled tree**.



Proof of the theorem

We identify a $\mathcal{P}$-configuration $w \in L$ by the **reversed word** $\widetilde{w} \in \widetilde{L}$. and we color the corresponding vertices of $\mathcal{T}$ by black:

$$k(black) := \widetilde{L}$$

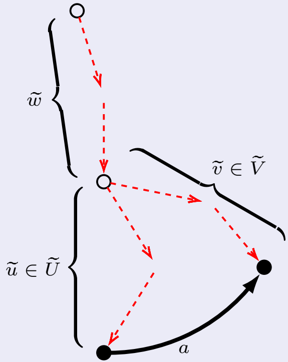


Proof of the theorem

We define a-labeled transitions via the inverse rational mapping

$$h(a) := \bigcup_{(U,a,V)\in\Delta} \overline{U} \cdot \widetilde{V}$$

$uw \xrightarrow[\mathcal{P}]{a} vw$ iff $(\widetilde{w}\widetilde{u}, \widetilde{w}\widetilde{v})$ is an a-labeled edge in $h^{-1}(\mathcal{T})$.



So far we know that

- pushdown transition graphs are obtained from the infinite binary tree via inverse finite mappings and rational restrictions.

  The converse is also true (Caucal '96):
  inverse finite mappings and rational restrictions applied to the infinite binary tree yield pushdown transition graphs.

So far we know that

- pushdown transition graphs are obtained from the infinite binary tree via inverse finite mappings and rational restrictions.

  The converse is also true (Caucal '96):
  inverse finite mappings and rational restrictions applied to the infinite binary tree yield pushdown transition graphs.

- prefix recognizable graphs are obtained from the infinite binary tree via inverse rational mappings and rational restrictions.

  The converse is also true (Caucal '96):
  inverse rational mappings and rational restrictions applied to the infinite binary tree yield prefix recognizable graphs.

So far we know that

- pushdown transition graphs are obtained from the infinite binary tree via inverse finite mappings and rational restrictions.

  The converse is also true (Caucal '96):
  inverse finite mappings and rational restrictions applied to the infinite binary tree yield pushdown transition graphs.

- prefix recognizable graphs are obtained from the infinite binary tree via inverse rational mappings and rational restrictions.

  The converse is also true (Caucal '96):
  inverse rational mappings and rational restrictions applied to the infinite binary tree yield prefix recognizable graphs.

$\Rightarrow$ inverse finite/rational mappings and rational restrictions can be thought of as **external presentations** of pushdown/prefix-recognizable graphs.

Context-free graphs have alternative representations based on **hyperedge-replacement graph grammars**.

Context-free graphs have alternative representations based on **hyperedge-replacement graph grammars**.

Definition (Hypergraph)

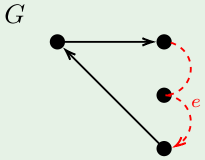
A **hyperedge** is a sequence of k vertices $(v_1, ..., v_k)$.
(an edge is a special form of hyperedge with 2 vertices only).

A **hypergraph** is a graph where edges are replaced with hyperedges (labels can be assigned to the hyperedges of a hypergraph).

A **hyperedge replacement** is the replacement of a hyperedge $e = (v_1, ..., v_k)$ in a hypergraph G with a (hyper)graph H

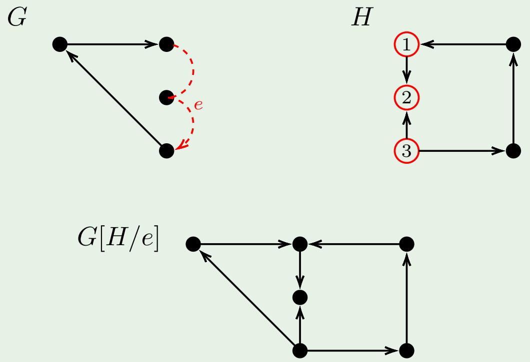
(a suitable marking of the vertices of H is used to identify the vertices of H that replace the vertices of G in e).

Example (Hyperedge replacement)



Example (Hyperedge replacement)

Example (Hyperedge replacement)



Definition (Graph grammar)

Given a finite set N of **nonterminal symbols**,
a **(hyperedge-replacement) graph grammar** is
a tuple $\mathcal{G} = (H_z)_{z \in N}$ of hypergraphs that defines
the grammar rules for the replacement of every z-labeled
hyperedge with the hypergraph H_z.

Definition (Graph grammar)

Given a finite set N of **nonterminal symbols**,
a **(hyperedge-replacement) graph grammar** is
a tuple $\mathcal{G} = (H_z)_{z \in N}$ of hypergraphs that defines
the grammar rules for the replacement of every z-labeled
hyperedge with the hypergraph H_z.

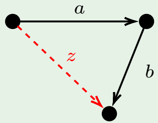
The **pattern graph** generated by $\mathcal{G}$ starting from
an axiom $z_0 \in N$ is the *limit* of the sequence of graphs
obtained by the repeated application of replacement rules in $\mathcal{G}$.

Note that

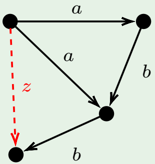
- *the limit operation does not take into account the nonterminal hyperedges*
- *every hyperedge is eventually replaced* (replacement order does not matter).

Example (a pattern graph)

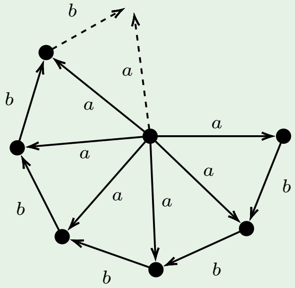
Example (a pattern graph)



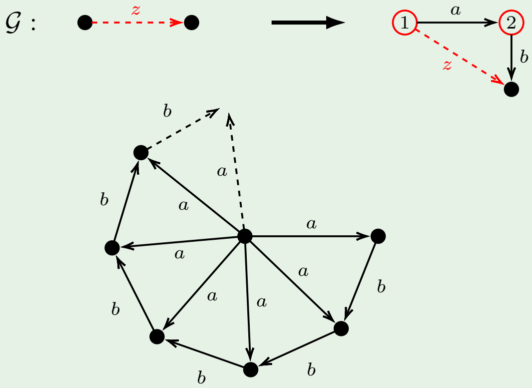
Example (a pattern graph)



Example (a pattern graph)



Example (a pattern graph)



Note: pattern graphs may have infinite in-/out- degree and unconnected components ...

If we restrict ourselves to **special** graph grammars $\mathcal{G} = (H_z)_{z \in N}$, where

- nonterminal hyperedges in H_z have no repeated vertices
- vertices of nonterminal hyperedges in H_z are not marked
- every vertex of a nonterminal hyperedge in H_z is also a vertex of a terminal edge
- every terminal edge in H_z has at least one marked vertex
- for every (non-terminal symbol) $z \in N$, the pattern graph generated by $\mathcal{G}$ starting from z is a connected graph,

If we restrict ourselves to **special** graph grammars $\mathcal{G} = (H_z)_{z \in N}$, where

- nonterminal hyperedges in H_z have no repeated vertices
- vertices of nonterminal hyperedges in H_z are not marked
- every vertex of a nonterminal hyperedge in H_z is also a vertex of a terminal edge
- every terminal edge in H_z has at least one marked vertex
- for every (non-terminal symbol) $z \in N$, the pattern graph generated by $\mathcal{G}$ starting from z is a connected graph,

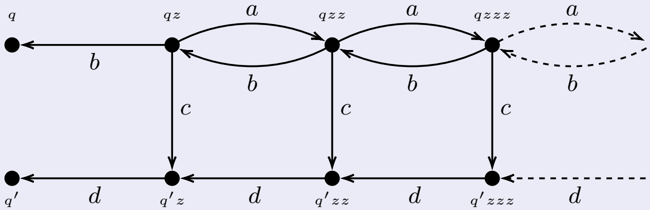
then:

Theorem (Muller and Schupp '85)

The context-free graphs are exactly the pattern graphs generated by special (hyperedge-replacement) graph grammars.

An intuitive account (one direction)

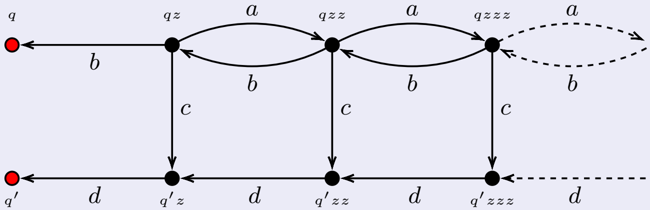
Consider the context-free graph $\mathcal{T} = (S, (\delta_a)_{a \in A})$



An intuitive account (one direction)

The **end-components** induced by $V_n = \{v \in S : |v| \geq n\}$ are

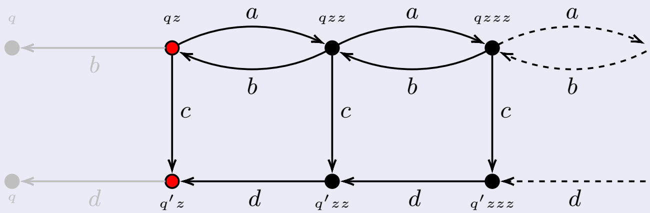
$n = 1$



An intuitive account (one direction)

The **end-components** induced by $V_n = \{v \in S : |v| \geq n\}$ are

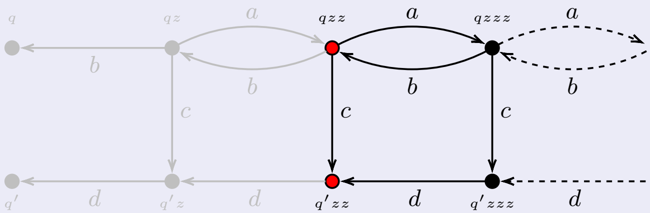
$n = 2$



An intuitive account (one direction)

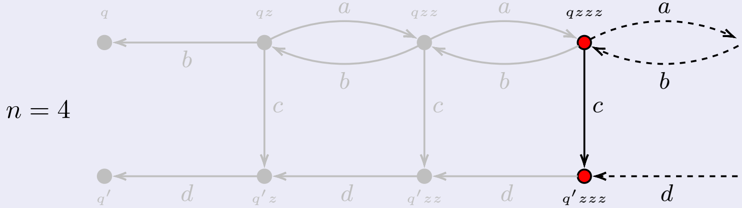
The **end-components** induced by $V_n = \{v \in S \, : \, |v| \geq n\}$ are

$n = 3$



An intuitive account (one direction)

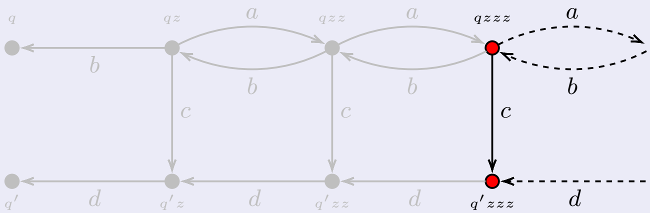
The **end-components** induced by $V_n = \{v \in S \,:\, |v| \geq n\}$ are



An intuitive account (one direction)

The **end-components** induced by $V_n = \{v \in S \,:\, |v| \geq n\}$ are

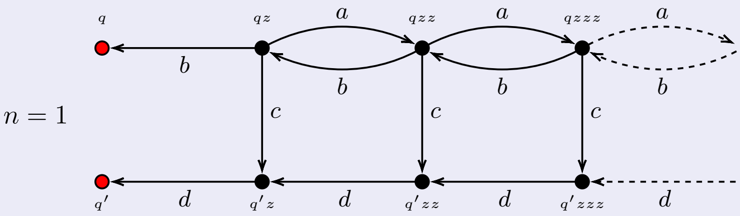
$n = 4$



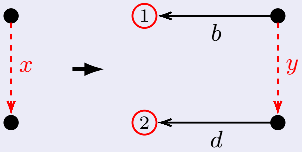
One can show that there are only finitely many non-isomorphic end-components, each one generating a distinct replacement rule

An intuitive account (one direction)

The **end-components** induced by $V_n = \{v \in S \,:\, |v| \geq n\}$ are



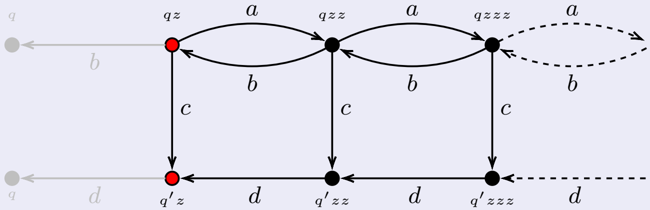
One can show that there are only finitely many non-isomorphic end-components, each one generating a distinct replacement rule



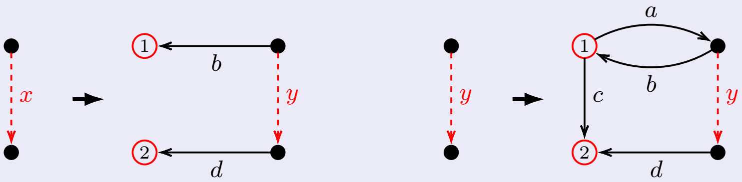
An intuitive account (one direction)

The **end-components** induced by $V_n = \{v \in S : |v| \geq n\}$ are

$n = 2$



One can show that there are only finitely many non-isomorphic end-components, each one generating a distinct replacement rule



An analogous characterization holds for prefix-recognizable graphs:

Theorem (Courcelle '92)

The prefix-recognizable graphs are exactly the pattern graphs generated by **vertex-replacement** *graph grammars.*

An analogous characterization holds for prefix-recognizable graphs:

Theorem (Courcelle '92)

The prefix-recognizable graphs are exactly the pattern graphs generated by **vertex-replacement** *graph grammars.*

Moreover, we have that

Theorem

The prefix-recognizable graphs are exactly the graphs in the second level of the Caucal hierarchy, namely, the graphs generated by MSO-interpretations over infinite regular trees.

Go