Equivalence of Deterministic Tree-to-String Transducers Is Decidable

Helmut Seidl, Sebastian Maneth, Gregor Kemper

TU München, U. of Edinburgh

GT-ALGA, April 12, 2016
Overview

Part 1: The General Setting
Part 2: Tree-to-Int Transducers
Part 3: Affine Spaces
Part 4: Polynomial Invariants
Overview

Part 1: The General Setting
Part 2: Tree-to-Int Transducers
Part 3: Affine Spaces
Part 4: Polynomial Invariants
Tree-to-String Translation

Input

defs
  height
    20
  width
    50

frame
  content
    button
      Do not press!
  content
Tree-to-String Translation

Output

<frame height=20 width=50>
  <button>Do not press!</button>
  ...
</frame>

...
Tree-to-String Translation

Output

\[
<frame \text{ height}=20 \text{ width}=50>
\]
\[
<button>Do not press!</button>
\]
\[
\ldots
\]
\[
</frame>
\]

Realized by:

\[
q(\text{frame}(x_1, x_2)) \rightarrow <frame q_1(x_1)q(x_2)</frame>
\]
\[
q_1(\text{end}) \rightarrow >
\]
\[
q_1(\text{defs}(x_1, x_2)) \rightarrow q(x_1)q_1(x_2)
\]
\[
q(\text{height}(x_1)) \rightarrow \text{height} = q(x_1)
\]
\[
\ldots
\]
Tree-to-String Translation

Output

\[
\begin{align*}
\langle \text{frame height=20 width=50} \rangle \\
\langle \text{button}>\text{Do not press!}</\text{button}> \\
\langle \text{\ldots} \rangle \\
\langle \text{\ldots} \rangle \\
\end{align*}
\]

Or realized by:

\[
\begin{align*}
q'(\text{frame}(x_1, x_2)) & \rightarrow \langle \text{frame } q'_1(x_1) > q'(x_2) \langle /\text{frame} \rangle \\
q'_1(\text{end}) & \rightarrow \epsilon \\
q'_1(\text{defs}(x_1, x_2)) & \rightarrow q'(x_1)q'_1(x_2) \\
q'(\text{height}(x_1)) & \rightarrow \text{height} = q'(x_1) \\
\ldots
\end{align*}
\]
Tree-to-String Translation

These two translations are equivalent.
Tree-to-String Translation

These two translations are equivalent.

Unstructured output, though, can be generated in surprisingly different ways ...

\[ q(f(x_1, x_2, x_3)) \rightarrow q(x_3) a q_1(x_2) b q(x_2) \]
\[ q_1(f(x_1, x_2, x_3)) \rightarrow q_1(x_3) q_1(x_2) q_1(x_2) ba \]
\[ q_1(e) \rightarrow ba \]
\[ q(e) \rightarrow ab \]
Tree-to-String Translation

These two translations are equivalent.

Unstructured output, though, can be generated in surprisingly different ways ... 

\[
\begin{align*}
q(f(x_1, x_2, x_3)) & \rightarrow q(x_3) a q_1(x_2) b q(x_2) \\
q_1(f(x_1, x_2, x_3)) & \rightarrow q_1(x_3)q_1(x_2)q_1(x_2) \, ba \\
q_1(e) & \rightarrow ba \\
q(e) & \rightarrow ab
\end{align*}
\]

versus

\[
\begin{align*}
q'(f(x_1, x_2, x_3)) & \rightarrow ab \, q'(x_2)q'(x_2)q'(x_3) \\
q'(e) & \rightarrow ab
\end{align*}
\]
Related Work

problem statement

Engelfriet, 1980
Related Work (cont.)

with monadic input

Culik II, Karhumäki, 1986
Ruohonen, 1986
Honkala, 2000
Related Work (cont.)

- with monadic input  
  - Culik II, Karhumäki, 1986
  - Ruohonen, 1986
  - Honkala, 2000

- MSO-definable
  - Engelfriet, Maneth, 2006

- sequential
  - Staworko et al., 2009
Related Work (cont.)

with monadic input
Culik II, Karhumäki, 1986
Ruohonen, 1986
Honkala, 2000

MSO-definable
Engelfriet, Maneth, 2006

sequential
Staworko et al., 2009

polynomial program invariants
Letichevsky, Lvov, 1996
Müller-Olm, S., 2004
General Idea

Obvious:

In-equivalence can be verified by counter example
General Idea

**Obvious:**

**In-equivalence** can be verified by counter example

**Required:**

Complete proof system for **equivalence**
Overview

Part 1: The General Setting

Part 1: Tree-to-Int Transducers

Part 3: Affine Spaces

Part 2: Polynomial Invariants
Simplification

- A **single** transducer with states $Q = \{1, \ldots, n\}$.
- The transducer is **total**.
Simplification

- A single transducer with states $Q = \{1, \ldots, n\}$.
- The transducer is total.
- There is a topdown-deterministic automaton $B$ with states $p \in P$

$\text{dom}(p)$ is the set of trees accepted at state $p$
Simplification

- A single transducer with states $Q = \{1, \ldots, n\}$.
- The transducer is total.
- There is a topdown-deterministic automaton $B$ with states $p \in P$
  
  // generalization of algebraic data type
  
  $\text{dom}(p)$ is the set of trees accepted at state $p$
  
  // trees of type $p$
Topdown Automaton

$p_0$

frame

defs

content

button

Do not press!

content

height

width

end

height

width

end

20

50
Topdown Automaton

Frame

Content

Button

Do not press!

Def

Height 20

Width 50

End
For states $q, q'$ of the transducer, $p_0 \in P$, does it hold that

$$\llbracket q \rrbracket(t) = \llbracket q' \rrbracket(t) \quad (t \in \text{dom}(p_0))$$
From Arbitrary Output to Ints

Unary Output

\[ q(f(x_1, x_2)) \rightarrow dd \ q_1(x_1) \ d \ q_1(x_1) \ q_2(x_2) \]
From Arbitrary Output to Ints

Unary Output

\[ q(f(x_1, x_2)) \rightarrow dd\ q_1(x_1)\ d\ q_1(x_1)\ q_2(x_2) \]

Succinct representation: Tree-to-int transducer

\[ q(f(x_1, x_2)) \rightarrow 3 + 2 \cdot q_1(x_1) + q_2(x_2) \]
From Arbitrary Output to Ints

Unary Output

\[ q(f(x_1, x_2)) \rightarrow dd q_1(x_1) d q_1(x_1) q_2(x_2) \]

Succinct representation: Tree-to-int transducer

\[ q(f(x_1, x_2)) \rightarrow 3 + 2 \cdot q_1(x_1) + q_2(x_2) \]

Arbitrary Output

letters \( a, b, c, \ldots \) \( \triangleq \) digits \( 1, \ldots, h - 1 \)

string \( aabc \) \( \triangleq \) \( 1 + h \cdot (1 + h \cdot (2 + h \cdot 3)) \)
Transformation

Wanted

Transformation of tree-to-string into tree-to-int ...
Transformation

Wanted

Transformation of tree-to-string into \textbf{tree-to-int} ...

\[ q(f(x_1, x_2)) \rightarrow a \, q_1(x_1) \, b \, q_2(x_2) \]

is simulated by:

\[ q(f(x_1, x_2)) \rightarrow \]
Transformation

Wanted
Transformation of tree-to-string into tree-to-int ...

\[
q(f(x_1, x_2)) \rightarrow a \ q_1(x_1) \ b \ q_2(x_2)
\]

is simulated by:

\[
q(f(x_1, x_2), y) \rightarrow
\]

// now with an accumulating parameter \( y \)
// for right context
Transformation

Wanted

Transformation of tree-to-string into tree-to-int ...

\[ q(f(x_1, x_2)) \rightarrow a \ q_1(x_1) \ b \ q_2(x_2) \]

is simulated by:

\[ q(f(x_1, x_2), y) \rightarrow 1 + 3 \cdot \]

// now with an accumulating parameter \( y \)
// for right context
Transformation

Wanted

Transformation of tree-to-string into tree-to-int ...

\[ q(f(x_1, x_2)) \rightarrow a \ q_1(x_1) \ b \ q_2(x_2) \]

is simulated by:

\[ q(f(x_1, x_2), y) \rightarrow 1 + 3 \cdot q_1(x_1, \quad ) \]

// now with an accumulating parameter \( y \)
// for right context
Wanted
Transformation of tree-to-string into tree-to-int ...

\[ q(f(x_1, x_2)) \rightarrow a \, q_1(x_1) \, b \, q_2(x_2) \]

is simulated by:

\[ q(f(x_1, x_2), y) \rightarrow 1 + 3 \cdot q_1(x_1, 2 + 3 \cdot q_2(x_2, y)) \]

// now with an accumulating parameter \( y \)
// for right context
Overview

Part 1: The General Setting
Part 2: Tree-to-Int Transducers
Part 3: Affine Spaces
Part 4: Polynomial Ideals
Equivalence of Tree-to-Int Transducers

Idea

- The **semantics** of a tree $t$ can be seen as

$$[t] = ([1](t), \ldots, [n](t)) \in Q^n$$
Equivalence of Tree-to-Int Transducers

Idea

- The **semantics** of a tree $t$ can be seen as
  \[
  \llbracket t \rrbracket = ([1](t), \ldots, [n](t)) \in \mathbb{Q}^n
  \]

- For state $p$ of $B$, let $V_p = \{ \llbracket t \rrbracket \mid t \in \text{dom}(p) \}$.
Equivalence of Tree-to-Int Transducers

Idea

- The semantics of a tree $t$ can be seen as
  \[
  \llbracket t \rrbracket = (\llbracket 1 \rrbracket(t), \ldots, \llbracket n \rrbracket(t)) \in \mathbb{Q}^n
  \]

- For state $p$ of $B$, let $V_p = \{ \llbracket t \rrbracket \mid t \in \text{dom}(p) \}$.

- Consider $H(\mathbf{v}) = \mathbf{v}_q - \mathbf{v}_{q'}$.

- The following statements are equivalent:
  1. $q, q'$ agree on inputs from $\mathcal{L}(B)$
  2. $H(\mathbf{v}) = 0 \quad (\mathbf{v} \in V_{p_0})$
Equivalence of Tree-to-Int Transducers

Idea

- The semantics of a tree $t$ can be seen as
  \[ [t] = ([1](t), \ldots, [n](t)) \in \mathbb{Q}^n \]

- For state $p$ of $B$, let $V_p = \{ [t] \mid t \in \text{dom}(p) \}$. 

- Consider $H(v) = v_q - v_{q'}$.

- The following statements are equivalent:
  1. $q, q'$ agree on inputs from $\mathcal{L}(B)$
  2. $H(v) = 0$ \hspace{1cm} ($v \in V_{p_0}$)
  3. $H(v) = 0$ \hspace{1cm} ($v \in \text{Aff}(V_{p_0})$)
     \hspace{1cm} // affine closure
Computing Affine Closures

Define

\[
[f](x_1, \ldots, x_k) = ([T_1](x), \ldots, [T_n](x)) \quad \text{where} \quad q(f(x_1, \ldots, x_k)) \rightarrow T_q
\]
Computing Affine Closures

Define

\[
[f](\mathbf{x}_1, \ldots, \mathbf{x}_k) = ([T_1](\mathbf{x}), \ldots, [T_n](\mathbf{x})) \quad \text{where}
q(f(x_1, \ldots, x_k)) \rightarrow T_q
\]

and

\[
[3 \cdot q(x_1) + q'(x_1) + 2 \cdot q'(x_2) + 5](\mathbf{x}_1, \mathbf{x}_2) =
\]
Computing Affine Closures

Define

$$[f](x_1, \ldots, x_k) = ([T_1](x), \ldots, [T_n](x))$$

where

$$q(f(x_1, \ldots, x_k)) \rightarrow T_q$$

and

$$[3 \cdot q(x_1) + q'(x_1) + 2 \cdot q'(x_2) + 5](x_1, x_2) = 3 \cdot x_{1q} + x_{1q'} + 2 \cdot x_{2q'} + 5$$
Computing Affine Closures

Define

\[ [f](x_1, \ldots, x_k) = ([T_1](x), \ldots, [T_n](x)) \]

where

\[ q(f(x_1, \ldots, x_k)) \to T_q \]

and

\[ 3 \cdot q(x_1) + q'(x_1) + 2 \cdot q'(x_2) + 5](x_1, x_2) = 3 \cdot x_{1q} + x_{1q}' + 2 \cdot x_{2q}' + 5 \]

\[ \implies [f] : \mathbb{Q}^n \times \ldots \times \mathbb{Q}^n \to \mathbb{Q}^n \text{ is affine.} \]
Computing Affine Closures (cont.)

Consequence

\[ V_p' = \text{Aff}(V_p) \] is the least solution of:

\[ V_p' \supseteq \llbracket f \rrbracket(V_{p_1}', \ldots, V_{p_k}') \]

\(((p, f) \mapsto p_1 \ldots p_k \text{ transition of } B)\) over the complete lattice of affine sub-spaces of \( \mathbb{Q}^n \).
Computing Affine Closures (cont.)

Consequence

\( V'_p = \text{Aff}(V_p) \) is the least solution of:

\[
V'_p \supseteq \llbracket f \rrbracket(V'_{p_1}, \ldots, V'_{p_k})
\]

\(((p, f) \mapsto p_1 \ldots p_k \text{ transition of } B) \) over the complete lattice of affine sub-spaces of \( \mathbb{Q}^n \)!

Theorem

- Equivalence of total tree-to-int transducers relative to some \( B \) is decidable in polynomial time.
Computing Affine Closures (cont.)

Consequence

\[ V'_p = \operatorname{Aff}(V_p) \] is the least solution of:

\[ V'_p \supseteq \llbracket f \rrbracket(V'_{p_1}, \ldots, V'_{p_k}) \]

\((p, f) \mapsto p_1 \ldots p_k\) transition of \(B\) over the complete lattice of affine sub-spaces of \(\mathbb{Q}^n\)!

Theorem

- Equivalence of total tree-to-int transducers relative to some \(B\) is decidable in polynomial time.
- In-Equivalence of linear tree-to-string transducers is decidable in randomized polynomial time.
Overview

Part 1: The General Setting
Part 2: Tree-to-Int Transducers
Part 3: Affine Spaces
Part 4: Polynomial Invariants
Tree-to-int Transducers with Parameters

\[ q(f(x_1, x_2), y) \rightarrow q_1(x_1, q_1(x_2, 1)) \]
\[ q_1(a(x_1), y) \rightarrow y + q_1(x_1, y) \]
\[ q_1(e, y) \rightarrow 0 \]
Tree-to-int Transducers with Parameters

\[ q(f(x_1, x_2), y) \rightarrow q_1(x_1, q_1(x_2, 1)) \]
\[ q_1(a(x_1), y) \rightarrow y + q_1(x_1, y) \]
\[ q_1(e, y) \rightarrow 0 \]
\[ q'(f(x_1, x_2), y) \rightarrow q_1(x_2, q_1(x_1, 1)) \]
Tree-to-int Transducers with Parameters

\[ q(f(x_1, x_2), y) \rightarrow q_1(x_1, q_1(x_2, 1)) \]
\[ q_1(a(x_1), y) \rightarrow y + q_1(x_1, y) \]
\[ q_1(e, y) \rightarrow 0 \]
\[ q'(f(x_1, x_2), y) \rightarrow q_1(x_2, q_1(x_1, 1)) \]

The semantics of a tree \( t \) is a vector

\[ \llbracket t \rrbracket : (\mathbb{Q}^l \rightarrow \mathbb{Q})^n \]

of affine functions in the parameters

\[ \llbracket t \rrbracket_{q_i} \in \mathbb{Q}^{n \times (l+1)} \]
The Semantics of Constructors

\[
[f] : (\mathbb{Q}^{n \times (l+1)} \times \ldots \times \mathbb{Q}^{n \times (l+1)}) \rightarrow \mathbb{Q}^{n \times (l+1)}
\]

thus is of the form:

\[
([f](x_1, \ldots, x_k))_{qj} = \text{polynomial in the } x_{iq'j'}
\]
In the Example

\[ q(f(x_1, x_2), y) \rightarrow q_1(x_1, q_1(x_2, 1)) \]
\[ q_1(a(x_1), y) \rightarrow y + q_1(x_1, y) \]
\[ q_1(e, y) \rightarrow 0 \]
\[ q'(f(x_1, x_2), y) \rightarrow q_1(x_2, q_1(x_1, 1)) \]
In the Example

\[ q(f(x_1, x_2), y) \rightarrow q_1(x_1, q_1(x_2, 1)) \]
\[ q_1(a(x_1), y) \rightarrow y + q_1(x_1, y) \]
\[ q_1(e, y) \rightarrow 0 \]
\[ q'(f(x_1, x_2), y) \rightarrow q_1(x_2, q_1(x_1, 1)) \]

\[
(\llbracket f \rrbracket(x_1, x_2))_{q_0} = x_{1q_10} + x_{1q_11} \cdot (x_{2q_10} + x_{2q_11} \cdot 1) \\
(\llbracket f \rrbracket(x_1, x_2))_{q'0} = x_{2q_10} + x_{2q_11} \cdot (x_{1q_10} + x_{1q_11} \cdot 1)
\]
Polynomial Invariant

polynomial equality:

\[ z_{q1} \cdot z_{q'1} \cdot z_{q'0} - 2 \cdot z_{q0} + 3 = 0 \]
Polynomial Invariant

polynomial equality:

\[ z_{q_1} \cdot z_{q_1'} \cdot z_{q_0'} - 2 \cdot z_{q_0} + 3 = 0 \]

\[ r_1 \neq 0 \land \ldots \land r_m \neq 0 \quad \text{invariant at } p \quad \text{iff} \]

\[ r_1(\llbracket t \rrbracket) = \ldots = r_m(\llbracket t \rrbracket) = 0 \quad (t \in \text{dom}(p)) \]
Polynomial Invariant

polynomial equality:

$$z_{q_1} \cdot z_{q'1} \cdot z_{q'0} - 2 \cdot z_{q0} + 3 \equiv 0$$

$$r_1 \equiv 0 \land \ldots \land r_m \equiv 0 \quad \text{invariant at } p \quad \text{iff}$$

$$r_1([t]) = \ldots = r_m([t]) = 0 \quad (t \in \text{dom}(p))$$

can be described by polynomial ideals ...
Polynomial Ideals: A Primer

$R$ ring. $I \subseteq R$ ideal, if

- $a + b \in I$ whenever $a, b \in I$;
- $r \cdot a \in I$ whenever $a \in I$ and $r \in R$. 

$I$ is finitely generated, if

$I = \langle a_1, \ldots, a_s \rangle_R = \left\{ \sum_{i=1}^{s} r_i \cdot a_i \mid r_i \in R \right\}$

$R = \mathbb{Q}[z]$ polynomial ring
A ring \( R \) ideal, if

- \( a + b \in I \) whenever \( a, b \in I \);
- \( r \cdot a \in I \) whenever \( a \in I \) and \( r \in R \).

\( I \) is finitely generated, if

\[
I = \langle a_1, \ldots, a_s \rangle_R = \left\{ \sum_{i=1}^{s} r_i \cdot a_i \mid r_i \in R \right\}
\]
$R$ ring. $I \subseteq R$ ideal, if

- $a + b \in I$ whenever $a, b \in I$;
- $r \cdot a \in I$ whenever $a \in I$ and $r \in R$.

$I$ is finitely generated, if

$$I = \langle a_1, \ldots, a_s \rangle_R = \{ \sum_{i=1}^{s} r_i \cdot a_i \mid r_i \in R \}$$

$R = \mathbb{Q}[z]$ polynomial ring
Polynomial Ideals — Basis Theorem

David Hilbert (1890)

Every ideal of $\mathbb{Q}[z]$ is finitely generated!
Consequence

- Polynomial Invariants can be represented by polynomial ideals!
- Finite conjunctions suffice!
Consequence

- Polynomial Invariants can be represented by polynomial ideals!
- Finite conjunctions suffice!
- There are **effective** algorithms for
  - membership
  - inclusion
  - equality
Inductive Invariant

Notation: \( r_{qj}^{(f)} = ([f](x_1, \ldots, x_k))_{qj} \)

\( z \) fresh set of variables
Inductive Invariant

Notation: \( r_{a_j}^{(f)} = ([f](x_1, \ldots, x_k))_{a_j} \)

\( z \)   fresh set of variables

\( p \mapsto l_p \subseteq \mathbb{Q}[z] \)
Inductive Invariant

**Notation:** \( r_{qj}^{(f)} = ([f](x_1, \ldots, x_k))_{qj} \)

\( z \) fresh set of variables

\( p \mapsto I_p \subseteq \mathbb{Q}[z] \) is inductive if for \( p \rightarrow f(p_1, \ldots, p_k) \),

\[
I_p \subseteq \{ r \in \mathbb{Q}[z] \mid r[r^{(f)}/z] \in I_{p_1} \oplus \ldots \oplus I_{p_k} \}
\]

holds.
Inductive Invariant

Notation: \( r_{aq}^{(f)} = ([f](x_1, \ldots, x_k))_q \)

\( z \) fresh set of variables

\( p \mapsto l_p \subseteq \mathbb{Q}[z] \) is inductive if for \( p \rightarrow f(p_1, \ldots, p_k) \),

\[
    l_p \subseteq \{ r \in \mathbb{Q}[z] \mid r[r^{(f)}/z] \in \}
    \]

\[
    l_{p_1}(x_1) \oplus \ldots \oplus l_{p_k}(x_k)
    \]

holds.
Inductive Invariant

Notation: \[ r_{qj}^{(f)} = \left( \llbracket f \rrbracket(x_1, \ldots, x_k) \right)_{qj} \]

\( z \) fresh set of variables

\( p \mapsto I_p \subseteq \mathbb{Q}[z] \) is inductive if for \( p \rightarrow f(p_1, \ldots, p_k) \),

\[ I_p \subseteq \{ r \in \mathbb{Q}[z] \mid r[r^{(f)}/z] \in \langle I_{p_1}(x_1) \rangle_{\mathbb{Q}[x]} \oplus \cdots \oplus \langle I_{p_k}(x_k) \rangle_{\mathbb{Q}[x]} \} \]

holds.
Main Theorem

- Assume $p \mapsto I_p$ is inductive. Then for every $r \in I_p$,
  $$r([t]) = 0 \quad (t \in \text{dom}(p))$$
Main Theorem

• Assume $p \mapsto I_p$ is inductive. Then for every $r \in I_p$,
  \[ r([t]) = 0 \quad (t \in \text{dom}(p)) \]

• For $p \in P$, let
  \[ \bar{I}_p = \{ r \in \mathbb{Q}[z] \mid r([t]) = 0 \quad (t \in \text{dom}(p)) \} \]
  Then $p \mapsto \bar{I}_p$ is inductive.
Main Theorem

• Assume $p \mapsto I_p$ is inductive. Then for every $r \in I_p$, 
  \[ r([t]) = 0 \quad (t \in \text{dom}(p)) \]

• For $p \in P$, let 
  \[ \bar{I}_p = \{ r \in \mathbb{Q}[z] \mid r([t]) = 0 \quad (t \in \text{dom}(p)) \} \]
  Then $p \mapsto \bar{I}_p$ is inductive.

Corollary

Let $H(z) = z_{q_0} - z_{q'0}$. Then $q, q'$ are equivalent (relative to $p_0$) iff

\[ H \in I_{p_0} \]

for some inductive invariant.
Discussion

- The best inductive invariant $p \mapsto \bar{I}_p$ is a greatest fixpoint.

Greatest fixpoint iteration may not terminate.
Discussion

- The best inductive invariant $p \mapsto \overline{I}_p$ is a greatest fixpoint.
  Greatest fixpoint iteration may not terminate.

- All inductive invariants, though, can be recursively enumerated!
Discussion

- The best inductive invariant $p \mapsto \bar{I}_p$ is a greatest fixpoint. Greatest fixpoint iteration may not terminate.

- All inductive invariants, though, can be recursively enumerated!

- All potential counter examples can be enumerated ...
Wrap-up

Theorem

- Equivalence of deterministic tree-to-int transducers with parameters is decidable.
Wrap-up

Theorem

- Equivalence of deterministic tree-to-int transducers with parameters is decidable.
- Equivalence of general deterministic tree-to-string transducers is decidable.
Summary

Parameters allow to encode general output alphabets by means of unaries, i.e., numbers.
Parameters allow to encode general output alphabets by means of unaries, i.e., numbers.

Equivalence for unary transducers can be handled by means of techniques from precise program analysis, i.e., program proving.
Thank you!