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

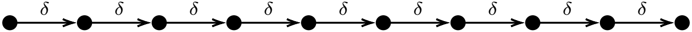
Go

In this part

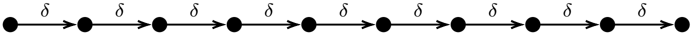
We present basic results and techniques related to the decidability of *MSO-theories* of infinite transition systems.

- Automaton-based approaches
  - decidability of the MSO-theory of a finite (discrete) line
  - **Büchi's Theorem**
    (decidability of the MSO-theory of the semi-infinite line)
  - **Rabin's Theorem**
    (decidability of the MSO-theory of the infinite binary tree)
- Transformational approaches
  - **interpretations**, **inverse mappings**, **markings**
  - **unfoldings**
  - **Caucal hierarchy**

Consider a **finite line** $\mathcal{L}_n = \big(\{1, ..., n\}, \delta\big)$



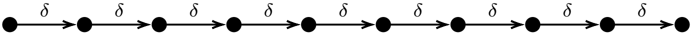
Consider a **finite line** $\mathcal{L}_n = \big(\{1, ..., n\}, \delta\big)$



We expand $\mathcal{L}_n$ by **unary predicates** $P_1, ..., P_m \subseteq \{1, ..., n\}$
and we obtain a colored line $\mathcal{L}_{n,\bar{P}} = \big(\{1, ..., n\}, \delta, P_1, ..., P_m\big)$

The automaton-based approach – The finite line

Consider a **finite line** $\mathcal{L}_n = (\{1, \ldots, n\}, \delta)$



We expand $\mathcal{L}_n$ by **unary predicates** $P_1, \ldots, P_m \subseteq \{1, \ldots, n\}$ and we obtain a colored line $\mathcal{L}_{n,\bar{P}} = (\{1, \ldots, n\}, \delta, P_1, \ldots, P_m)$

We encode $\mathcal{L}_{n,\bar{P}}$ by a **finite word** $w_{n,\bar{P}}$ over $\mathbb{B}^m = \{0, 1\}^m$ such that

$$|w_{n,\bar{P}}| = n$$

$$w_{n,\bar{P}}[i] = (b_1, \ldots, b_m) \qquad \text{where } b_j = \begin{cases} 1 & \text{if } i \in P_j \\ 0 & \text{if } i \notin P_j \end{cases}$$

Example

The colored line $\mathcal{L}_{10,\bar{P}} = (\{1, \ldots, 10\}, \delta, P_{even}, P_{prime})$

is encoded by $w_{10,\bar{P}} = \binom{0}{0}\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{0}{1}\binom{1}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}\binom{1}{0}$.

W.l.o.g. we can think of each *FO-variable* x in an MSO-formula ψ as an *MSO-variable* X *which stands for the singleton* $\{x\}$.

$\Rightarrow$ we can get rid of FO-variables
(by taking $\delta(X_i, X_j)$, with X_i, X_j singletons, and $X_i \subseteq X_j$ as atomic formulas)

W.l.o.g. we can think of each *FO-variable* x in an MSO-formula ψ as an *MSO-variable* X *which stands for the singleton* $\{x\}$.

$\Rightarrow$ we can get rid of FO-variables
(by taking $\delta(X_i, X_j)$, with X_i, X_j singletons, and $X_i \subseteq X_j$ as atomic formulas)

We evaluate an MSO-formula ψ with free variables $X_1, ..., X_m$ over the finite line expanded by any tuple of unary predicates $P_1, ..., P_m$:

$$\mathcal{L}_{n,\bar{P}} \overset{?}{\models} \psi[P_1/X_1, ..., P_m/X_m]$$

The above problem can be reduced to the (decidable) acceptance problem for finite state automata.

Theorem (Acceptance problem)

For any MSO-formula ψ with free variables $X_1, \ldots, X_m$, one can compute a **finite state automaton** $\mathcal{A}_\psi$ *over* $\mathbb{B}^m$ *such that, for every n-length colored line* $\mathcal{L}_{n,\bar{P}}$

$$\mathcal{L}_{n,\bar{P}} \models \psi[P_1/X_1, \ldots, P_m/X_m] \quad \textit{iff} \quad w_{n,\bar{P}} \in \mathscr{L}(\mathcal{A}_\psi)$$

Intuitively: the words accepted by the automaton $\mathcal{A}_\psi$ are all and only the encodings of the linear models of the formula ψ.

Theorem (Acceptance problem)

For any MSO-formula ψ with free variables $X_1, \ldots, X_m$, one can compute a **finite state automaton** $\mathcal{A}_\psi$ *over* $\mathbb{B}^m$ *such that, for every n-length colored line* $\mathcal{L}_{n,\bar{P}}$

$$\mathcal{L}_{n,\bar{P}} \models \psi[P_1/X_1, \ldots, P_m/X_m] \quad \textit{iff} \quad w_{n,\bar{P}} \in \mathscr{L}(\mathcal{A}_\psi)$$

Intuitively: the words accepted by the automaton $\mathcal{A}_\psi$ are all and only the encodings of the linear models of the formula ψ.

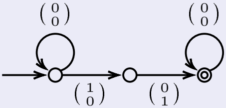
Note: if ψ contains no free variables ($m = 0$), then $\mathcal{A}_\psi$ is an **input-free** automaton.

Corollary

The MSO-theory of a finite line $\mathcal{L}_n = (\{1, \ldots, n\}, \delta)$ *is reducible to the (decidable)* **acceptance problem** *for finite state automata.*

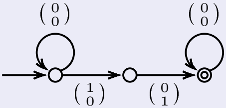
Proof of the theorem

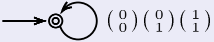
By *induction on the structure* of the formula ψ:

- if $\psi = \delta(X_1, X_2)$, then $\mathcal{A}_\psi :=$ 

Proof of the theorem

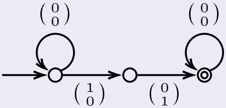
By *induction on the structure* of the formula ψ:

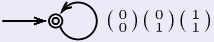
- if $\psi = \delta(X_1, X_2)$, then $\mathcal{A}_\psi :=$ 

- if $\psi = X_1 \subseteq X_2$, then $\mathcal{A}_\psi :=$ 

Proof of the theorem

By *induction on the structure* of the formula ψ:

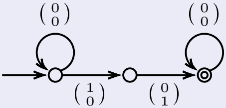
- if $\psi = \delta(X_1, X_2)$, then $\mathcal{A}_\psi :=$ 

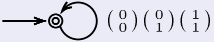
- if $\psi = X_1 \subseteq X_2$, then $\mathcal{A}_\psi :=$ 

- if $\psi = \varphi_1 \wedge \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cap \mathcal{A}_{\varphi_2}$

Proof of the theorem

By *induction on the structure* of the formula ψ:

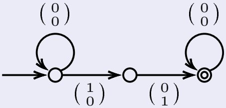
- if $\psi = \delta(X_1, X_2)$, then $\mathcal{A}_\psi :=$ 

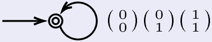
- if $\psi = X_1 \subseteq X_2$, then $\mathcal{A}_\psi :=$ 

- if $\psi = \varphi_1 \wedge \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cap \mathcal{A}_{\varphi_2}$
- if $\psi = \varphi_1 \vee \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cup \mathcal{A}_{\varphi_2}$

Proof of the theorem

By *induction on the structure* of the formula ψ:

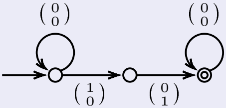
- if $\psi = \delta(X_1, X_2)$, then $\mathcal{A}_\psi :=$ 

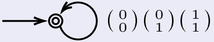
- if $\psi = X_1 \subseteq X_2$, then $\mathcal{A}_\psi :=$ 

- if $\psi = \varphi_1 \ \wedge \ \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cap \mathcal{A}_{\varphi_2}$
- if $\psi = \varphi_1 \ \vee \ \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cup \mathcal{A}_{\varphi_2}$
- if $\psi = \neg\varphi$, then $\mathcal{A}_\psi$ is the complement automaton of $\mathcal{A}_\varphi$

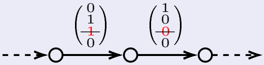
Proof of the theorem

By *induction on the structure* of the formula ψ:

- if $\psi = \delta(X_1, X_2)$, then $\mathcal{A}_\psi :=$ 

- if $\psi = X_1 \subseteq X_2$, then $\mathcal{A}_\psi :=$ 

- if $\psi = \varphi_1 \wedge \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cap \mathcal{A}_{\varphi_2}$
- if $\psi = \varphi_1 \vee \varphi_2$, then $\mathcal{A}_\psi := \mathcal{A}_{\varphi_1} \cup \mathcal{A}_{\varphi_2}$
- if $\psi = \neg\varphi$, then $\mathcal{A}_\psi$ is the complement automaton of $\mathcal{A}_\varphi$
- if $\psi = \exists\, X_i.\ \varphi(X_1, \ldots, X_i, \ldots, X_m)$, then $\mathcal{A}_\psi$ is obtained from $\mathcal{A}_\varphi$ by removing the i-th component of each input symbol



We exploited closure properties of finite state automata
w.r.t. union, intersection, complementation, and projection

to reduce the problem of deciding the MSO-theory of a finite line
to a (decidable) acceptance problem over finite state automata.

We exploited closure properties of finite state automata w.r.t. union, intersection, complementation, and projection

to reduce the problem of deciding the MSO-theory of a finite line to a (decidable) acceptance problem over finite state automata.

What about the **semi-infinite line** $\mathcal{L}_\omega = (\mathbb{N}, \delta)$?

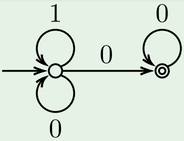
Basic ingredients

- We need to use **infinite words**, rather than finite ones, to encode expansions of $\mathcal{L}_\omega$ by unary predicates
- We need to introduce a suitable class of automata working on infinite words: **Büchi automata**!

Definition (**Büchi automaton**)

It is a non-deterministic finite state automaton that accepts an infinite word w iff there is a run ρ on w such that $\mathcal{I}nf(\rho) \cap F \neq \emptyset$ (‘*ρ contains at least one final state that occurs infinitely often*’).

Example

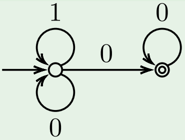


is a Büchi automaton recognizing the language $\{0,1\}^* \cdot \{0\}^\omega$ (note: non-determinism is needed)

Definition (Büchi automaton)

It is a non-deterministic finite state automaton that accepts an infinite word w iff there is a run ρ on w such that $\mathcal{I}nf(\rho) \cap F \neq \emptyset$ (‘*ρ contains at least one final state that occurs infinitely often*’).

Example



is a Büchi automaton recognizing the language $\{0,1\}^* \cdot \{0\}^\omega$ (note: non-determinism is needed)

Lemma (Büchi ’62)

Büchi automata are effectively closed under union, intersection, complementation, and projection.

Theorem (Büchi '62)

For any MSO-formula ψ *with free variables* $X_1, \ldots, X_m$, *one can compute a Büchi automaton* $\mathcal{A}_\psi$ *over* $\mathbb{B}^m$ *such that, for every tuple of unary predicates* $P_1, \ldots, P_m \subseteq \mathbb{N}$

$$\mathcal{L}_{\omega,\bar{P}} \models \psi[P_1/X_1, \ldots, P_m/X_m] \quad \textit{iff} \quad w_{\omega,\bar{P}} \in \mathscr{L}(\mathcal{A}_\psi)$$

Theorem (Büchi '62)

For any MSO-formula ψ with free variables $X_1, \ldots, X_m$, one can compute a Büchi automaton $\mathcal{A}_\psi$ over $\mathbb{B}^m$ such that, for every tuple of unary predicates $P_1, \ldots, P_m \subseteq \mathbb{N}$

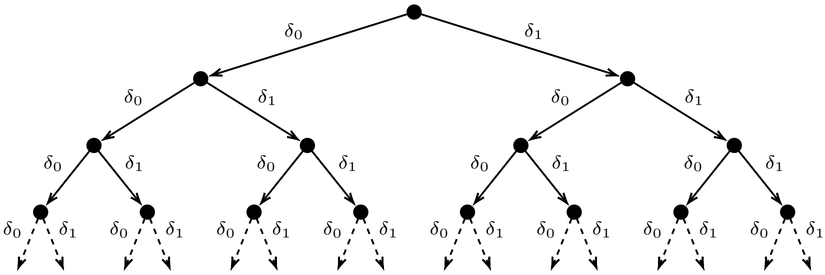
$$\mathcal{L}_{\omega,\bar{P}} \vDash \psi[P_1/X_1, \ldots, P_m/X_m] \quad \textit{iff} \quad w_{\omega,\bar{P}} \in \mathscr{L}(\mathcal{A}_\psi)$$

Note: if ψ contains no free variables ($m = 0$)
then $\mathcal{A}_\psi$ is input free and it recognizes
either the empty language or a singleton.

Corollary

The MSO-theory of the semi-infinite line $\mathcal{L} = (\mathbb{N}, \delta)$ is reducible to the (decidable) **emptiness problem** *for Büchi automata.*

What about MSO-theories of branching structures, in particular, of the **infinite binary tree** $\mathcal{T}_2 = (\mathbb{B}^*, \delta_0, \delta_1)$?

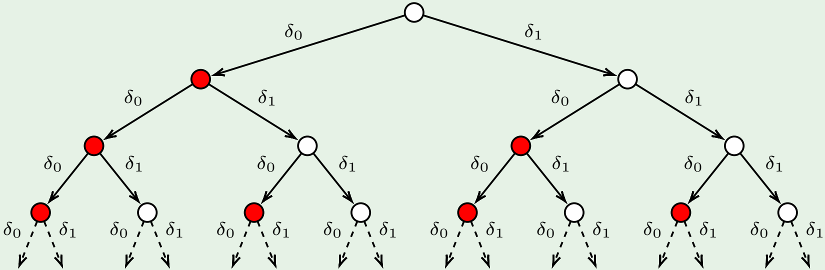


In analogy to the previous cases, we shall describe an automaton-based method to decide the MSO-theory of $\mathcal{T}_2$.

Now, expansions of the infinite binary tree $\mathcal{T}_2$ with unary predicates $P_1, \ldots, P_m \subseteq \mathbb{B}^*$ are encoded by $\mathbb{B}^m$-colored trees.

Example

The expanded tree $\left(\mathbb{B}^*, \delta_0, \delta_1, P\right)$, where $P = \{\textit{left successors}\}$, is encoded by the colored tree $\mathcal{T}_{2,P}$



We need a suitable class of automata running on *colored trees*, rather than words: **Rabin tree automata**!

Definition (Rabin tree automaton)

A **Rabin tree automaton** is a tuple $\mathcal{A} = \big(Q, C, \Delta, q_0, \{(G_1, F_1), \ldots, (G_k, F_k)\}\big)$, where:

- Q is a finite set of states
- C is a finite set of vertex colors (e.g., $\mathbb{B}^m$)
- $\Delta \subseteq Q \times C \times Q \times Q$ is a transition relation
- $q_0 \in Q$ is the initial state
- for all $1 \leq i \leq k$, (G_i, F_i) is an **accepting pair**, with $G_i, F_i \subseteq Q$.

How does a Rabin tree automaton $\mathcal{A}$ accept a colored tree?

Definition (Successful run)

A **successful run** of $\mathcal{A}$ on an infinite binary C-colored tree $\mathcal{T}$ is an infinite binary Q-colored tree $\mathcal{R}$ such that:

- $\mathcal{R}(\varepsilon) = q_0$
  '*the state at the root is the initial state of* $\mathcal{A}$'

How does a Rabin tree automaton $\mathcal{A}$ accept a colored tree?

Definition (Successful run)

A **successful run** of $\mathcal{A}$ on an infinite binary C-colored tree $\mathcal{T}$ is an infinite binary Q-colored tree $\mathcal{R}$ such that:

- $\mathcal{R}(\varepsilon) = q_0$
  '*the state at the root is the initial state of* $\mathcal{A}$'
- for every vertex v, $\big(\mathcal{R}(v), \mathcal{T}(v), \mathcal{R}(v \cdot 0), \mathcal{R}(v \cdot 1)\big) \in \Delta$
  '*if* $\mathcal{A}$ *lies at* v *with color* $c = \mathcal{T}(v)$ *and state* $q = \mathcal{R}(v)$, $\mathcal{A}$ *can associate the states* $q' = \mathcal{R}(v \cdot 0), q'' = \mathcal{R}(v \cdot 1)$ *with the two successors of* v *iff* (q, c, q', q'') *is a valid transition*'

How does a Rabin tree automaton $\mathcal{A}$ accept a colored tree?

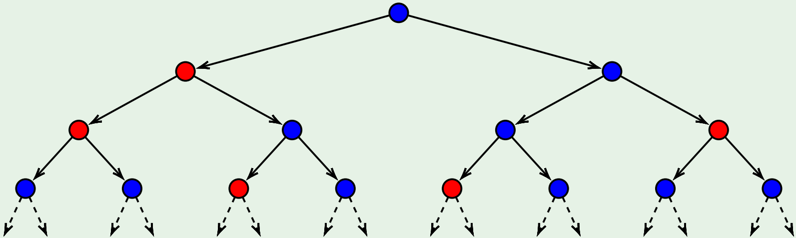
Definition (Successful run)

A **successful run** of $\mathcal{A}$ on an infinite binary C-colored tree $\mathcal{T}$ is an infinite binary Q-colored tree $\mathcal{R}$ such that:

- $\mathcal{R}(\varepsilon) = q_0$
  '*the state at the root is the initial state of* $\mathcal{A}$'
- for every vertex v, $\big(\mathcal{R}(v), \mathcal{T}(v), \mathcal{R}(v \cdot 0), \mathcal{R}(v \cdot 1)\big) \in \Delta$
  '*if* $\mathcal{A}$ *lies at* v *with color* $c = \mathcal{T}(v)$ *and state* $q = \mathcal{R}(v)$, $\mathcal{A}$ *can associate the states* $q' = \mathcal{R}(v \cdot 0), q'' = \mathcal{R}(v \cdot 1)$ *with the two successors of* v *iff* (q, c, q', q'') *is a valid transition*'
- for every infinite path π, there is $1 \leq i \leq k$ such that $\mathcal{I}nf(\mathcal{R}|\pi) \cap G_i \neq \emptyset$ and $\mathcal{I}nf(\mathcal{R}|\pi) \cap F_i = \emptyset$
  '*at least one state of* G_i *occurs infinitely often in* $\mathcal{R}$ *along* π' and '*all states of* F_i *occur only finitely often in* $\mathcal{R}$ *along* π'

Example

Consider the {*red*, *blue*}-colored tree

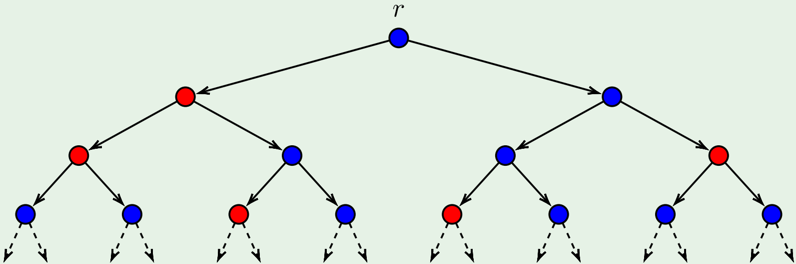


and the Rabin tree automaton having

- two states, r and b, that keep track of which color was seen last
- transitions (r, red, r, r), (b, red, r, r),
  $(r, blue, b, b)$, $(b, blue, b, b)$
- a single accepting pair (G_1, F_1), with $G_1 = \{b\}$, $F_1 = \{r\}$

Example

Consider the {*red*, *blue*}-colored tree

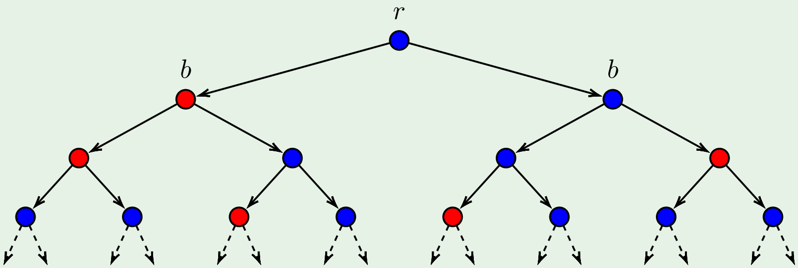


and the Rabin tree automaton having

- two states, r and b, that keep track of which color was seen last
- transitions $(r, \textit{red}, r, r)$, $(b, \textit{red}, r, r)$,
  $(r, \textit{blue}, b, b)$, $(b, \textit{blue}, b, b)$
- a single accepting pair (G_1, F_1), with $G_1 = \{b\}$, $F_1 = \{r\}$

Example

Consider the {*red*, *blue*}-colored tree

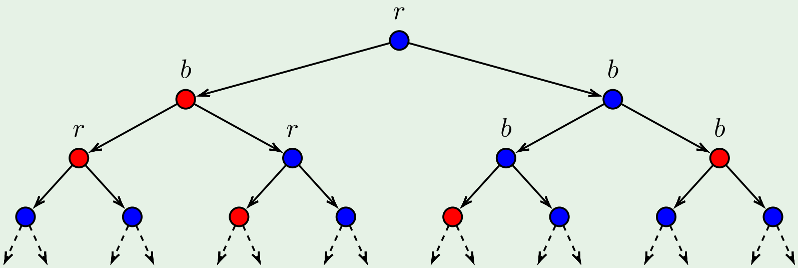


and the Rabin tree automaton having

- two states, r and b, that keep track of which color was seen last
- transitions $(r, \textit{red}, r, r)$, $(b, \textit{red}, r, r)$,
  $(r, \textit{blue}, b, b)$, $(b, \textit{blue}, b, b)$
- a single accepting pair (G_1, F_1), with $G_1 = \{b\}$, $F_1 = \{r\}$

Example

Consider the {*red*, *blue*}-colored tree

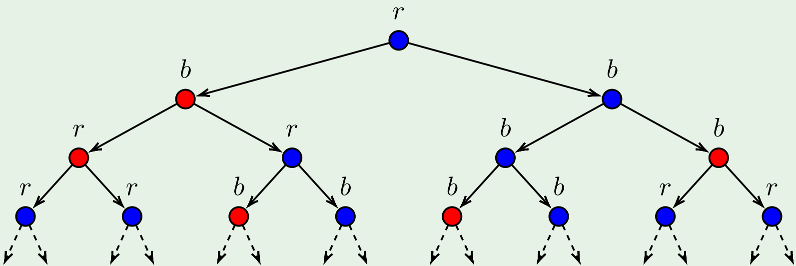


and the Rabin tree automaton having

- two states, r and b, that keep track of which color was seen last
- transitions $(r, \textit{red}, r, r)$, $(b, \textit{red}, r, r)$,
  $(r, \textit{blue}, b, b)$, $(b, \textit{blue}, b, b)$
- a single accepting pair (G_1, F_1), with $G_1 = \{b\}$, $F_1 = \{r\}$

Example

Consider the {*red*, *blue*}-colored tree

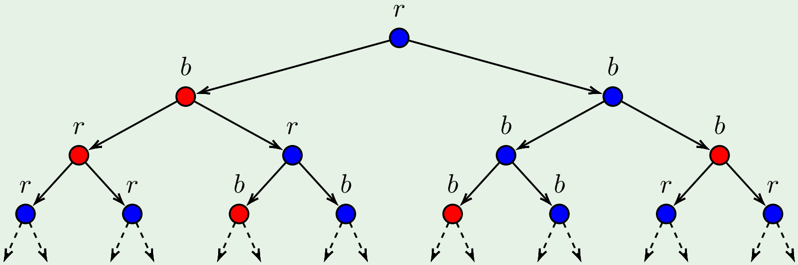


and the Rabin tree automaton having

- two states, r and b, that keep track of which color was seen last
- transitions $(r, \textit{red}, r, r)$, $(b, \textit{red}, r, r)$,
  $(r, \textit{blue}, b, b)$, $(b, \textit{blue}, b, b)$
- a single accepting pair (G_1, F_1), with $G_1 = \{b\}$, $F_1 = \{r\}$

Example

Consider the {*red*, *blue*}-colored tree



and the Rabin tree automaton having

- two states, r and b, that keep track of which color was seen last
- transitions $(r, \textit{red}, r, r)$, $(b, \textit{red}, r, r)$,
  $(r, \textit{blue}, b, b)$, $(b, \textit{blue}, b, b)$
- a single accepting pair (G_1, F_1), with $G_1 = \{b\}$, $F_1 = \{r\}$

$\Rightarrow$ $\mathcal{A}$ accepts those trees whose paths encompass *only finitely many red-colored vertices*

Lemma (Rabin '69)

Rabin tree automata are effectively closed under union, intersection, complementation, and projection.

Lemma (Rabin '69)

Rabin tree automata are effectively closed under union, intersection, complementation, and projection.

Theorem (Rabin '69)

For any MSO-formula ψ with free variables $X_1, \ldots, X_m$, one can compute a Rabin tree automaton $\mathcal{A}_\psi$ over $\mathbb{B}^m$ such that, for every tuple of unary predicates $P_1, \ldots, P_m \subseteq \mathbb{B}^*$

$$(\mathbb{B}^*, \delta_0, \delta_1, \bar{P}) \vDash \psi[P_1/X_1, \ldots, P_m/X_m] \quad \textit{iff} \quad \mathcal{T}_{2,\bar{P}} \in \mathscr{L}(\mathcal{A}_\psi)$$

Corollary

The MSO-theory of the infinite binary tree $(\mathbb{B}^*, \delta_0, \delta_1)$ is reducible to the (decidable) **emptiness problem** *for Rabin tree automata.*

Summing up, we have the following decidability results:

MSO-theory	**Model**	**Automata**
S1S	finite line	finite state automata
S1S	semi-infinite line	Büchi automata
S2S	infinite tree	Rabin tree automata

Summing up, we have the following decidability results:

MSO-theory	**Model**	**Automata**
S1S	finite line	finite state automata
S1S	semi-infinite line	Büchi automata
S2S	infinite tree	Rabin tree automata

What next?
to find infinite transition systems *in between* the infinite tree and the infinite grid that enjoy a decidable MSO-theory.

Basic ingredients of the transformational approach:

1. We start from a structure $\mathcal{T}$ that enjoys a decidable MSO-theory (e.g., the infinite binary tree)
2. We apply to $\mathcal{T}$ a suitable transformation that preserves the decidability of MSO-theories (e.g., interpretation), thus obtaining a new (decidable) structure $\mathcal{T}'$
3. We iterate the above construction to generate more and more structures

Basic ingredients of the transformational approach:

1. We start from a structure $\mathcal{T}$ that enjoys a decidable MSO-theory (e.g., the infinite binary tree)
2. We apply to $\mathcal{T}$ a suitable transformation that preserves the decidability of MSO-theories (e.g., interpretation), thus obtaining a new (decidable) structure $\mathcal{T}'$
3. We iterate the above construction to generate more and more structures

A noticeable class of transformations that preserve decidability of MSO-theories is the class of **MSO-compatible transformations**.

Definition (MSO-compatible transformation)

A transformation t for transition systems is said to be **MSO-compatible** if for any transition system $\mathcal{T}$ and any MSO-sentence ψ over $t(\mathcal{T})$, one can compute an MSO-sentence $\overleftarrow{\psi}$ over $\mathcal{T}$ (which depends on ψ only) such that

$$t(\mathcal{T}) \models \psi \quad \text{iff} \quad \mathcal{T} \models \overleftarrow{\psi}$$

Intuitively, MSO-compatibility allows one to map a property about $t(\mathcal{T})$ into a corresponding property about $\mathcal{T}$

$\Rightarrow$ If $\mathcal{T}$ has a decidable MSO-theory,
then $t(\mathcal{T})$ has a decidable MSO-theory as well.

Definition (MSO-compatible transformation)

A transformation t for transition systems is said to be **MSO-compatible** if for any transition system $\mathcal{T}$ and any MSO-sentence ψ over $t(\mathcal{T})$, one can compute an MSO-sentence $\overleftarrow{\psi}$ over $\mathcal{T}$ (which depends on ψ only) such that

$$t(\mathcal{T}) \models \psi \quad \text{iff} \quad \mathcal{T} \models \overleftarrow{\psi}$$

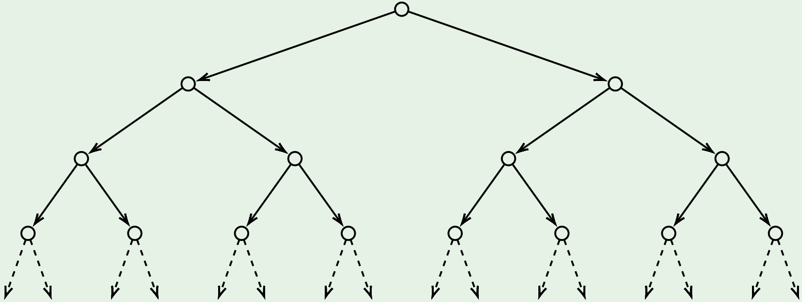
Intuitively, MSO-compatibility allows one to map a property about $t(\mathcal{T})$ into a corresponding property about $\mathcal{T}$

$\Rightarrow$ If $\mathcal{T}$ has a decidable MSO-theory, then $t(\mathcal{T})$ has a decidable MSO-theory as well.

The first transformation we consider is the MSO-interpretation.

Example (MSO-interpretation)

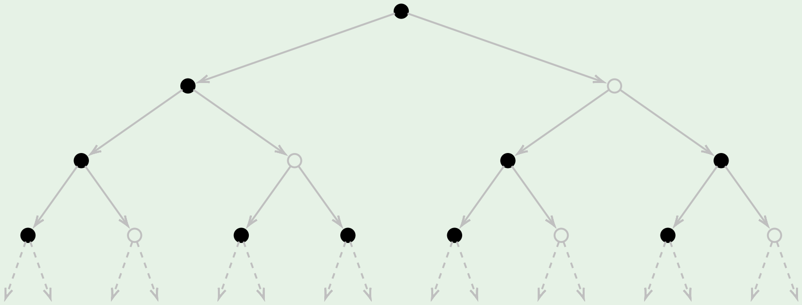
Consider the infinite binary tree $\mathcal{T}_2$.



We describe the **infinite ternary tree** $\mathcal{T}_3$ $(= t(\mathcal{T}_2))$ inside $\mathcal{T}_2$:

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.

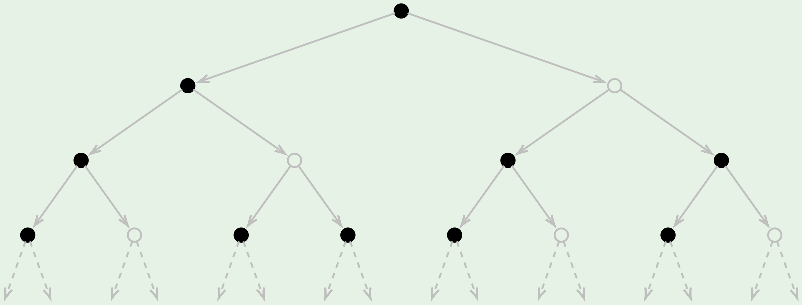


We describe the **infinite ternary tree** $\mathcal{T}_3$ $(= t(\mathcal{T}_2))$ inside $\mathcal{T}_2$:

- we select some vertices of $\mathcal{T}_2$ (black-colored ones)

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.



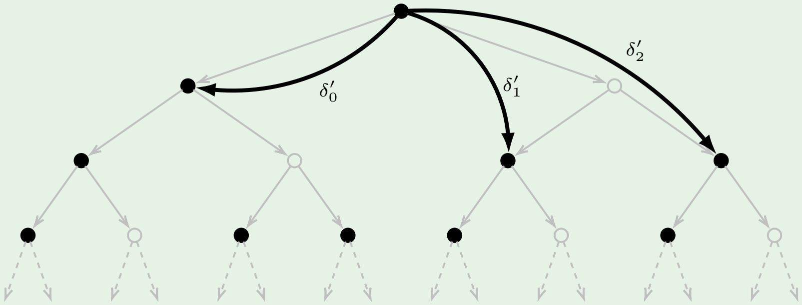
We describe the **infinite ternary tree** $\mathcal{T}_3$ $(= t(\mathcal{T}_2))$ inside $\mathcal{T}_2$:

- we select some vertices of $\mathcal{T}_2$ (black-colored ones)

  $\psi_{dom}(x) := \big(\delta_0 \cup (\delta_1 \circ \delta_0) \cup (\delta_1 \circ \delta_1)\big)^*(\varepsilon, x)$

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.



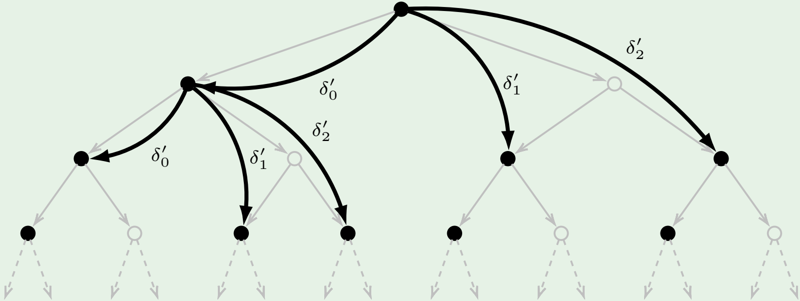
We describe the **infinite ternary tree** $\mathcal{T}_3$ $(= t(\mathcal{T}_2))$ inside $\mathcal{T}_2$:

- we select some vertices of $\mathcal{T}_2$ (black-colored ones)

  $\psi_{dom}(x) := \left(\delta_0 \cup (\delta_1 \circ \delta_0) \cup (\delta_1 \circ \delta_1)\right)^*(\varepsilon, x)$
- we then define the successor relations $\delta'_0, \delta'_1, \delta'_2$ of $\mathcal{T}_3$

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.

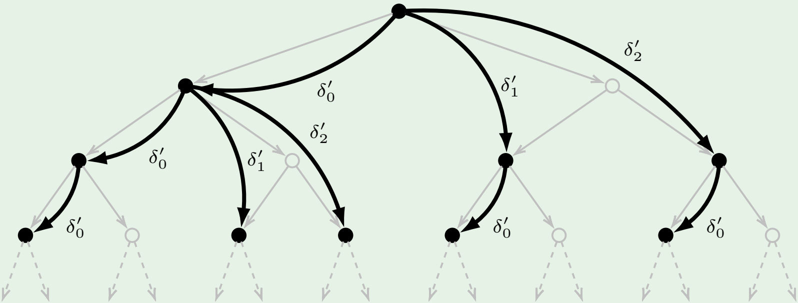


We describe the **infinite ternary tree** $\mathcal{T}_3$ $(= t(\mathcal{T}_2))$ inside $\mathcal{T}_2$:

- we select some vertices of $\mathcal{T}_2$ (black-colored ones)
  $\psi_{dom}(x) := \big(\delta_0 \cup (\delta_1 \circ \delta_0) \cup (\delta_1 \circ \delta_1)\big)^*(\varepsilon, x)$
- we then define the successor relations $\delta'_0, \delta'_1, \delta'_2$ of $\mathcal{T}_3$

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.



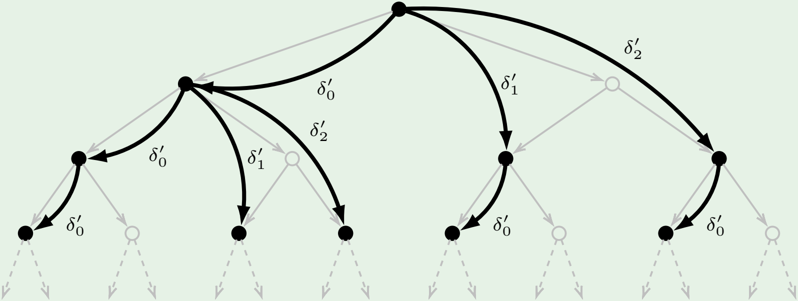
We describe the **infinite ternary tree** $\mathcal{T}_3$ $(= t(\mathcal{T}_2))$ inside $\mathcal{T}_2$:

- we select some vertices of $\mathcal{T}_2$ (black-colored ones)

  $\psi_{dom}(x) := \left(\delta_0 \cup (\delta_1 \circ \delta_0) \cup (\delta_1 \circ \delta_1)\right)^*(\varepsilon, x)$
- we then define the successor relations $\delta'_0, \delta'_1, \delta'_2$ of $\mathcal{T}_3$

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.



We describe the **infinite ternary tree** $\mathcal{T}_3$ ($= t(\mathcal{T}_2)$) inside $\mathcal{T}_2$:

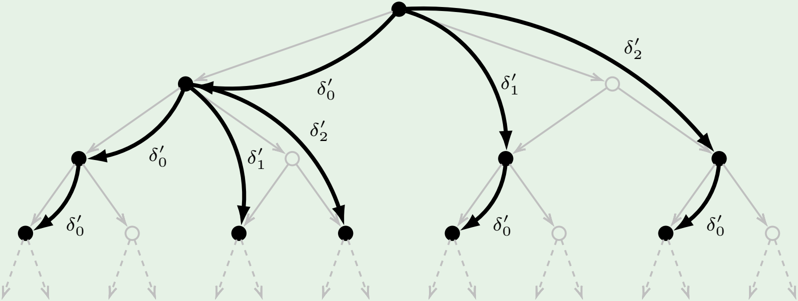
- we select some vertices of $\mathcal{T}_2$ (black-colored ones)

  $\psi_{dom}(x) := (\delta_0 \cup (\delta_1 \circ \delta_0) \cup (\delta_1 \circ \delta_1))^*(\varepsilon, x)$
- we then define the successor relations $\delta'_0, \delta'_1, \delta'_2$ of $\mathcal{T}_3$

  $\psi_{\delta'_0}(x, y) := \delta_0(x, y) \qquad \psi_{\delta'_{i+1}}(x, y) := (\delta_1 \circ \delta_i)(x, y)$

Example (MSO-interpretation)

Consider the infinite binary tree $\mathcal{T}_2$.



Any MSO-formula ϕ over $\mathcal{T}_3$ can be mapped into a corresponding formula $\overleftarrow{\phi}$ over $\mathcal{T}_2$.

For instance, the formula $\phi = \forall\, x.\ \exists\, y.\ \delta_2(x, y)$ becomes
$\overleftarrow{\phi} = \forall\, x.\ (\psi_{dom}(x)\ \rightarrow\ \exists\, y.\ (\psi_{dom}(y)\ \wedge\ \psi_{\delta'_2}(x, y)))$

Definition (MSO-interpretation)

An **MSO-interpretation** is a tuple of MSO-formulas

$$\underbrace{\psi_{dom}(x)}_{\text{domain formula}} \quad \underbrace{\psi_{b_1}(x,y) \quad \dots \quad \psi_{b_k}(x,y)}_{\text{edge formulas}} \quad \underbrace{\psi_{d_1}(x) \quad \dots \quad \psi_{d_m}(x)}_{\text{color formulas}}$$

It defines a B-labeled D-colored structure $\mathcal{T}'$ *inside* an A-labeled C-colored structure $\mathcal{T}$ as follows:

Definition (MSO-interpretation)

An **MSO-interpretation** is a tuple of MSO-formulas

$$\underbrace{\psi_{dom}(x)}_{\text{domain formula}} \quad \underbrace{\psi_{b_1}(x,y) \quad \ldots \quad \psi_{b_k}(x,y)}_{\text{edge formulas}} \quad \underbrace{\psi_{d_1}(x) \quad \ldots \quad \psi_{d_m}(x)}_{\text{color formulas}}$$

It defines a B-labeled D-colored structure $\mathcal{T}'$ *inside* an A-labeled C-colored structure $\mathcal{T}$ as follows:

- if $\psi_{dom}(x)$ holds in $\mathcal{T}$ by interpreting x as v, then v is a vertex of $\mathcal{T}'$

Definition (MSO-interpretation)

An **MSO-interpretation** is a tuple of MSO-formulas

$$\underbrace{\psi_{dom}(x)}_{\text{domain formula}} \quad \underbrace{\psi_{b_1}(x,y) \quad \dots \quad \psi_{b_k}(x,y)}_{\text{edge formulas}} \quad \underbrace{\psi_{d_1}(x) \quad \dots \quad \psi_{d_m}(x)}_{\text{color formulas}}$$

It defines a B-labeled D-colored structure $\mathcal{T}'$ *inside* an A-labeled C-colored structure $\mathcal{T}$ as follows:

- if $\psi_{dom}(x)$ holds in $\mathcal{T}$ by interpreting x as v, then v is a vertex of $\mathcal{T}'$
- if $\psi_{b_i}(x,y)$ holds in $\mathcal{T}$ by interpreting x as u, resp. y as v then (u,v) is an b_i-labeled transition of $\mathcal{T}'$

Definition (MSO-interpretation)

An **MSO-interpretation** is a tuple of MSO-formulas

$$\underbrace{\psi_{dom}(x)}_{\text{domain formula}} \quad \underbrace{\psi_{b_1}(x,y) \quad \dots \quad \psi_{b_k}(x,y)}_{\text{edge formulas}} \quad \underbrace{\psi_{d_1}(x) \quad \dots \quad \psi_{d_m}(x)}_{\text{color formulas}}$$

It defines a B-labeled D-colored structure $\mathcal{T}'$ *inside* an A-labeled C-colored structure $\mathcal{T}$ as follows:

- if $\psi_{dom}(x)$ holds in $\mathcal{T}$ by interpreting x as v, then v is a vertex of $\mathcal{T}'$
- if $\psi_{b_i}(x,y)$ holds in $\mathcal{T}$ by interpreting x as u, resp. y as v then (u,v) is an b_i-labeled transition of $\mathcal{T}'$
- if $\psi_{d_j}(x)$ holds in $\mathcal{T}$ by interpreting x as v, then v is a d_j-colored vertex of $\mathcal{T}'$

Theorem

MSO-interpretations are MSO-compatible.

Theorem

MSO-interpretations are MSO-compatible.

Proof (sketch)

Rewrite a given MSO-sentence ψ over $\mathcal{T}'$ into a corresponding MSO-sentence $\overleftarrow{\psi}$ over $\mathcal{T}$:

- if $\psi = \delta'_{b_i}(x, y)$, then $\overleftarrow{\psi} := \psi_{b_i}(x, y)$
- if $\psi = P_{d_i}(x)$, then $\overleftarrow{\psi} := \psi_{d_i}(x)$
- if $\psi = \exists\, x.\ \varphi(x)$, then $\overleftarrow{\psi} := \exists\, x.\ (\psi_{dom}(x)\ \wedge\ \overleftarrow{\varphi}(x))$
- ...

Theorem

MSO-interpretations are MSO-compatible.

Proof (sketch)

Rewrite a given MSO-sentence ψ over $\mathcal{T}'$ into a corresponding MSO-sentence $\overleftarrow{\psi}$ over $\mathcal{T}$:

- if $\psi = \delta'_{b_i}(x, y)$, then $\overleftarrow{\psi} := \psi_{b_i}(x, y)$
- if $\psi = P_{d_i}(x)$, then $\overleftarrow{\psi} := \psi_{d_i}(x)$
- if $\psi = \exists\, x.\ \varphi(x)$, then $\overleftarrow{\psi} := \exists\, x.\ (\psi_{dom}(x) \ \wedge\ \overleftarrow{\varphi}(x))$
- ...

Corollary

The infinite ternary tree $\mathcal{T}_3$ has a decidable MSO-theory.

Most MSO-formulas *with two free variables* can be conveniently written as **regular** (path) **expressions**:

- let A and C be disjoint sets of edge labels and vertex colors
- for each label $a \in A$, we introduce an **inverse label** $\bar{a}$ denoting a-labeled edges traversed in backward direction
- we describe paths traversing edges in both directions by words over the alphabet $A \cup \bar{A} \cup C$

Example

The set of paths on an A-labeled C-colored transition system that

- start from a vertex with color c
- traverse a sequence of edges labeled with a
- reach a vertex colored with c'
- and finally traverse a edge labeled with a' in backward direction

is described by the regular expression $c \cdot a^* \cdot c' \cdot \bar{a}'$

Fact

Regular path expressions are **shorthands**
of (a subset of) MSO-formulas with two free variables.

For instance:

- the expression a
  abbreviates $\psi(x,y) := \delta_a(x,y)$
- the expression $a \cdot a'$
  abbreviates $\psi(x,y) := \exists\, z.\ \delta_a(x,z) \ \wedge\ \delta_{a'}(z,y)$
- the expression $a + \overline{a}'$
  abbreviates $\psi(x,y) := \delta_a(x,y) \ \vee\ \delta_{a'}(y,x)$
- the expression a^*
  abbreviates $\psi(x,y) := \delta_a^*(x,y)$

Fact

Regular path expressions are **shorthands** of (a subset of) MSO-formulas with two free variables.

For instance:

- the expression a
  abbreviates $\psi(x,y) := \delta_a(x,y)$
- the expression $a \cdot a'$
  abbreviates $\psi(x,y) := \exists\, z.\ \delta_a(x,z) \ \wedge\ \delta_{a'}(z,y)$
- the expression $a + \overline{a}'$
  abbreviates $\psi(x,y) := \delta_a(x,y) \ \vee\ \delta_{a'}(y,x)$
- the expression a^*
  abbreviates $\psi(x,y) := \delta_a^*(x,y)$

Note: the converse is not true in the general case (e.g., $\psi(x,y) := \nexists\, z.\ (\delta_0(z,x) \ \vee\ \delta_1(z,y)) \ \wedge\ \nexists\, z.\ (\delta_0(y,z) \ \vee\ \delta_1(y,z))$).

However, regular path expressions suffice for most cases.

For the usual MSO-interpretations, we can replace every edge formula $\psi_b(x, y)$ with a regular path expression, namely, a regular language over $A \cup \bar{A} \cup C$.

Definition (Inverse rational mapping)

A **rational mapping** is a function $h : B \rightarrow \mathscr{P}((A \cup \bar{A} \cup C)^*)$ such that $\forall b \in B$, $h(b)$ is a regular language over $A \cup \bar{A} \cup C$.

For the usual MSO-interpretations, we can replace every edge formula $\psi_b(x, y)$ with a regular path expression, namely, a regular language over $A \cup \bar{A} \cup C$.

Definition (Inverse rational mapping)

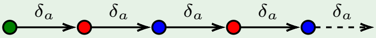
A **rational mapping** is a function $h : B \rightarrow \mathscr{P}((A \cup \bar{A} \cup C)^*)$ such that $\forall b \in B$, $h(b)$ is a regular language over $A \cup \bar{A} \cup C$.

The 'inverse' h^{-1} of h (**inverse rational mapping**) can be applied to an A-labeled transition system $\mathcal{T}$ to produce the B-labeled transition system $h^{-1}(\mathcal{T})$ such that:

- $h^{-1}(\mathcal{T})$ has the same vertices of $\mathcal{T}$
- (u, v) is a b-labeled edge of $h^{-1}(\mathcal{T})$ iff $\mathcal{T}$ contains a w-**marked path** from u to v, for some $w \in h(b)$.

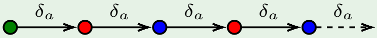
Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = \left(\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0\right)$

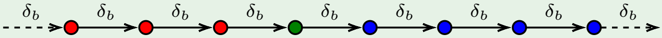


Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



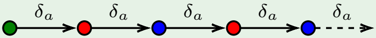
Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$



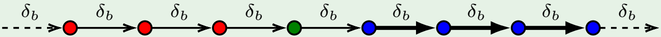
We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

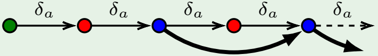


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have *b*-labeled edges of 4 types:

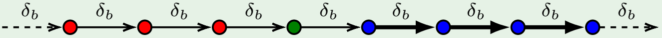
- between *blue*-colored vertices

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

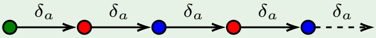


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

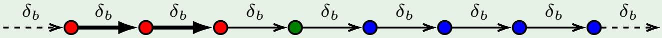
- between *blue*-colored vertices: $\textit{blue} \cdot a \cdot a$

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

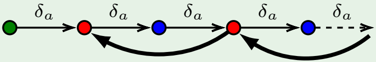


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

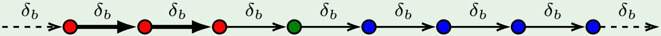
- between *blue*-colored vertices: $blue \cdot a \cdot a$
- between *red*-colored vertices

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

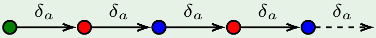


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

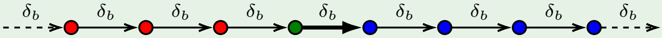
- between *blue*-colored vertices: $blue \cdot a \cdot a$
- between *red*-colored vertices: $red \cdot \bar{a} \cdot \bar{a}$

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

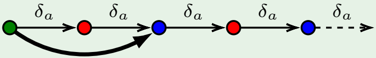


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

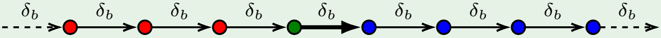
- between *blue*-colored vertices: $\textit{blue} \cdot a \cdot a$
- between *red*-colored vertices: $\textit{red} \cdot \bar{a} \cdot \bar{a}$
- from the *green*-colored vertex to the *blue*-colored vertex

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

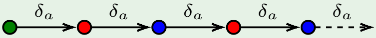


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

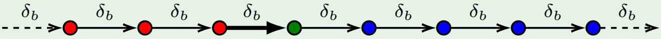
- between *blue*-colored vertices: *blue* $\cdot a \cdot a$
- between *red*-colored vertices: *red* $\cdot \bar{a} \cdot \bar{a}$
- from the *green*-colored vertex to the *blue*-colored vertex: $0 \cdot a \cdot a$

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

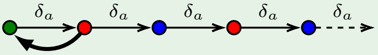


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

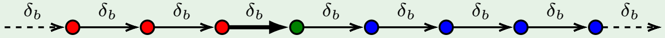
- between *blue*-colored vertices: $blue \cdot a \cdot a$
- between *red*-colored vertices: $red \cdot \bar{a} \cdot \bar{a}$
- from the *green*-colored vertex to the *blue*-colored vertex: $0 \cdot a \cdot a$
- from the *red*-colored vertex to the *green*-colored one

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$

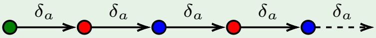


We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have b-labeled edges of 4 types:

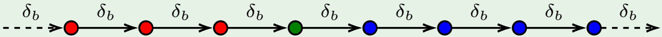
- between *blue*-colored vertices: $blue \cdot a \cdot a$
- between *red*-colored vertices: $red \cdot \bar{a} \cdot \bar{a}$
- from the *green*-colored vertex to the *blue*-colored vertex: $0 \cdot a \cdot a$
- from the *red*-colored vertex to the *green*-colored one: $\bar{a} \cdot 0$

Example (Inverse rational mapping)

Colored semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_{even}, P_{odd}, P_0)$



Colored bi-infinite line $\mathcal{L}_{-\omega,\omega} = (\mathbb{Z}, \delta_b, P_{pos}, P_{neg}, P_0)$



We define $\mathcal{L}_{-\omega,\omega}$ inside $\mathcal{L}_\omega$.
In $\mathcal{L}_{-\omega,\omega}$ we have *b*-labeled edges of 4 types:

- between *blue*-colored vertices: $blue \cdot a \cdot a$
- between *red*-colored vertices: $red \cdot \bar{a} \cdot \bar{a}$
- from the *green*-colored vertex to the *blue*-colored vertex: $0 \cdot a \cdot a$
- from the *red*-colored vertex to the *green*-colored one: $\bar{a} \cdot 0$

$\Rightarrow \quad h(b) = blue \cdot a \cdot a + red \cdot \bar{a} \cdot \bar{a} + 0 \cdot a \cdot a + \bar{a} \cdot 0$

Similarly, color formulas can be replaced with rational markings:

Definition (Rational marking)

A **rational marking** is a function $k : D \rightarrow \mathscr{P}((A \cup \bar{A} \cup C)^*)$ such that $\forall d \in D$, $k(d)$ is a regular language over $A \cup \bar{A} \cup C$.

It induces a **recoloring** of the rooted transition system $\mathcal{T}$ as follows:

- for each $d \in D$, the color d is assigned
  to all vertices v of $\mathcal{T}$ such that there is
  a w-marked path from the root to v, for some $w \in k(d)$.

Similarly, color formulas can be replaced with rational markings:

Definition (Rational marking)

A **rational marking** is a function $k : D \rightarrow \mathscr{P}((A \cup \bar{A} \cup C)^*)$ such that $\forall d \in D$, $k(d)$ is a regular language over $A \cup \bar{A} \cup C$.

It induces a **recoloring** of the rooted transition system $\mathcal{T}$ as follows:

- for each $d \in D$, the color d is assigned to all vertices v of $\mathcal{T}$ such that there is a w-marked path from the root to v, for some $w \in k(d)$.

Example (Rational marking)

The $\{pos, neg, 0\}$-coloring of the bi-infinite line is encoded in the rooted semi-infinite line $\mathcal{L}_\omega = (\mathbb{N}, \delta_a, P_0)$ via the rational marking k such that

$$k(pos) = 0 \cdot a \cdot a \cdot (a \cdot a)^* \quad k(neg) = 0 \cdot a \cdot (a \cdot a)^* \quad k(0) = 0$$

Finally, domain formulas can be replaced with rational restrictions:

Definition (Rational restriction)

A **rational restriction** is specified by a regular language L over $A \cup \bar{A} \cup C$.

It induces a **restriction** $\mathcal{T}|_L$ of the rooted transition system $\mathcal{T}$ as follows:

- for each vertex v of $\mathcal{T}$, v belongs to $\mathcal{T}|_L$ iff there is a w-marked path from the root to v, for some $w \in L$.

Another useful transformation is the unfolding:

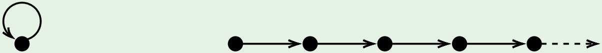
Definition (Unfolding)

The **unfolding** of a rooted transition system $\mathcal{T}$ is the tree $\mathcal{U}nf(\mathcal{T})$ such that:

- the vertices of $\mathcal{U}nf(\mathcal{T})$ are all and only the finite paths in $\mathcal{T}$ originating from the root
- the edges of $\mathcal{U}nf(\mathcal{T})$ are given by the path-extension relation, namely, if π is path in $\mathcal{T}$ from the root and π' is the extension of π with an a-labeled edge, then (π, π') is an a-labeled edge in $\mathcal{U}nf(\mathcal{T})$
- the color of a vertex in $\mathcal{U}nf(\mathcal{T})$ is the color of the target vertex of the corresponding path in $\mathcal{T}$

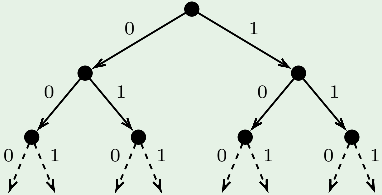
Example (unfoldings)

Example (unfoldings)

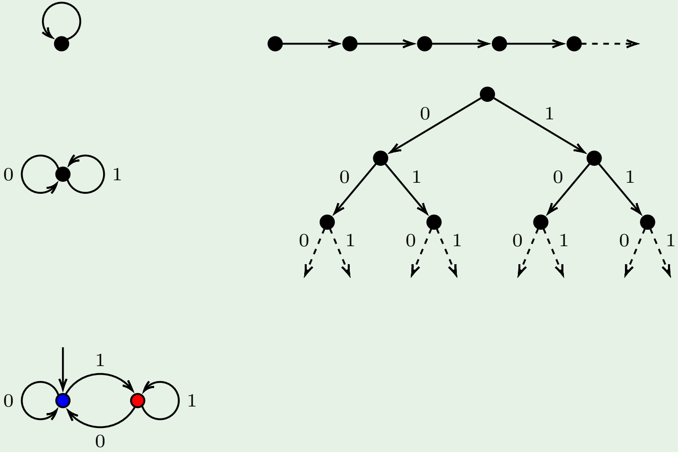


Example (unfoldings)

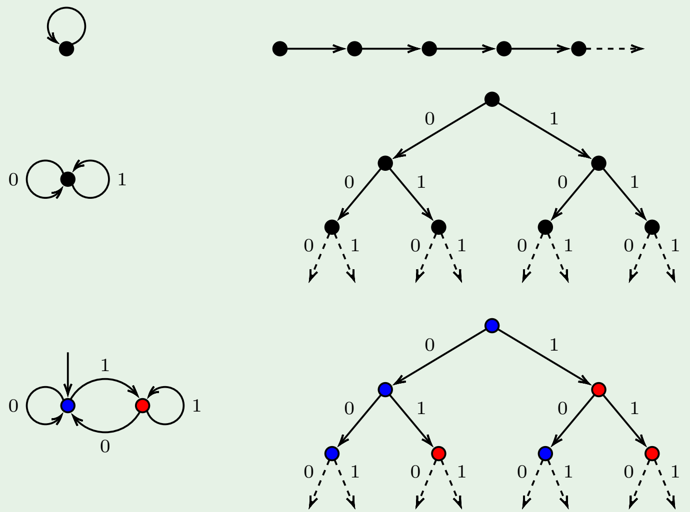
Example (unfoldings)



Example (unfoldings)



Example (unfoldings)



Theorem (Semenov-Muchnik '84 – proved by Walukiewicz '96)

The unfolding operation is MSO-compatible.

$\Rightarrow$ Since finite transition systems enjoy decidable MSO-theories, Muchnik's Theorem subsumes Büchi's and Rabin's theorems (in fact, the proof is strongly based on Rabin's Theorem...)

We know that MSO-interpretation and unfolding preserve the decidability of MSO-theories of transition systems.

We can start from finite (hence decidable) graphs and iterate MSO-interpretation and unfolding:

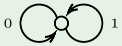
$$\begin{aligned} \mathit{Graph}_0 &:= \{\text{finite rooted graphs}\} \\ \mathit{Tree}_{n+1} &:= \{\text{trees obtained by unfolding graphs in } \mathit{Graph}_n\} \\ \mathit{Graph}_n &:= \{\text{rooted graphs obtained via} \\ &\qquad \text{interpretation from trees in } \mathit{Tree}_n\} \end{aligned}$$

(e.g., $\mathit{Tree}_1 = \{\text{regular trees}\}$)

$\Rightarrow$ a **hierarchy** of graphs with decidable MSO-theories arises (this is commonly known as **Caucal's hierarchy**)

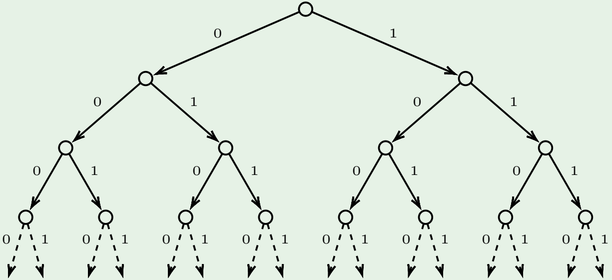
Example

We start from the finite graph



Example

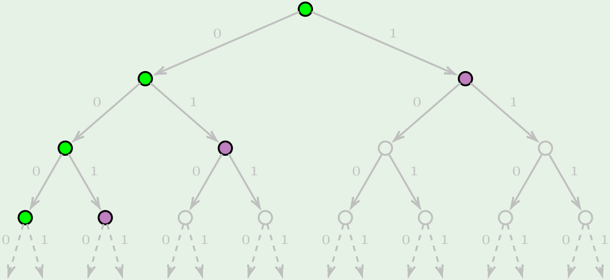
We unfold it, obtaining the infinite binary tree ...



Example

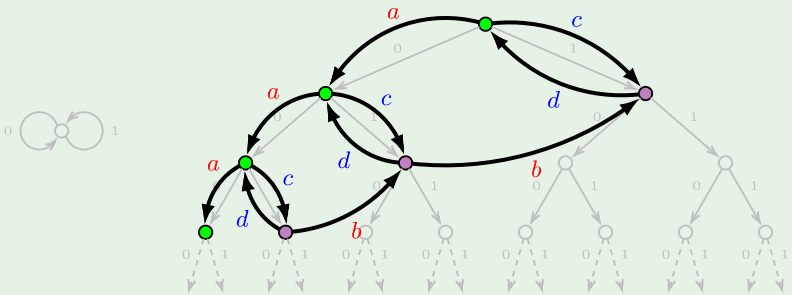
... we apply the rational marking

$k(A) = 0^*$, $k(B) = 0^*1$



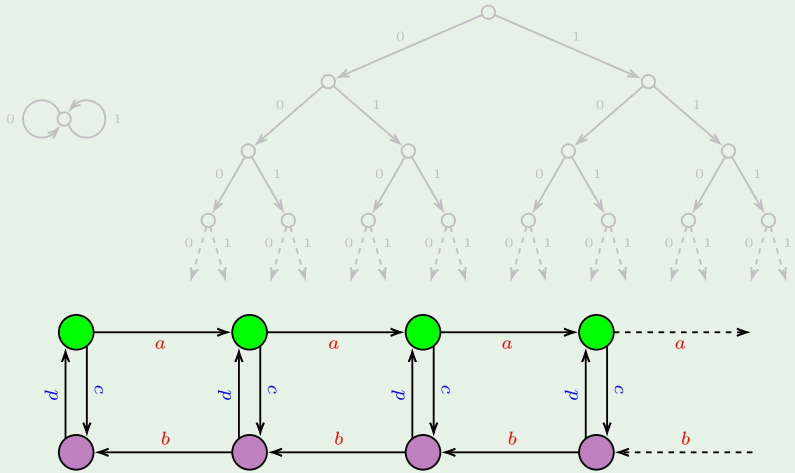
Example

... the inverse rational mapping
$h(a) = 0,\ h(b) = \bar{1}\bar{0}1,\ h(c) = 1,\ h(d) = \bar{1}$



Example

... and finally the rational restriction $L = 0^* + 0^*1$, obtaining the following transition system (do you remember it?)



Go