Decision Problems for Origin-Close Top-Down Tree Transducers

Sarah Winter

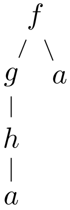
Université libre de Bruxelles (ULB), Belgium

August 26, 2021
MFCS, online

Top-Down Tree Transducers – Example I

Example.

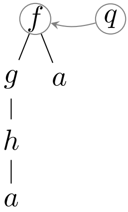
$$
\begin{array}{llcl}
R1 & q(f(x_1, x_2)) & \rightarrow & f(q(x_1), q(x_2)) \\
R2 & q(g(x_1)) & \rightarrow & q(x_1) \\
R3 & q(h(x_1)) & \rightarrow & q(h(x_1)) \\
R4 & q(a) & \rightarrow & b
\end{array}
$$



Top-Down Tree Transducers – Example I

Example.

$$\begin{array}{llcl} R1 & q(f(x_1, x_2)) & \rightarrow & f(q(x_1), q(x_2)) \\ R2 & q(g(x_1)) & \rightarrow & q(x_1) \\ R3 & q(h(x_1)) & \rightarrow & q(h(x_1)) \\ R4 & q(a) & \rightarrow & b \end{array}$$

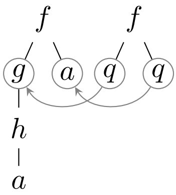


Apply $R1$

Top-Down Tree Transducers – Example I

Example.

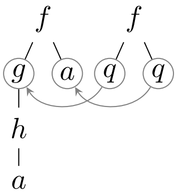
$$
\begin{array}{llcl}
R1 & q(f(x_1, x_2)) & \rightarrow & f(q(x_1), q(x_2)) \\
R2 & q(g(x_1)) & \rightarrow & q(x_1) \\
R3 & q(h(x_1)) & \rightarrow & q(h(x_1)) \\
R4 & q(a) & \rightarrow & b
\end{array}
$$



Top-Down Tree Transducers – Example I

Example.

$$\begin{array}{llcl} R1 & q(f(x_1,x_2)) & \to & f(q(x_1),q(x_2)) \\ R2 & q(g(x_1)) & \to & q(x_1) \\ R3 & q(h(x_1)) & \to & q(h(x_1)) \\ R4 & q(a) & \to & b \end{array}$$

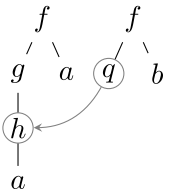


Apply $R2$ and $R4$

Top-Down Tree Transducers – Example I

Example.

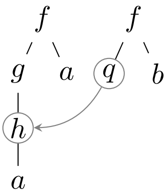
$$\begin{array}{llcl} R1 & q(f(x_1,x_2)) & \to & f(q(x_1),q(x_2)) \\ R2 & q(g(x_1)) & \to & q(x_1) \\ R3 & q(h(x_1)) & \to & q(h(x_1)) \\ R4 & q(a) & \to & b \end{array}$$



Top-Down Tree Transducers – Example I

Example.

$$
\begin{array}{llcl}
R1 & q(f(x_1, x_2)) & \to & f(q(x_1), q(x_2)) \\
R2 & q(g(x_1)) & \to & q(x_1) \\
R3 & q(h(x_1)) & \to & q(h(x_1)) \\
R4 & q(a) & \to & b
\end{array}
$$

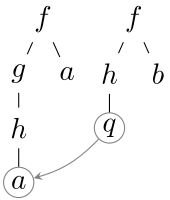


Apply $R3$

Top-Down Tree Transducers – Example I

Example.

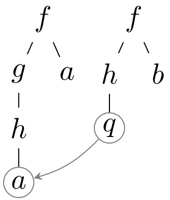
$$
\begin{array}{llcl}
R1 & q(f(x_1,x_2)) & \rightarrow & f(q(x_1),q(x_2)) \\
R2 & q(g(x_1)) & \rightarrow & q(x_1) \\
R3 & q(h(x_1)) & \rightarrow & q(h(x_1)) \\
R4 & q(a) & \rightarrow & b
\end{array}
$$



Top-Down Tree Transducers – Example I

Example.

$$\begin{array}{llcl} R1 & q(f(x_1,x_2)) & \to & f(q(x_1),q(x_2)) \\ R2 & q(g(x_1)) & \to & q(x_1) \\ R3 & q(h(x_1)) & \to & q(h(x_1)) \\ R4 & q(a) & \to & b \end{array}$$

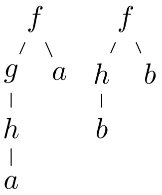


Apply $R4$

Top-Down Tree Transducers – Example I

Example.

$$
\begin{array}{llcl}
R1 & q(f(x_1, x_2)) & \rightarrow & f(q(x_1), q(x_2)) \\
R2 & q(g(x_1)) & \rightarrow & q(x_1) \\
R3 & q(h(x_1)) & \rightarrow & q(h(x_1)) \\
R4 & q(a) & \rightarrow & b
\end{array}
$$

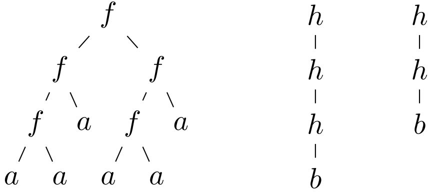


Top-Down Tree Transducers – Example II

Example.

$$
\begin{array}{lcl}
q(f(x_1,x_2)) & \to & h(q(x_1)) \\
q(f(x_1,x_2)) & \to & h(q(x_2)) \\
q(a) & \to & b
\end{array}
$$

input tree possible output trees



Motivation

Tree transducers are used in many fields

- Natural language processing
- Syntax directed translations
- Databases (XML)
- ...

Decision Problems I

Theorem ([Ési80]). The inclusion and equivalence problem are undecidable for top-down tree transducers.

Already the case for word transducers.

Decision Problems II

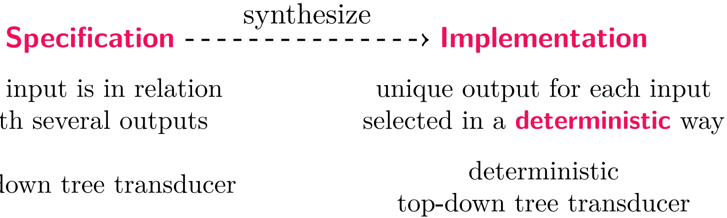
Theorem. The synthesis problem is undecidable for top-down tree transducers.

Already the case for word transducers [CL14].

Decision Problems II

Theorem. The synthesis problem is undecidable for top-down tree transducers.

Already the case for word transducers [CL14].



Decision Problems II

Theorem. The synthesis problem is undecidable for top-down tree transducers.

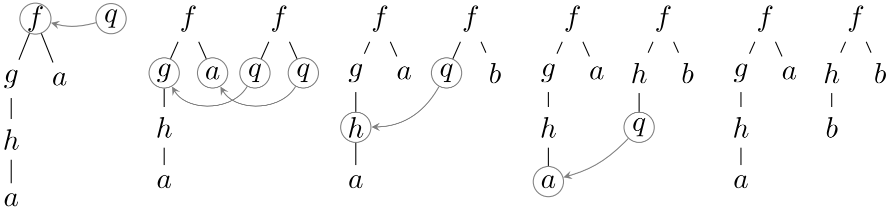
Already the case for word transducers [CL14].

Specification	--- synthesize --->	**Implementation**
one input is in relation with several outputs		unique output for each input selected in a **deterministic** way
top-down tree transducer		deterministic top-down tree transducer

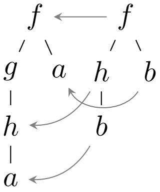
Deterministic TDTT: Rules have a unique right-hand side.

Origin Mapping I

Example. Top-down tree transducer transduction



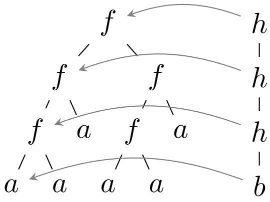
Resulting origin mapping



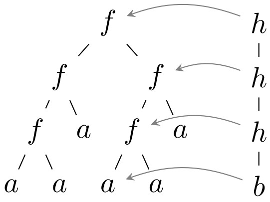
Origin Mapping II

Example.

$$
\begin{array}{lcl}
q(f(x_1, x_2)) & \to & h(q(x_1)) \\
q(f(x_1, x_2)) & \to & h(q(x_2)) \\
q(a) & \to & b
\end{array}
$$



origin mapping o_1



origin mapping o_2

Decision Problems With Origin Semantics

- TDTT $\mathcal{T}$ (with origin semantics) defines a set $R_o(\mathcal{T})$ of triples $(\underbrace{t}_{\text{input tree}}, \underbrace{s}_{\text{output tree}}, \underbrace{o : \text{dom}_s \to \text{dom}_t}_{\text{origin mapping}})$

Decision Problems With Origin Semantics

- TDTT $\mathcal{T}$ (with origin semantics) defines a set $R_o(\mathcal{T})$ of triples $(\underbrace{t}_{\text{input tree}}, \underbrace{s}_{\text{output tree}}, \underbrace{o : \mathrm{dom}_s \to \mathrm{dom}_t}_{\text{origin mapping}})$
- e.g., TDTT $\mathcal{T}_1$ is origin equivalent to TDTT $\mathcal{T}_2$ iff $R_o(\mathcal{T}_1) = R_o(\mathcal{T}_2)$

Decision Problems With Origin Semantics

- TDTT $\mathcal{T}$ (with origin semantics) defines a set $R_o(\mathcal{T})$ of triples $(\underbrace{t}_{\text{input tree}}, \underbrace{s}_{\text{output tree}}, \underbrace{o : \text{dom}_s \to \text{dom}_t}_{\text{origin mapping}})$
- e.g., TDTT $\mathcal{T}_1$ is origin equivalent to TDTT $\mathcal{T}_2$ iff $R_o(\mathcal{T}_1) = R_o(\mathcal{T}_2)$

Theorem ([FMRT18]). The origin inclusion and origin equivalence problem are decidable for top-down tree transducers.

Decision Problems With Origin Semantics

- TDTT $\mathcal{T}$ (with origin semantics) defines a set $R_o(\mathcal{T})$ of triples $(\underbrace{t}_{\text{input tree}}, \underbrace{s}_{\text{output tree}}, \underbrace{o : \text{dom}_s \to \text{dom}_t}_{\text{origin mapping}})$
- e.g., TDTT $\mathcal{T}_1$ is origin equivalent to TDTT $\mathcal{T}_2$ iff $R_o(\mathcal{T}_1) = R_o(\mathcal{T}_2)$

Theorem ([FMRT18]). The origin inclusion and origin equivalence problem are decidable for top-down tree transducers.

Also shown for other tree transducer models in [FMRT18].

Decision Problems With Origin Semantics

- TDTT $\mathcal{T}$ (with origin semantics) defines a set $R_o(\mathcal{T})$ of triples $(\underbrace{t}_{\text{input tree}}, \underbrace{s}_{\text{output tree}}, \underbrace{o : \text{dom}_s \to \text{dom}_t}_{\text{origin mapping}})$
- e.g., TDTT $\mathcal{T}_1$ is origin equivalent to TDTT $\mathcal{T}_2$ iff $R_o(\mathcal{T}_1) = R_o(\mathcal{T}_2)$

Theorem ([FMRT18]). The origin inclusion and origin equivalence problem are decidable for top-down tree transducers.

Also shown for other tree transducer models in [FMRT18].

Origin semantics is rather rigid.

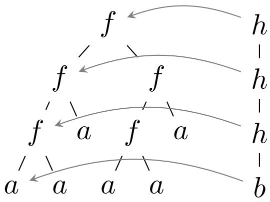
Origin Distance

Distance between two nodes Length of the shortest path.

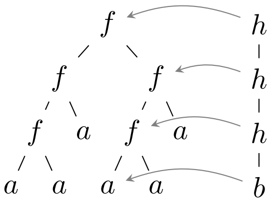
Origin Distance

Distance between two nodes Length of the shortest path.

Example. Input tree t, output tree s



origin mapping o_1

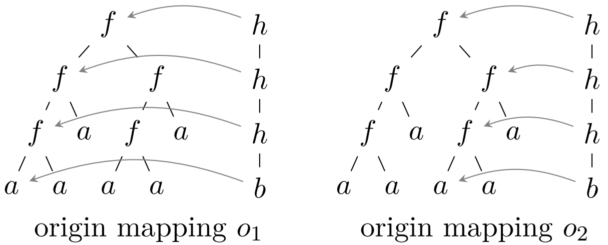


origin mapping o_2

Origin Distance

Distance between two nodes Length of the shortest path.

Example. Input tree t, output tree s



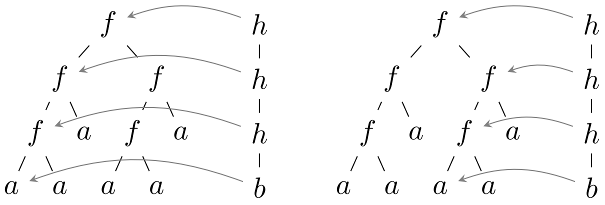
$o_1(11) = 11$ $o_2(11) = 21$ $d(11, 12) = 2$

$o_1(1111) = 1111$ $o_2(1111) = 2111$ $d(1111, 1112) = 6$

Origin Distance

Distance between two nodes Length of the shortest path.

Example. Input tree t, output tree s



origin mapping o_1 origin mapping o_2

$o_1(11) = 11$ $o_2(11) = 21$ $d(11, 12) = 2$

$o_1(1111) = 1111$ $o_2(1111) = 2111$ $d(1111, 1112) = 6$

Biggest distance between origins is 6.

$(t, s, o_1) \neq (t, s, o_2)$, but $(t, s, o_1) =_k (t, s, o_2)$ for all $k \geq 6$.

Results

- Input tree t, output tree s, origin mapping $o\colon \mathrm{dom}_s \rightarrow \mathrm{dom}_t$

Results

- Input tree t, output tree s, origin mapping $o\colon \mathrm{dom}_s \to \mathrm{dom}_t$
- Given $k \geq 0$, $(t, s, o) \in_k R_o(\mathcal{T})$ for a TDTT $\mathcal{T}$ if there is
  1. $(t, s, o') \in R_o(\mathcal{T})$, and
  2. $d(o(i), o'(i)) \leq k$ for all $i \in \mathrm{dom}_s$

Results

- Input tree t, output tree s, origin mapping $o\colon \mathrm{dom}_s \to \mathrm{dom}_t$
- Given $k \geq 0$, $(t, s, o) \in_k R_o(\mathcal{T})$ for a TDTT $\mathcal{T}$ if there is
  1. $(t, s, o') \in R_o(\mathcal{T})$, and
  2. $d(o(i), o'(i)) \leq k$ for all $i \in \mathrm{dom}_s$
- e.g., Given TDTTs $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_1$ is k-origin included in $\mathcal{T}_2$ iff $(t, s, o) \in_k R_o(\mathcal{T}_2)$ for all $(t, s, o) \in R_o(\mathcal{T}_1)$

Results

- Input tree t, output tree s, origin mapping $o\colon \mathrm{dom}_s \to \mathrm{dom}_t$
- Given $k \geq 0$, $(t, s, o) \in_k R_o(\mathcal{T})$ for a TDTT $\mathcal{T}$ if there is
  1. $(t, s, o') \in R_o(\mathcal{T})$, and
  2. $d(o(i), o'(i)) \leq k$ for all $i \in \mathrm{dom}_s$
- e.g., Given TDTTs $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_1$ is k-origin included in $\mathcal{T}_2$ iff $(t, s, o) \in_k R_o(\mathcal{T}_2)$ for all $(t, s, o) \in R_o(\mathcal{T}_1)$

Theorem. Given $k \geq 0$, the k-origin inclusion, k-origin equivalence, and k-origin synthesis problem are decidable for top-down tree transducers.

Proof Idea

Theorem. Given $k \geq 0$, the k-origin inclusion, k-origin equivalence, and k-origin synthesis problem are decidable for top-down tree transducers.

- Given $k \geq 0$ and a TDTT $\mathcal{T}$, the set $\{(t, s, o) \in_k R_o(\mathcal{T})\}$ has a representation as an ω-regular tree language.

Proof Idea

Theorem. Given $k \geq 0$, the k-origin inclusion, k-origin equivalence, and k-origin synthesis problem are decidable for top-down tree transducers.

- Given $k \geq 0$ and a TDTT $\mathcal{T}$, the set $\{(t, s, o) \in_k R_o(\mathcal{T})\}$ has a representation as an ω-regular tree language.
- Using closure properties of ω-regular tree languages, we arrive at our results.

Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

Infinite tree $\mathsf{H}_{\mathcal{T},0}$

Proof Idea – Example I

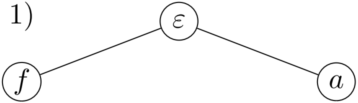
- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

Infinite tree $\mathsf{H}_{\mathcal{T},0}$

Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

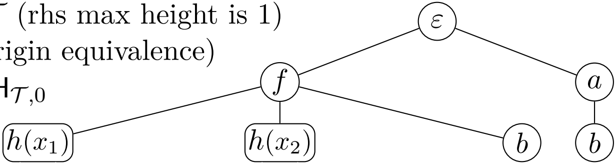
Infinite tree $\mathsf{H}_{\mathcal{T},0}$



Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

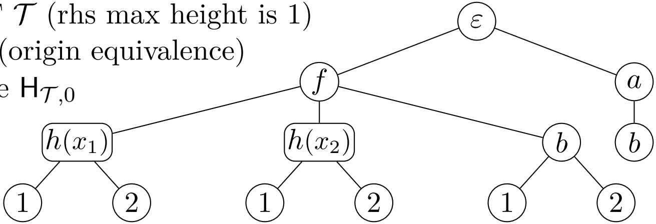
Infinite tree $\mathsf{H}_{\mathcal{T},0}$



Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

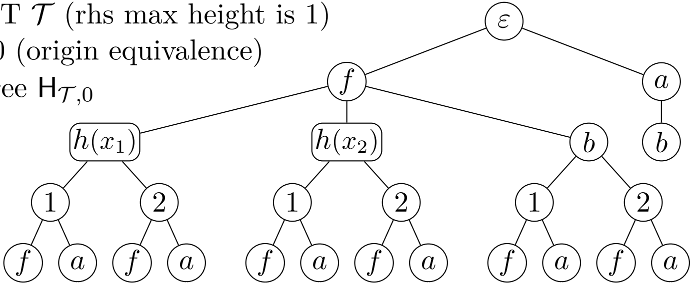
Infinite tree $\mathsf{H}_{\mathcal{T},0}$



Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

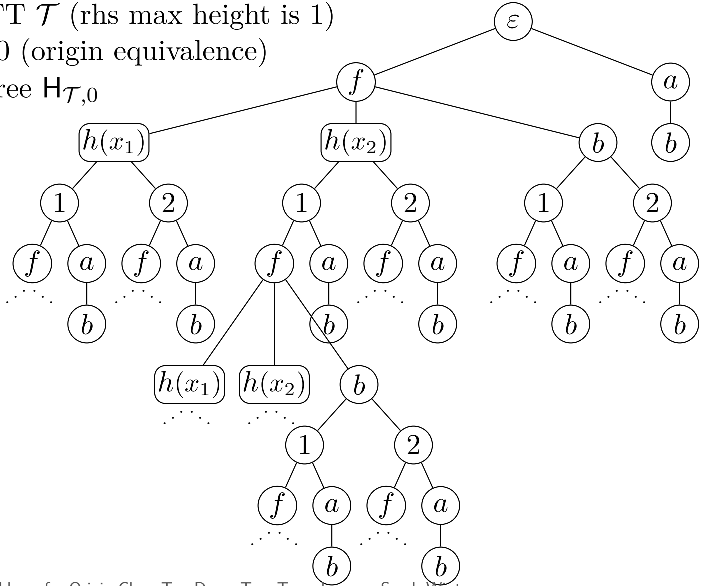
Infinite tree $\mathsf{H}_{\mathcal{T},0}$



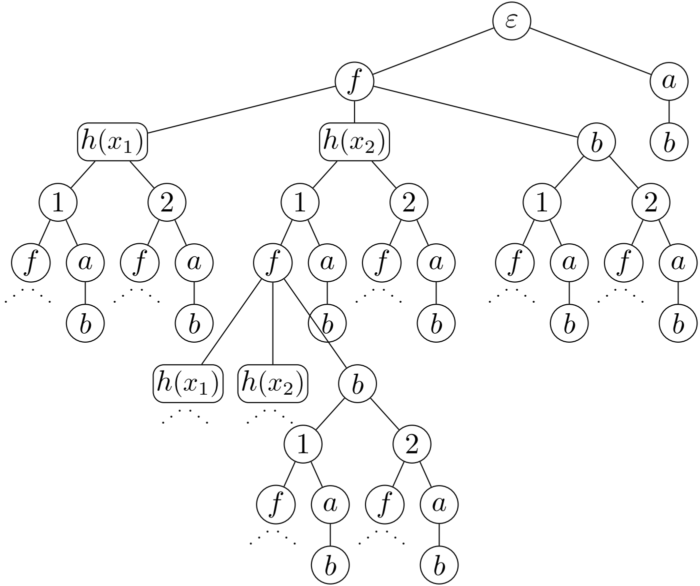
Proof Idea – Example I

- Input alphabet: $f(,), a$ Output alphabet: $h(), b$
- TDTT $\mathcal{T}$ (rhs max height is 1)
- $k = 0$ (origin equivalence)

Infinite tree $\mathsf{H}_{\mathcal{T},0}$

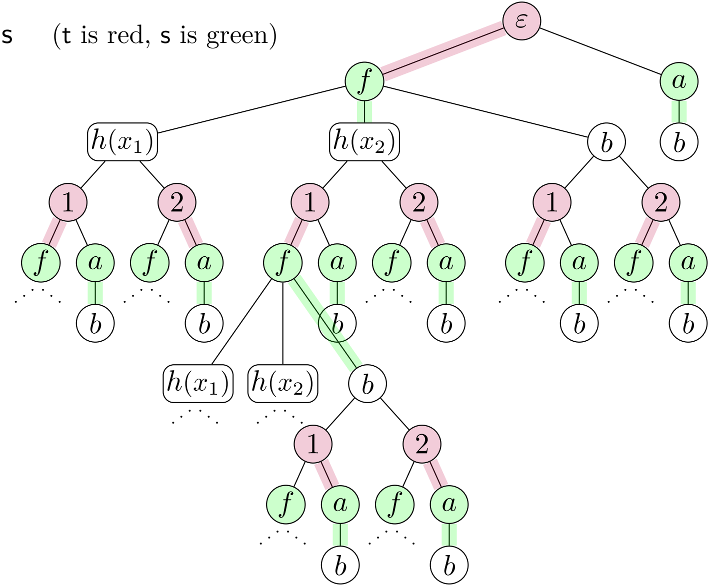


Proof Idea – Example II



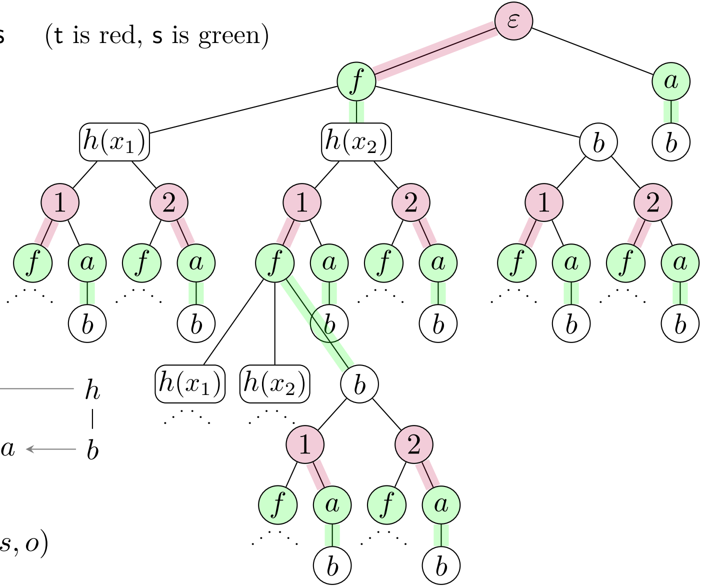
Proof Idea – Example II

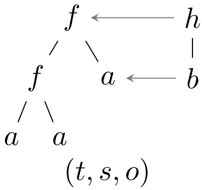
$H_{\mathcal{T},0} \frown t \frown s$ (t is red, s is green)



Proof Idea – Example II

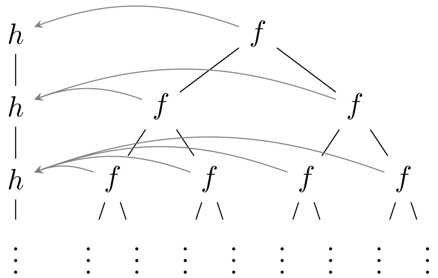
$\mathsf{H}_{\mathcal{T},0} \frown \mathsf{t} \frown \mathsf{s}$ (t is red, s is green)





Difficulty: Non-Linear Transductions

Example. $q(h(x_1)) \to f(q(x_1), q(x_1))$ and $q(a) \to b$



Number of output nodes that depend on the same input node is unbounded.

Arnaud Carayol and Christof Löding.
Uniformization in Automata Theory.
In *Proceedings of the 14th Congress of Logic, Methodology and Philosophy of Science Nancy, July 19-26, 2011*, pages 153–178. London: College Publications, 2014.

Zoltán Ésik.
Decidability results concerning tree transducers i.
Acta Cybernetica, 5(1):1–20, 1980.

Emmanuel Filiot, Sebastian Maneth, Pierre-Alain Reynier, and Jean-Marc Talbot.
Decision problems of tree transducers with origin.
Inf. Comput., 261:311–335, 2018.