A Contraction Method to Decide MSO Theories of Deterministic Trees

Angelo Montanari and Gabriele Puppis

Departement of Mathematics and Computer Science
University of Udine, Italy
{angelo.montanari, gabriele.puppis}@dimi.uniud.it

What is the talk about?

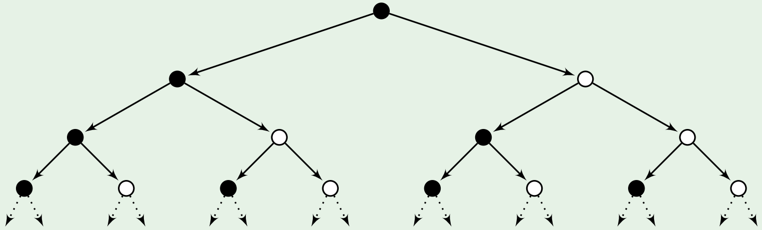
We shall consider the **model-checking** problem for **Monadic Second-Order (MSO) logic** over **deterministic colored trees**.

Example

To decide whether the MSO formula

$$\varphi(X) \ = \ X(\texttt{root}) \ \wedge \ \forall\, x, y.\ \big(\texttt{left\_child}(x, y) \ \rightarrow \ X(y)\big)$$

holds in the binary tree by interpreting the variable X with the set of *black colored vertices*:



Outline

1. *Short introduction*
   (the **automaton-based approach** to model-checking)
2. *The proposed method*
   (extension of **Elgot-Rabins's contraction method** to trees)
3. *Application examples*
   (**reducible trees**, closure properties, and open problems)

Theorem (Rabin '69)

For every MSO formula $\varphi(X_1, ..., X_n)$, we can compute a Rabin tree automaton $\mathcal{A}$ over the alphabet $C = \mathscr{P}(\{1, ..., n\})$ (and vice versa) such that, for every C-colored tree T,

$$T \vDash \varphi(X_1, ..., X_n) \quad \Leftrightarrow \quad T \in \mathscr{L}(\mathcal{A})$$

(read $\varphi(X_1, ..., X_n)$ holds in T iff $\mathcal{A}$ accepts T).

Definition

The **acceptance problem** Acc_T of a tree T is the problem of deciding whether, for any given tree automaton $\mathcal{A}$, $T \in \mathscr{L}(\mathcal{A})$.

Corollary

The model-checking problem of a tree T is reducible to the acceptance problem of T.

Fact

For any **regular** tree T, Acc_T *is decidable* (simply test, for any automaton $\mathcal{A}$, whether the language $\mathscr{L}(\mathcal{A}) \cap \{T\}$ is non-empty).

Problem

What about non-regular trees?

Solution idea

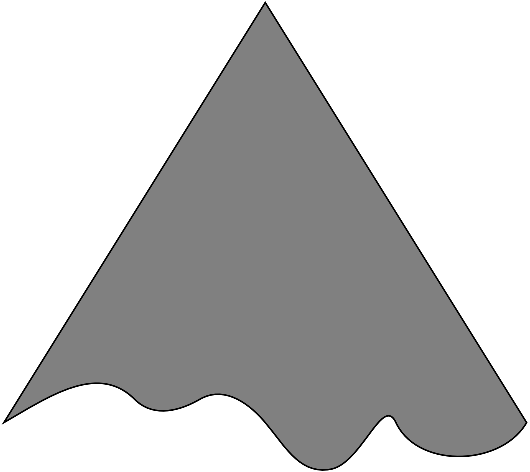
Generalize "**Contraction method**" (Elgot-Rabin '66) to trees.

Given a (non-regular) tree T and an automaton $\mathcal{A}$:

1. decompose T into **factors**,
2. 'distill' the relevant **features** of each factor F w.r.t. the behaviors of $\mathcal{A}$ and collect them into an $\mathcal{A}$-**type**,
3. reason on the **contraction** tree (i.e., a tree-shaped arrangement of $\mathcal{A}$-types).

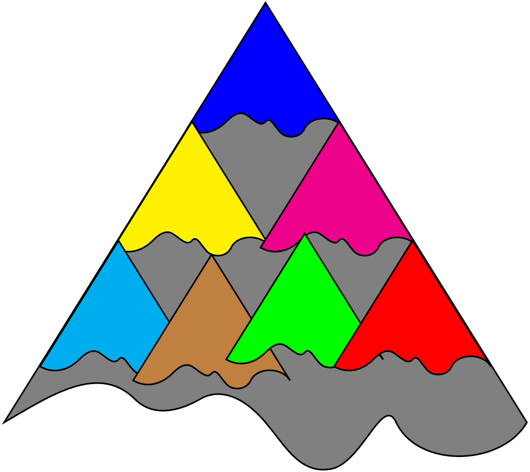
A picture of the method:

A tree automaton can have **similar behaviors** on different trees ...



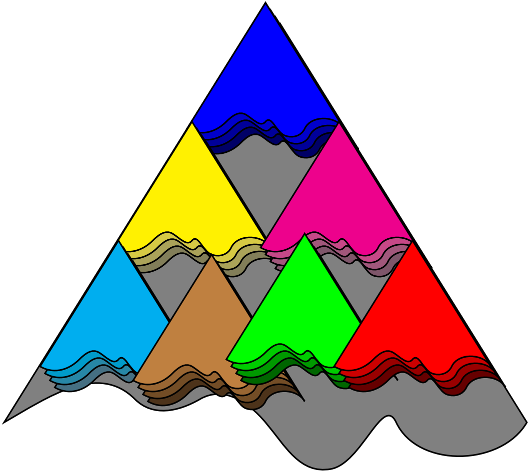
A picture of the method:

Given a tree T and an automaton $\mathcal{A}$, decompose T into **factors** ...



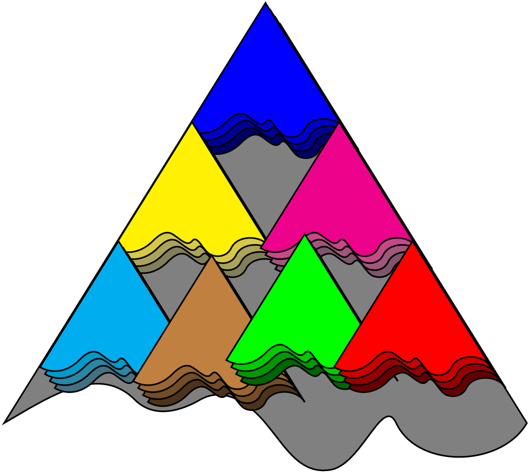
A picture of the method:

... and then consider the **equivalence classes** induced by the behavior of $\mathcal{A}$ on each factor.



A picture of the method:

$\Rightarrow$ We can replace $\mathcal{A}$ with an automaton $\vec{\mathcal{A}}$ that runs on the (possibly regular) abstracted tree and mimics $\mathcal{A}$.



Basic ingredients

The following notions will be briefly explained in the following:

- **factorization** Π of a tree T,
- **marked factor** $\Pi^{+}(v)$,
- $\mathcal{A}$-**type** $[\Pi^{+}(v)]_{\mathcal{A}}$,
- $\mathcal{A}$-**contraction** $\overrightarrow{T}$.

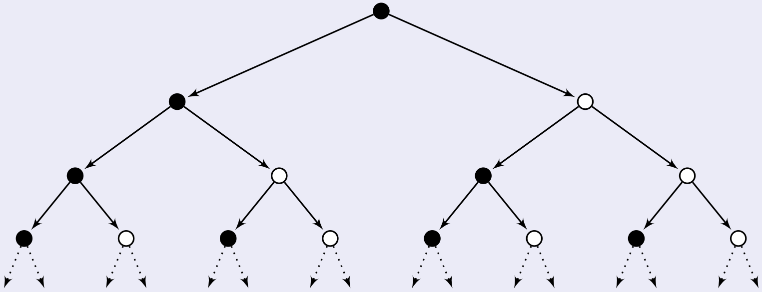
We will use the above notions to reduce
an instance of Acc_{T} to a (hopefully simpler) instance of $Acc_{\overrightarrow{T}}$

(for instance, the case where $\overrightarrow{T}$ is a regular tree).

Definition

A **factorization** of a tree T is an *uncolored* tree Π such that:

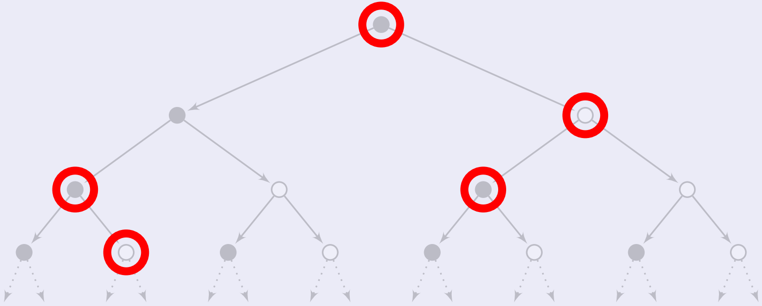
- $\mathcal{D}om(\Pi)$ is a subset of $\mathcal{D}om(T)$ that includes the root,
- the edges are given by the *ancestor relation* $\sqsubseteq$ of T,
- the edge labels are arbitrarily chosen from a finite set B.



Definition

A **factorization** of a tree T is an *uncolored* tree Π such that:

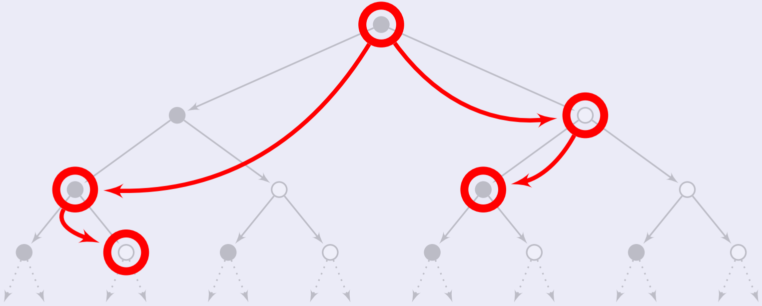
- $\mathcal{D}om(\Pi)$ is a subset of $\mathcal{D}om(T)$ that includes the root,
- the edges are given by the *ancestor relation* $\sqsubseteq$ of T,
- the edge labels are arbitrarily chosen from a finite set B.



Definition

A **factorization** of a tree T is an *uncolored* tree Π such that:

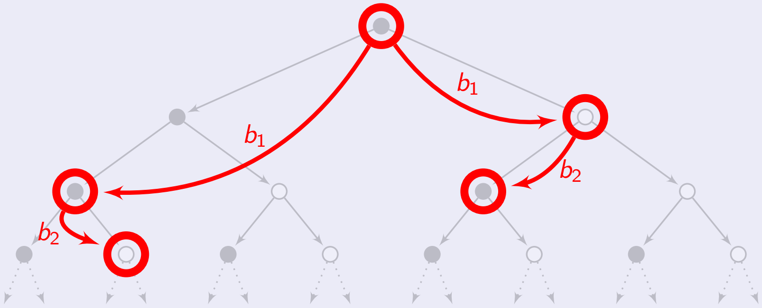
- $\mathcal{D}om(\Pi)$ is a subset of $\mathcal{D}om(T)$ that includes the root,
- the edges are given by the *ancestor relation* $\sqsubseteq$ of T,
- the edge labels are arbitrarily chosen from a finite set B.



Definition

A **factorization** of a tree T is an *uncolored* tree Π such that:

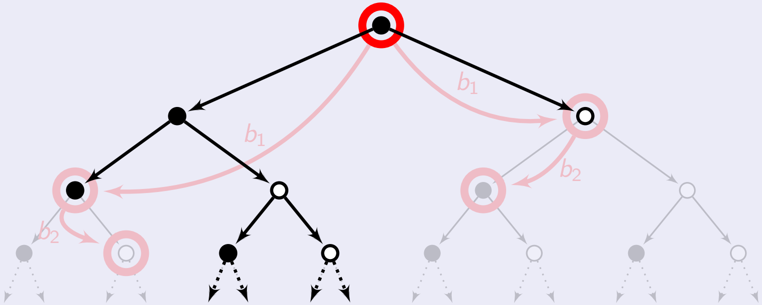
- $\mathcal{D}om(\Pi)$ is a subset of $\mathcal{D}om(T)$ that includes the root,
- the edges are given by the *ancestor relation* $\sqsubseteq$ of T,
- the edge labels are arbitrarily chosen from a finite set B.



Definition

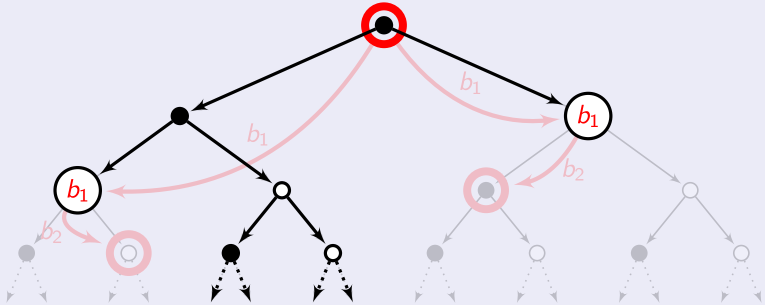
For every vertex v of Π, the **factor** $\Pi(v)$ **of** T **in** v is *the subgraph of* T *induced by the set*

$$\{w \in \mathcal{D}om(T) \ : \ v \sqsubseteq w \sqsubseteq v' \text{ for all successors } v' \text{ of } v \text{ in } \Pi\}$$



Definition

For every vertex v of Π, the **marked factor** $\Pi^+(v)$ is obtained from the (unmarked) factor $\Pi(v)$ by *recoloring* each leaf w with *the label of the incoming edge* of Π.

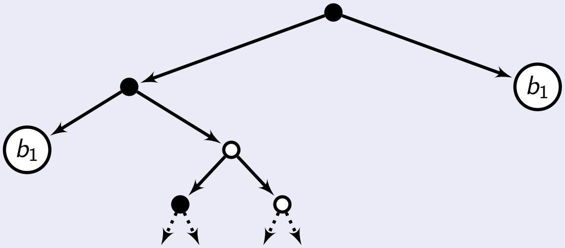


Definition

Given an automaton $\mathcal{A}$ and a marked factor F, the $\mathcal{A}$-**type** $[F]_{\mathcal{A}}$ is the *set of triples* of the form

$$\left(\begin{array}{c} R(\texttt{root}) \\ \{\mathcal{I}nf\mathcal{O}cc(R|\pi) \ : \ \pi \text{ branch of } F\} \\ \{(F(w), R(w), \mathcal{O}cc(R|w)) \ : \ w \text{ leaf of } F\} \end{array} \right)$$

over *all possible partial runs* R of $\mathcal{A}$ on F.

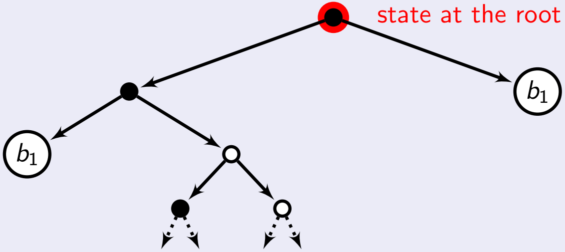


Definition

Given an automaton $\mathcal{A}$ and a marked factor F,
the $\mathcal{A}$-**type** $[F]_{\mathcal{A}}$ is the *set of triples* of the form

$$\begin{pmatrix} R(\texttt{root}) \\ \{\mathcal{I}nf\mathcal{O}cc(R|\pi) \,:\, \pi \text{ branch of } F\} \\ \{(F(w), R(w), \mathcal{O}cc(R|w)) \,:\, w \text{ leaf of } F\} \end{pmatrix}$$

over *all possible partial runs* R of $\mathcal{A}$ on F.

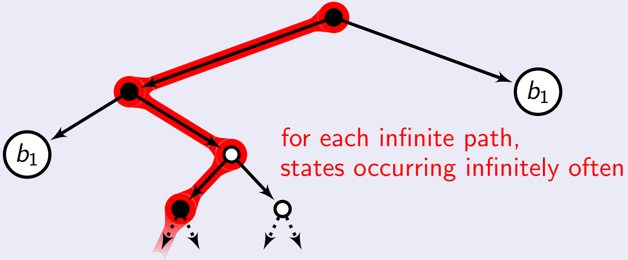


Definition

Given an automaton $\mathcal{A}$ and a marked factor F, the $\mathcal{A}$-**type** $[F]_{\mathcal{A}}$ is the *set of triples* of the form

$$\left(\begin{array}{c} R(\texttt{root}) \\ \{\mathcal{I}nf\mathcal{O}cc(R|\pi) \,:\, \pi \text{ branch of } F\} \\ \{(F(w), R(w), \mathcal{O}cc(R|w)) \,:\, w \text{ leaf of } F\} \end{array} \right)$$

over *all possible partial runs* R of $\mathcal{A}$ on F.

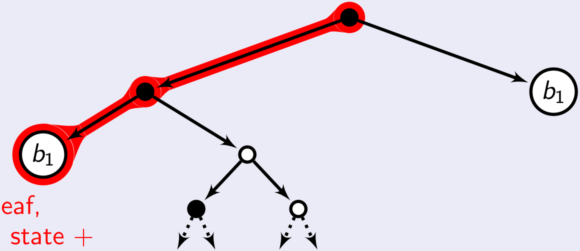


Definition

Given an automaton $\mathcal{A}$ and a marked factor F,
the $\mathcal{A}$-**type** $[F]_{\mathcal{A}}$ is the *set of triples* of the form

$$\begin{pmatrix} R(\mathtt{root}) \\ \{\mathcal{I}nf\mathcal{O}cc(R|\pi)\ :\ \pi \text{ branch of } F\} \\ \{(F(w), R(w), \mathcal{O}cc(R|w))\ :\ w \text{ leaf of } F\} \end{pmatrix}$$

over *all possible partial runs* R of $\mathcal{A}$ on F.



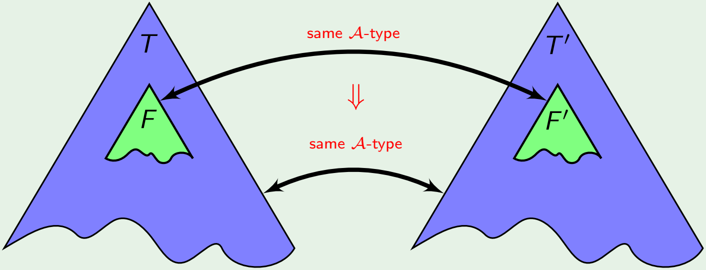
for each leaf,
marker + state +
states along access path

Proposition

The equivalence relation induced by $\mathcal{A}$-types is compatible with second-order tree substitutions.

Intuitive explanation

Consider a tree T and a factor F inside it.
Take F' such that $[F']_{\mathcal{A}} = \left[F\right]_{\mathcal{A}}$ and let $T' = T[\![F/F']\!]$.
Then $[T']_{\mathcal{A}} = \left[T\right]_{\mathcal{A}}$.



Remarks

We shall see that $\mathcal{A}$-types capture the concept of inditinguishability of trees w.r.t. the automaton $\mathcal{A}$.

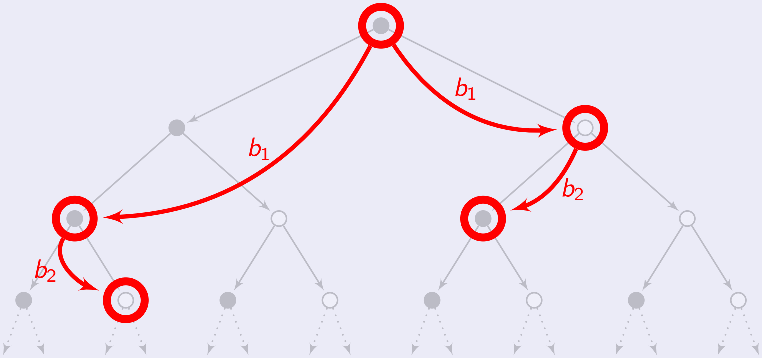
Moreover, the amount of information stored in an $\mathcal{A}$-type is bounded.

This implies that:

- there exist *only finitely many* $\mathcal{A}$-types (equivalently, the automaton $\mathcal{A}$ can distinguish between only finitely many classes of trees),
- we can see each $\mathcal{A}$-type as a *color*,
- we can arrange the $\mathcal{A}$-types of the factors of a tree T in a colored tree structure $\overrightarrow{T}$, called the $\mathcal{A}$-**contraction**.

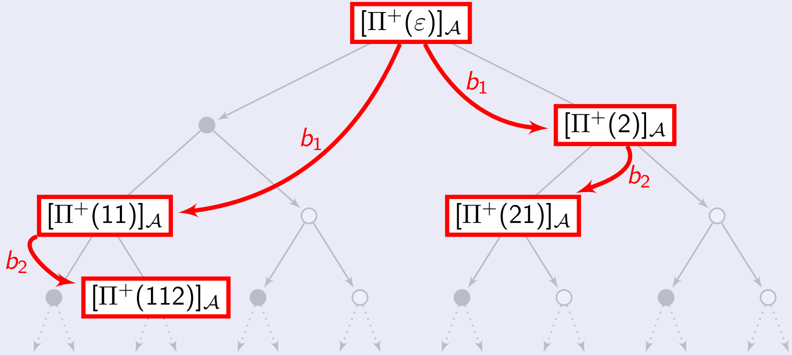
Definition

Given a tree T, an automaton $\mathcal{A}$, and a factorization Π of T, the $\mathcal{A}$-**contraction** $\vec{T}$ of T is the tree obtained from Π *by coloring each vertex v with the $\mathcal{A}$-type* $\left[\Pi^+(v)\right]_{\mathcal{A}}$.



Definition

Given a tree T, an automaton $\mathcal{A}$, and a factorization Π of T, the $\mathcal{A}$-**contraction** $\vec{T}$ of T is the tree obtained from Π *by coloring each vertex v with the $\mathcal{A}$-type $[\Pi^+(v)]_{\mathcal{A}}$.*



Remark

In the general case, a contraction can be a **non-deterministic** tree.

In order to reason by means of Rabin automata, we need to identify contractions with suitable deterministic trees.

Definition

An $\mathcal{A}$-contraction is said to be **valid**
if it is bisimilar to a deterministic tree.

From now on, we restrict ourselves to *valid contractions only*...

Theorem (Main result)

Given a (valid) $\mathcal{A}$-contraction $\overrightarrow{T}$ of T, we can build a suitable automaton $\overrightarrow{\mathcal{A}}$, running on $\overrightarrow{T}$, such that

$$\overrightarrow{T} \in \mathscr{L}(\overrightarrow{\mathcal{A}}) \quad \Leftrightarrow \quad T \in \mathscr{L}(\mathcal{A}).$$

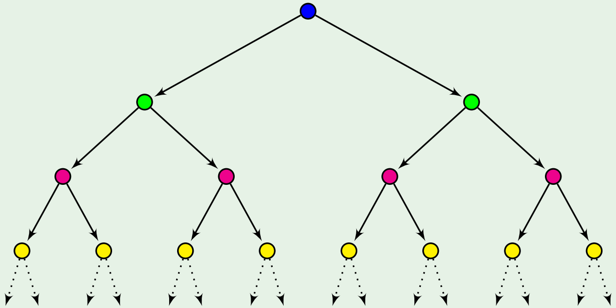
Proof idea

Define $\overrightarrow{\mathcal{A}}$ in such a way that it *mimics* the computations of $\mathcal{A}$ on T at a "coarser level":

- the input alphabet of $\overrightarrow{\mathcal{A}}$ is the set of all $\mathcal{A}$-types
- the states of $\overrightarrow{\mathcal{A}}$ encode the finite amount of information processed by $\mathcal{A}$ up to a certain point,
- the transitions of $\overrightarrow{\mathcal{A}}$ compute new states by "merging" the information of the current state with the information provided by the input symbol (i.e., the $\mathcal{A}$-type of the current factor).

Application example

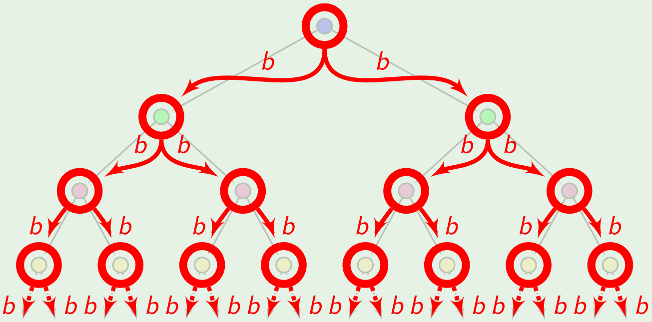
Let T be a tree with *homogeneously-colored levels*,
and Π the factorization of T with $\mathcal{D}om(\Pi) = \mathcal{D}om(T)$.
Consider now the marked factors at each level: *their $\mathcal{A}$-types uniquely depends on the color of the level they belong to*.
$\Rightarrow$ The $\mathcal{A}$-contraction $\overrightarrow{T}$ of T is bisimilar to a colored line L.



Application example

Let T be a tree with *homogeneously-colored levels*,
and Π the factorization of T with $\mathcal{D}om(\Pi) = \mathcal{D}om(T)$.
Consider now the marked factors at each level: *their $\mathcal{A}$-types uniquely depends on the color of the level they belong to.*

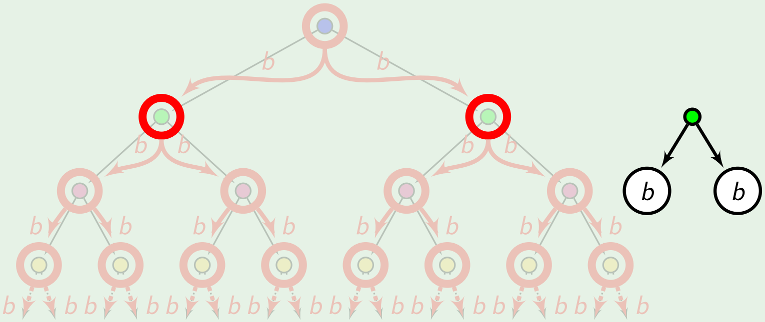
$\Rightarrow$ The $\mathcal{A}$-contraction $\overrightarrow{T}$ of T is bisimilar to a colored line L.



Application example

Let T be a tree with *homogeneously-colored levels*,
and Π the factorization of T with $\mathcal{D}om(\Pi) = \mathcal{D}om(T)$.
Consider now the marked factors at each level: *their $\mathcal{A}$-types uniquely depends on the color of the level they belong to.*

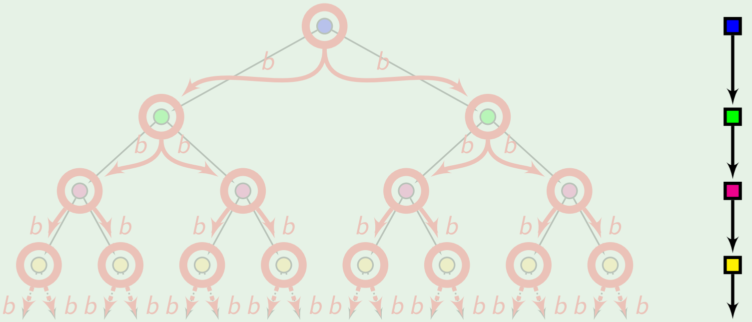
$\Rightarrow$ The $\mathcal{A}$-contraction $\overrightarrow{T}$ of T is bisimilar to a colored line L.



Application example

Let T be a tree with *homogeneously-colored levels*,
and Π the factorization of T with $\mathcal{D}om(\Pi) = \mathcal{D}om(T)$.
Consider now the marked factors at each level: *their $\mathcal{A}$-types uniquely depends on the color of the level they belong to*.

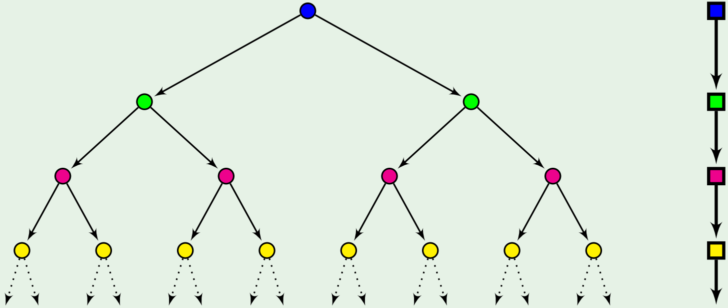
$\Rightarrow$ The $\mathcal{A}$-contraction $\vec{T}$ of T is bisimilar to a colored line L.



Application example

In this way we proved a simplified version of Muchnik's Theorem:

Acc_T is reducible to Acc_L.



One can also iterate reductions in order to show that the acceptance problem of a tree T is decidable ...

Example

Consider the problem of deciding if $T \in \mathscr{L}(\mathcal{A})$:

If T has an $\mathcal{A}$-contraction $\vec{T}$, and
$\vec{T}$ has a *regular* $\vec{\mathcal{A}}$-contraction $\vec{\vec{T}}$

Then we can decide if $\vec{\vec{T}} \in \mathscr{L}(\vec{\vec{\mathcal{A}}})$, $\vec{T} \in \mathscr{L}(\vec{\mathcal{A}})$, and $T \in \mathscr{L}(\mathcal{A})$.

Definition

It comes natural to define a

hierarchy of reducible trees:

- **rank** 0 **trees** := regular trees
- **rank** $n + 1$ **trees** := trees enjoying a rank n $\mathcal{A}$-contraction, for any automaton $\mathcal{A}$.

Corollary

The acceptance problem of any reducible tree is decidable.

Theorem

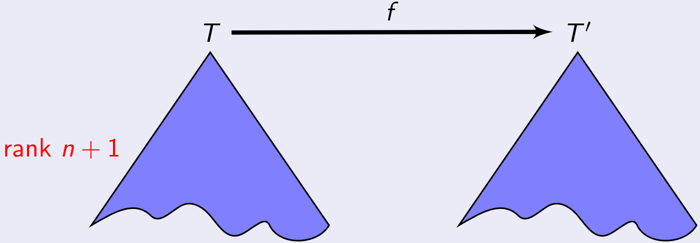
Rank n trees are closed under the following operations:

- **rational colorings**
  specified by regular path expressions,
  in a similar way to inverse rational mappings
  (alternative specifications in terms of Mealy tree automata)
- **rational colorings with bounded lookahead**
  rational colorings extended with the facility of inspecting the subtree issued from current position, up to bounded depth
- **regular tree morphisms**
  specified by a tuple of regular trees $F_{c_1}, \ldots, F_{c_k}$
  and mapping an input tree T *to* $T[\![c_1/F_{c_1}, \ldots, c_k/F_{c_k}]\!]$

$\Rightarrow$ **top-down tree transducers with bounded lookahead**
equivalent to functional compositions of rational colorings with bounded lookahead and regular tree morphisms.

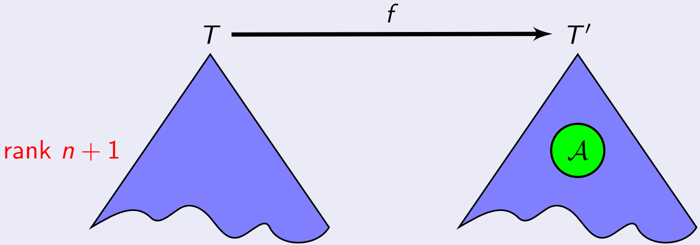
Proof idea

By induction on the rank n of the tree T (case $n = 0$ is trivial...).



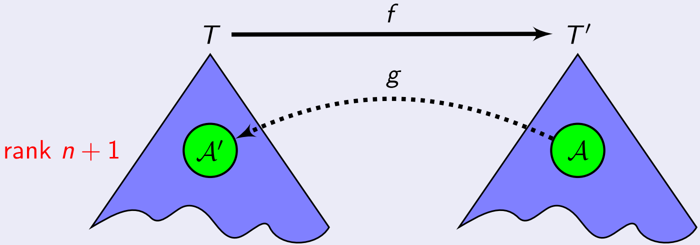
Proof idea

By induction on the rank n of the tree T (case $n = 0$ is trivial...).



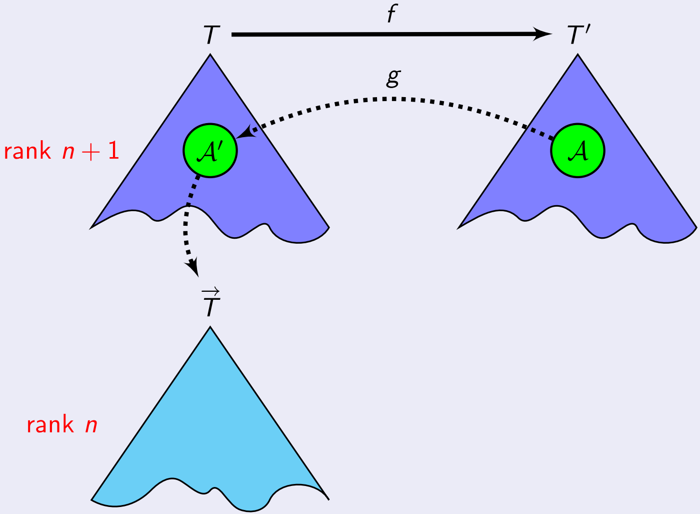
Proof idea

By induction on the rank n of the tree T (case $n = 0$ is trivial...).



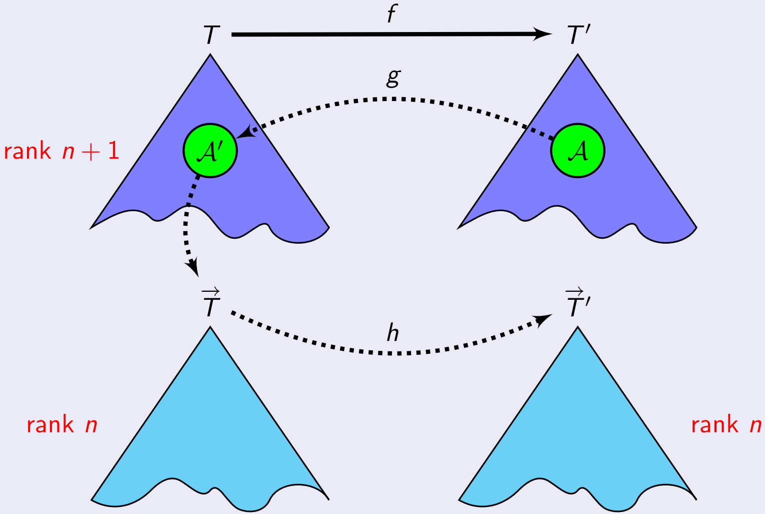
Proof idea

By induction on the rank n of the tree T (case $n = 0$ is trivial...).



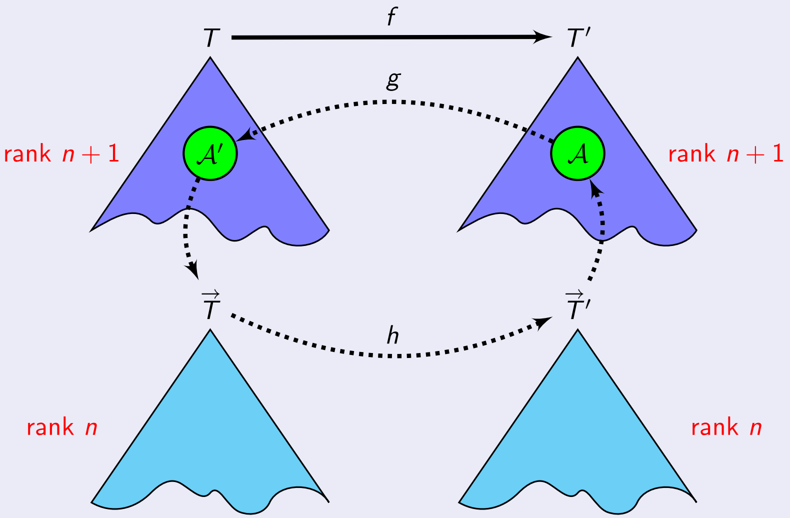
Proof idea

By induction on the rank n of the tree T (case $n = 0$ is trivial...).



Proof idea

By induction on the rank n of the tree T (case $n = 0$ is trivial...).



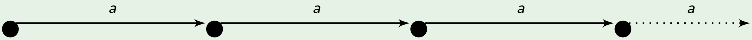
Theorem

The class of reducible trees is closed under the operation of **unfolding with backward edges and loops** $\mathcal{B}ack\mathcal{U}nfolding$.

More precisely, for every $n \in \mathbb{N}$,
if T *is a rank* n *tree, then* $\mathcal{B}ack\mathcal{U}nfolding(T)$ *is a rank* $n+1$ *tree.*

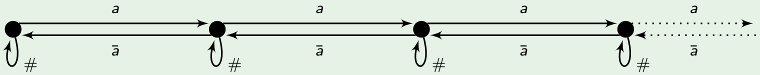
Proof by example

As a simple case, consider the **semiinfinite line** L (*a rank* 0 *tree*).
We have to show that $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a rank 1 tree.



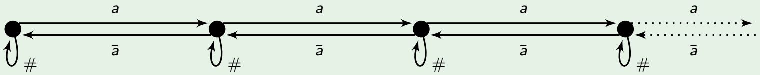
Proof by example

As a simple case, consider the **semiinfinite line** L (*a rank* 0 *tree*).
We have to show that $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a rank 1 tree.



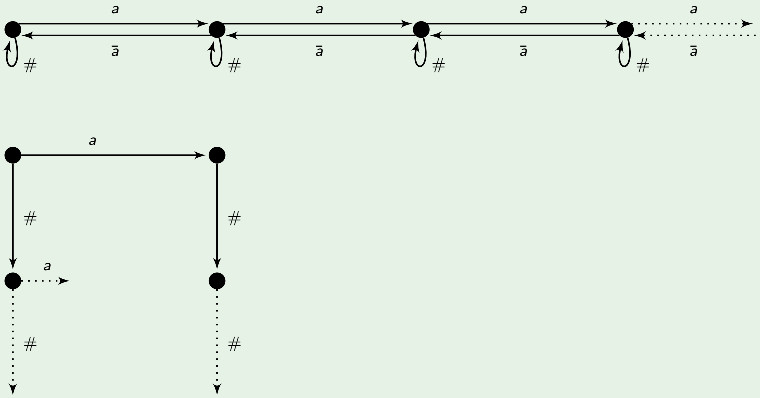
Proof by example

As a simple case, consider the **semiinfinite line** L (*a rank* 0 *tree*).
We have to show that $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a rank 1 tree.



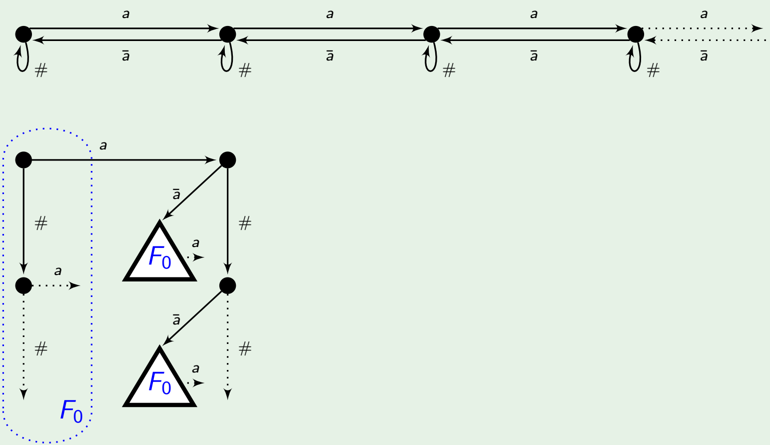
Proof by example

As a simple case, consider the **semiinfinite line** L (*a rank* 0 *tree*).
We have to show that $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a rank 1 tree.



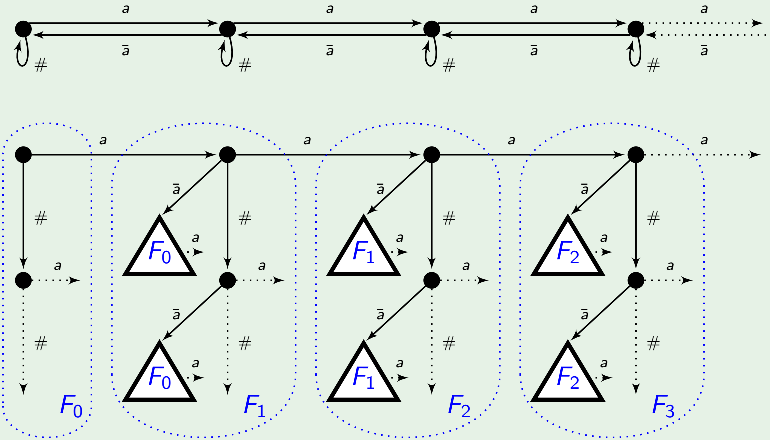
Proof by example

As a simple case, consider the **semiinfinite line** L (*a rank* 0 *tree*).
We have to show that $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a rank 1 tree.



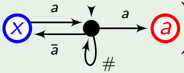
Proof by example

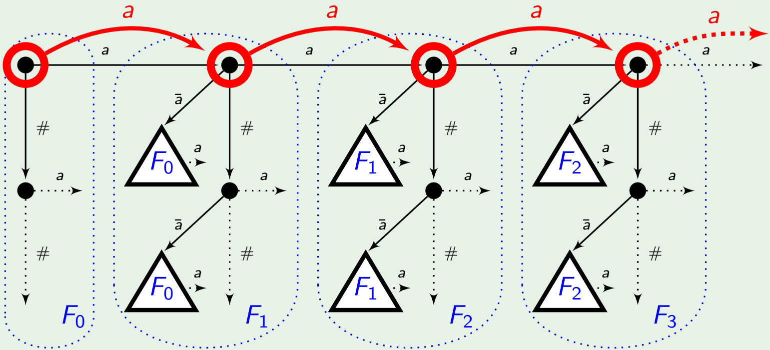
As a simple case, consider the **semiinfinite line** L (*a rank* 0 *tree*). We have to show that $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a rank 1 tree.



Proof by example

Let Π be the factorization of T such that $\mathcal{D}om(\Pi) = a^*$. Every marked factor is obtained from its predecessor via a substitution:

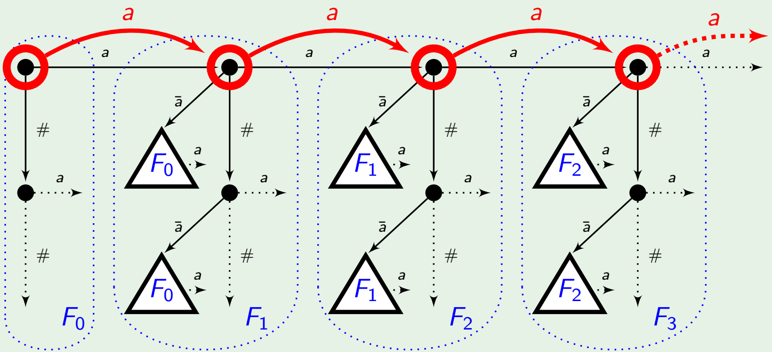
$F_{n+1} = \mathcal{U}nfolding\Big($  $\Big)[\![x/F_n]\!]$.



Proof by example

⇒ The sequence of the $\mathcal{A}$-types $t_n := [F_n]_{\mathcal{A}}$ of the marked factors can be recursively characterized as follows:

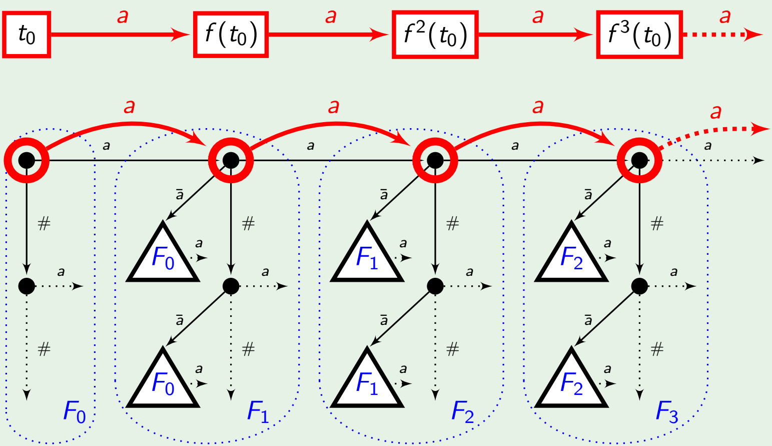
$$\begin{cases} t_0 = [F_0]_{\mathcal{A}} \\ t_{n+1} = f(t_n) \quad \text{(for a suitable function } f\text{)} \end{cases}$$



Proof by example

⇒ The $\mathcal{A}$-contraction $\vec{T}$ of $T = \mathcal{B}ack\mathcal{U}nfolding(L)$ is a *rational coloring* of L, thus a rank 0 tree.

⇒ T is a rank 1 tree.



Corollary

Reducible trees contain the deterministic trees obtained from regular trees via **unfoldings** *and* **inverse finite mappings** *(see Caucal '02).*

Proof idea

Exploit the following facts:

1. given an inverse finite mapping g^{-1} and a tree T, there is an inverse finite mapping h^{-1} that preserves bisimilarity and such that $\mathcal{U}nfolding\big(g^{-1}(T)\big) = h^{-1}\big(\mathcal{B}ack\mathcal{U}nfolding(T)\big)$ (e.g., for every label a, define $h(a) := g(a)[\varepsilon/\#]$),
2. h^{-1} can be implemented by a top-down tree transducer with bounded lookahead (see Colcombet and Löding '04),
3. reducible trees are closed under transducers with bounded lookahead and unfoldings with backward edges and loops.

Open problem / Conjecture

Generalize closure properties of reducible trees to **rational colorings with rational lookahead**.

$\Rightarrow$ This would allow us to capture all the deterministic trees in the **Caucal hierarchy**.

Other open problems:

- to establish whether the hierarchy of reducible trees is *strictly increasing* or not,
- to generalize the approach towards *colored graphs*.