Equivalence of Deterministic Tree-to-String Transducers Is Decidable

Helmut Seidl

Joint work with:

Sebastian Maneth, Gregor Kemper

PUMA, May 22, 2015

Overview

Part 1: The General Setting

Part 2: Tree-to-Int Transducers

Part 3: Affine Spaces

Part 4: Polynomial Ideals

Overview

Part 1: The General Setting

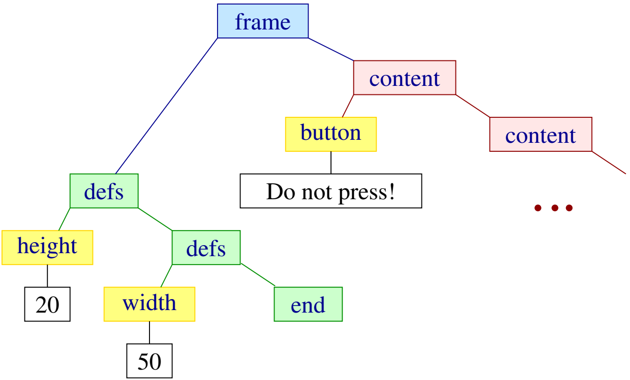
Part 2: Tree-to-Int Transducers

Part 3: Affine Spaces

Part 4: Polynomial Ideals

Tree-to-String Translation

Input



Tree-to-String Translation

Output

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

Tree-to-String Translation

Output

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

Realized by:

$$
\begin{array}{lcl}
q(\text{frame}(x_1, x_2)) & \rightarrow & \texttt{<frame } q_1(x_1)q(x_2)\texttt{</frame>} \\
q_1(\text{end}) & \rightarrow & \texttt{>} \\
q_1(\text{defs}(x_1, x_2)) & \rightarrow & q_2(x_1)q_1(x_2) \\
q_2(\text{height}(x_1)) & \rightarrow & \texttt{height = } q_3(x_1) \\
 & \ldots &
\end{array}
$$

Tree-to-String Translation (cont.)

Output

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

Or realized by:

$$
\begin{array}{lcl}
q(\text{frame}(x_1, x_2)) & \to & \texttt{<frame } q_1(x_1)\texttt{>}q(x_2)\texttt{</frame>} \\
q_1(\text{end}) & \to & \epsilon \\
q_1(\text{defs}(x_1, x_2)) & \to & q_2(x_1)q_1(x_2) \\
q_2(\text{height}(x_1)) & \to & \texttt{height = } q_3(x_1) \\
 & \dots &
\end{array}
$$

Question

Are these two translations equivalent ?

Question

Are these two translations equivalent ?

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

$$
\begin{array}{lcl}
q(f(x_1, x_2, x_3)) & \rightarrow & q(x_3)aq_1(x_2)bq(x_2) \\
q_1(f(x_1, x_2, x_3)) & \rightarrow & q_1(x_3)q_1(x_2)q_1(x_2)ba \\
q_1(e) & \rightarrow & ba \\
q(e) & \rightarrow & ab
\end{array}
$$

Question

Are these two translations equivalent ?

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

$$
\begin{array}{lcl}
q(f(x_1, x_2, x_3)) & \to & q(x_3)aq_1(x_2)bq(x_2) \\
q_1(f(x_1, x_2, x_3)) & \to & q_1(x_3)q_1(x_2)q_1(x_2)ba \\
q_1(e) & \to & ba \\
q(e) & \to & ab
\end{array}
$$

versus

$$
\begin{array}{lcl}
q'(f(x_1, x_2, x_3)) & \to & ab\, q'(x_2)q'(x_2)q'(x_3) \\
q'(e) & \to & ab
\end{array}
$$

Notation

yDT — det. topdown tree-to-string transducer

$$q(f(x_1,\ldots,x_k)) \;\to\; T$$

Notation

yDT — det. topdown tree-to-string transducer

$$q(f(x_1, \ldots, x_k)) \;\rightarrow\; T$$

yMDT — det. topdown macro tree-to-string transducer

$$q(f(x_1, \ldots, x_k), y_1, \ldots, y_l) \;\rightarrow\; T$$

// initially $y_i = \epsilon$

Related Work

problem statement	Engelfriet, 1980
MSO-definable	Engelfriet, Maneth, 2006
with monadic input	Honkala, 2000
sequential	Staworko et al., 2009

Related Work

problem statement	Engelfriet, 1980
MSO-definable	Engelfriet, Maneth, 2006
with monadic input	Honkala, 2000
sequential	Staworko et al., 2009
affine program invariants	MMO, S., 2004
polynomial program invariants	Letichevsky, Lvov, 1996 MMO, S., 2004

Overview

Part 1: The General Setting

Part 2: Tree-to-Int Transducers

Part 3: Affine Spaces

Part 4: Polynomial Ideals

From Arbitrary to Unary Output

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

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

From Arbitrary to Unary Output

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

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

Encoding

letters $a, b, c, \ldots$	$\hat{=}$	digits $1, \ldots, h-1$
string $aabc$	$\hat{=}$	$1 + h \cdot (1 + h \cdot (2 + h \cdot 3))$

From Arbitrary to Unary Output

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

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

Encoding

letters $a, b, c, \ldots$	$\hat{=}$	digits $1, \ldots, h-1$
string $aabc$	$\hat{=}$	$1 + h \cdot (1 + h \cdot (2 + h \cdot 3))$

Wanted

Transformation yDT $\Longrightarrow$ tree-to-int transducer

Transformation

The yDT rule

$$q(f(x_1, \ldots, x_k)) \;\to\; T$$

is simulated with the yMDT rule:

$$q(f(x_1, \ldots, x_k), y) \;\to\; [T]$$

where

$$\begin{array}{lcl} [\epsilon] & = & y \\ [a\, T] & = & a + h \cdot [T] \\ [q(x_i)\; T] & = & q(x_i, [T]) \end{array}$$

Overview

Part 1: The General Setting

Part 2: Tree-to-Int Transducers

Part 3: Affine Spaces

Part 4: Polynomial Ideals

Unary Transducers

Simplification

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

Unary Transducers

Simplification

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

  dom(p) is the set of trees accepted at state p.

  $\mathcal{L}(B) = \text{dom}(p_0)$ for initial state p_0.

Unary Transducers

Simplification

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

  dom(p) is the set of trees accepted at state p.

  $\mathcal{L}(B) = \text{dom}(p_0)$ for initial state p_0.

Question

For states q, q' of the transducer, does it hold that

$$[\![q]\!](t) = [\![q']\!](t) \qquad (t \in \mathcal{L}(B))$$

Unary Transducers

Idea

- The semantics of a tree t can be seen as

$$[\![t]\!] = ([\![1]\!](t), \ldots, [\![n]\!](t)) \in \mathbb{Q}^n$$

Unary 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 $V_p = \{[\![t]\!] \mid t \in \text{dom}(p)\}$.

Unary 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 $H(\mathbf{v}) = \mathbf{v}_q - \mathbf{v}_{q'}$.
- The following statements are equivalent:

  1. q, q' agree on inputs from $\mathcal{L}(B)$
  2. $H(\mathbf{v}) = 0 \qquad (\mathbf{v} \in V_{p_0})$

Unary Transducers

Idea

- The semantics of a tree t can be seen as

$$[\![t]\!] = ([\![1]\!](t), \dots, [\![n]\!](t)) \in \mathbb{Q}^n$$

- For state p of B, let $V_p = \{[\![t]\!] \mid t \in \text{dom}(p)\}$.
- 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. $H(\mathbf{v}) = 0 \qquad (\mathbf{v} \in V_{p_0})$
  3. $H(\mathbf{v}) = 0 \qquad (\mathbf{v} \in \text{Aff}(V_{p_0}))$

     // affine closure

Computing Affine Closures

Define

$$\begin{array}{rcl} [\![f]\!](\mathbf{x}_1, \ldots, \mathbf{x}_k) & = & ([\![T_1]\!](\mathbf{x}), \ldots, [\![T_n]\!](\mathbf{x})) \qquad \text{where} \\ & & q(f(x_1, \ldots, x_k)) \;\to\; T_q \end{array}$$

Computing Affine Closures

Define

$$
\begin{array}{lll}
[\![f]\!](\mathbf{x}_1,\ldots,\mathbf{x}_k) & = & ([\![T_1]\!](\mathbf{x}),\ldots,[\![T_n]\!](\mathbf{x})) \qquad \text{where} \\
 & & q(f(x_1,\ldots,x_k)) \;\to\; T_q \\
[\![z]\!](\mathbf{x}) & = & z \\
[\![j(x_i)]\!](\mathbf{x}) & = & \mathbf{x}_{ij} \\
[\![c \cdot T]\!](\mathbf{x}) & = & c \cdot [\![T]\!](\mathbf{x}) \\
[\![T_1 + T_2]\!](\mathbf{x}) & = & [\![T_1]\!](\mathbf{x}) + [\![T_2]\!](\mathbf{x})
\end{array}
$$

Computing Affine Closures

Define

$$
\begin{array}{lcl}
[\![f]\!](\mathbf{x}_1, \ldots, \mathbf{x}_k) & = & ([\![T_1]\!](\mathbf{x}), \ldots, [\![T_n]\!](\mathbf{x})) \qquad \text{where} \\
 & & q(f(x_1, \ldots, x_k)) \;\to\; T_q \\
[\![z]\!](\mathbf{x}) & = & z \\
[\![j(x_i)]\!](\mathbf{x}) & = & \mathbf{x}_{ij} \\
[\![c \cdot T]\!](\mathbf{x}) & = & c \cdot [\![T]\!](\mathbf{x}) \\
[\![T_1 + T_2]\!](\mathbf{x}) & = & [\![T_1]\!](\mathbf{x}) + [\![T_2]\!](\mathbf{x})
\end{array}
$$

Observation

$[\![f]\!] : \mathbb{Q}^n \times \ldots \times \mathbb{Q}^n \to \mathbb{Q}^n$ is affine.

Computing Affine Closures (cont.)

Consequence

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

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

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

Computing Affine Closures (cont.)

Consequence

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

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

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

Theorem

- Equivalence of total unary yDTs relative to some B is decidable in polynomial time.

Computing Affine Closures (cont.)

Consequence

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

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

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

Theorem

- Equivalence of total unary yDTs relative to some B is decidable in polynomial time.
- In-Equivalence of linear yDTs 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 Ideals

Unary Transducers with Parameters

$$
\begin{array}{lcl}
q(f(x_1, x_2), y) & \rightarrow & q_1(x_1, q_1(x_2, 1)) \\
q_1(a(x_1), y) & \rightarrow & y + q_1(x_1, y) \\
q_1(e, y) & \rightarrow & 0 \\
q'(f(x_1, x_2), y) & \rightarrow & q_1(x_2, q_1(x_1, 1))
\end{array}
$$

Unary Transducers with Parameters

$$\begin{array}{lcl} q(f(x_1,x_2),y) & \to & q_1(x_1,q_1(x_2,1)) \\ q_1(a(x_1),y) & \to & y + q_1(x_1,y) \\ q_1(e,y) & \to & 0 \\ q'(f(x_1,x_2),y) & \to & q_1(x_2,q_1(x_1,1)) \end{array}$$

The semantics of a tree t is now a vector

$$[\![t]\!] : (\mathbb{Q} {\to} \mathbb{Q})^n$$

of affine functions in the parameters.

Unary Transducers with Parameters

$$
\begin{array}{lcl}
q(f(x_1, x_2), y) & \to & q_1(x_1, q_1(x_2, 1)) \\
q_1(a(x_1), y) & \to & y + q_1(x_1, y) \\
q_1(e, y) & \to & 0 \\
q'(f(x_1, x_2), y) & \to & q_1(x_2, q_1(x_1, 1))
\end{array}
$$

The semantics of a tree t is now a vector

$$[\![t]\!] : (\mathbb{Q} \to \mathbb{Q})^n$$

of affine functions in the parameters.

$\Longrightarrow$

It can be represented by a matrix $([\![t]\!]_{qi})$.

The Semantics of Constructors

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

thus is of the form:

$$([\![f]\!](\mathbf{x}))_{qj} = \text{polynomial in the } \mathbf{x}_{iq'j'}$$

The Semantics of Constructors

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

thus is of the form:

$$([\![f]\!](\mathbf{x}))_{qj} = \text{polynomial in the } \mathbf{x}_{iq'j'}$$

$\Longrightarrow$ The affine closure trick fails :-(

In the Example

$$\begin{array}{lcl} q(f(x_1, x_2), y) & \rightarrow & q_1(x_1, q_1(x_2, 1)) \\ q_1(a(x_1), y) & \rightarrow & y + q_1(x_1, y) \\ q_1(e, y) & \rightarrow & 0 \\ q'(f(x_1, x_2), y) & \rightarrow & q_1(x_2, q_1(x_1, 1)) \end{array}$$

In the Example

$$\begin{array}{lcl} q(f(x_1, x_2), y) & \to & q_1(x_1, q_1(x_2, 1)) \\ q_1(a(x_1), y) & \to & y + q_1(x_1, y) \\ q_1(e, y) & \to & 0 \\ q'(f(x_1, x_2), y) & \to & q_1(x_2, q_1(x_1, 1)) \end{array}$$

$$\begin{array}{lcl} ([\![f]\!](\mathbf{x}))_{q0} & = & \mathbf{x}_{1q_1 0} + \mathbf{x}_{1q_1 1} \cdot (\mathbf{x}_{2q_1 0} + \mathbf{x}_{2q_1 1} \cdot 1) \\ ([\![f]\!](\mathbf{x}))_{q'0} & = & \mathbf{x}_{2q_1 0} + \mathbf{x}_{2q_1 1} \cdot (\mathbf{x}_{1q_1 0} + \mathbf{x}_{1q_1 1} \cdot 1) \end{array}$$

In the Example

$$\begin{array}{lcl} q(f(x_1, x_2), y) & \to & q_1(x_1, q_1(x_2, 1)) \\ q_1(a(x_1), y) & \to & y + q_1(x_1, y) \\ q_1(e, y) & \to & 0 \\ q'(f(x_1, x_2), y) & \to & q_1(x_2, q_1(x_1, 1)) \end{array}$$

$$\begin{array}{lcl} ([\![f]\!](\mathbf{x}))_{q0} & = & \mathbf{x}_{1q_10} + \mathbf{x}_{1q_11} \cdot (\mathbf{x}_{2q_10} + \mathbf{x}_{2q_11} \cdot 1) \\ ([\![f]\!](\mathbf{x}))_{q'0} & = & \mathbf{x}_{2q_10} + \mathbf{x}_{2q_11} \cdot (\mathbf{x}_{1q_10} + \mathbf{x}_{1q_11} \cdot 1) \end{array}$$

$$([\![f]\!](\mathbf{x}))_{q1} \quad = \quad 0 \quad = \quad ([\![f]\!](\mathbf{x}))_{q'1}$$

In the Example

$$\begin{array}{lcl} q(f(x_1,x_2),y) & \to & q_1(x_1,q_1(x_2,1)) \\ q_1(a(x_1),y) & \to & y + q_1(x_1,y) \\ q_1(e,y) & \to & 0 \\ q'(f(x_1,x_2),y) & \to & q_1(x_2,q_1(x_1,1)) \end{array}$$

$$\begin{array}{lcl} (\llbracket f \rrbracket(\mathbf{x}))_{q0} & = & \mathbf{x}_{1q_10} + \mathbf{x}_{1q_11} \cdot (\mathbf{x}_{2q_10} + \mathbf{x}_{2q_11} \cdot 1) \\ (\llbracket f \rrbracket(\mathbf{x}))_{q'0} & = & \mathbf{x}_{2q_10} + \mathbf{x}_{2q_11} \cdot (\mathbf{x}_{1q_10} + \mathbf{x}_{1q_11} \cdot 1) \\ (\llbracket f \rrbracket(\mathbf{x}))_{q1} & = & 0 \quad = \quad (\llbracket f \rrbracket(\mathbf{x}))_{q'1} \\ (\llbracket a \rrbracket(\mathbf{x}))_{q_10} & = & \mathbf{x}_{1q_10} \\ (\llbracket a \rrbracket(\mathbf{x}))_{q_11} & = & 1 + \mathbf{x}_{1q_11} \\ (\llbracket e \rrbracket)_{q_10} & = & 0 \quad = \quad (\llbracket e \rrbracket)_{q_11} \end{array}$$

Polynomial Invariants

Idea

- polynomial equality:

$$\mathbf{z}_{q1} \cdot \mathbf{z}_{q'1} \cdot \mathbf{z}_{q'0} - 2 \cdot \mathbf{z}_{q0} + 3 \doteq 0$$

Polynomial Invariants

Idea

- polynomial equality:

$$\mathbf{z}_{q1} \cdot \mathbf{z}_{q'1} \cdot \mathbf{z}_{q'0} - 2 \cdot \mathbf{z}_{q0} + 3 \doteq 0$$

- $q_1 \doteq 0 \wedge \ldots \wedge q_r \doteq 0$ invariant at p iff

$$q_1([\![t]\!]) = \ldots = q_r([\![t]\!]) = 0 \qquad (t \in \mathbf{dom}(p))$$

Polynomial Invariants

Idea

- polynomial equality:

$$\mathbf{z}_{q1} \cdot \mathbf{z}_{q'1} \cdot \mathbf{z}_{q'0} - 2 \cdot \mathbf{z}_{q0} + 3 \doteq 0$$

- $q_1 \doteq 0 \wedge \ldots \wedge q_r \doteq 0$ invariant at p iff

$$q_1([\![t]\!]) = \ldots = q_r([\![t]\!]) = 0 \qquad (t \in \mathbf{dom}(p))$$

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 $r \in R$.

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

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

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

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

$R = \mathbb{Q}[\mathbf{z}]$ polynomial ring

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

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

$R = \mathbb{Q}[\mathbf{z}]$ polynomial ring

Hilbert (1890):

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

Consequences

Vanishing Ideal:

$$\mathcal{I}(V) = \{q \in \mathbb{Q}[\mathbf{z}] \mid \forall\, \mathbf{v} \in V.\, q(\mathbf{v}) = 0\}$$

Consequences

Vanishing Ideal:

$$\mathcal{I}(V) = \{q \in \mathbb{Q}[\mathbf{z}] \mid \forall\, \mathbf{v} \in V.\, q(\mathbf{v}) = 0\}$$

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

Consequences

Vanishing Ideal:

$$\mathcal{I}(V) = \{q \in \mathbb{Q}[\mathbf{z}] \mid \forall\, \mathbf{v} \in V.\, q(\mathbf{v}) = 0\}$$

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

Inductive Invariants

Notation: $q_{qj}^{(f)} = (\llbracket f \rrbracket(\mathbf{x}))_{qj}$

$\mathbf{z}$ fresh set of variables

Inductive Invariants

Notation: $q^{(f)}_{qj} = (\llbracket f \rrbracket(\mathbf{x}))_{qj}$

$\mathbf{z}$ fresh set of variables

$p \mapsto I_p$ is inductive if for every transition $(p, f) \mapsto p_1 \ldots p_k$ of the automaton B,

$$I_p \subseteq \{q \in \mathbb{Q}[\mathbf{z}] \mid q[q^{(f)}/\mathbf{z}] \in \langle I_{p_1}(\mathbf{x}_1)\rangle_{\mathbb{Q}[\mathbf{x}]} \oplus \ldots \oplus \langle I_{p_k}(\mathbf{x}_k)\rangle_{\mathbb{Q}[\mathbf{x}]}\}$$

holds.

Main Result

Theorem

- Assume $p \mapsto I_p$ is inductive. Then for every $q \in I_p$,

$$q([\![t]\!]) = 0 \qquad (t \in \mathrm{dom}(p))$$

Main Result

Theorem

- Assume $p \mapsto I_p$ is inductive. Then for every $q \in I_p$,

$$q([\![t]\!]) = 0 \qquad (t \in \text{dom}(p))$$

- For $\bar{I}_p = \mathcal{I}(\{[\![t]\!] \mid t \in \text{dom}(p)\})$, $p \mapsto \bar{I}_p$ is inductive.

Main Result

Theorem

- Assume $p \mapsto I_p$ is inductive. Then for every $q \in I_p$,

$$q([\![t]\!]) = 0 \qquad (t \in \mathrm{dom}(p))$$

- For $\overline{I}_p = \mathcal{I}(\{[\![t]\!] \mid t \in \mathrm{dom}(p)\})$, $p \mapsto \overline{I}_p$ is inductive.

Corollary

Let $H(\mathbf{z}) = \mathbf{z}_{q0} - \mathbf{z}_{q'0}$. Then q, q' are equivalent (relative to $\mathcal{L}(B)$) iff

$$H \in \overline{I}_{p_0}$$

Discussion

- Inductive $p \mapsto I_p$ with $H \in I_{p_0}$ proves that H holds.

Discussion

- Inductive $p \mapsto I_p$ with $H \in I_{p_0}$ proves that H holds.
- If H holds, it can be proven (somehow).

Discussion

- Inductive $p \mapsto I_p$ with $H \in I_{p_0}$ proves that H holds.
- If H holds, it can be proven (somehow).
- $p \mapsto \bar{I}_p$ is a greatest fixpoint.

  Greatest fixpoint iteration may not terminate ...

Discussion

- Inductive $p \mapsto I_p$ with $H \in I_{p_0}$ proves that H holds.
- If H holds, it can be proven (somehow).
- $p \mapsto \bar{I}_p$ is a greatest fixpoint.

  Greatest fixpoint iteration may not terminate ...

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

Discussion

- Inductive $p \mapsto I_p$ with $H \in I_{p_0}$ proves that H holds.
- If H holds, it can be proven (somehow).
- $p \mapsto \bar{I}_p$ is a greatest fixpoint.

  Greatest fixpoint iteration may not terminate ...

- All inductive invariants, though, can be recursively enumerated!
- All potential counter examples can also be enumerated!!

Wrap-up

Theorem

- Equivalence of unary yMDTs is decidable.

Wrap-up

Theorem

- Equivalence of unary yMDTs is decidable.
- Equvalence of general yDTs is decidable.

Summary

Parameters allow to encode general output alphabets by means of unaries.

Summary

Parameters allow to encode general output alphabets by means of unaries.

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

Summary

Parameters allow to encode general output alphabets by means of unaries.

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

Unary	General	Equalities
yDT		affine

Summary

Parameters allow to encode general output alphabets by means of unaries.

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

Unary	General	Equalities
yDT		affine
nsn yMDT	linear yDT	affine

Summary

Parameters allow to encode general output alphabets by means of unaries.

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

Unary	General	Equalities
yDT		affine
nsn yMDT	linear yDT	affine
yMDT	yDT	polynomial

Thank you!