The Equivalence Problem for Deterministic MSO Tree Transducers is Decidable

Sebastian Maneth
National ICT Australia Ltd

Joint Work with Joost Engelfriet

The imagination driving Australia's ICT future.

Problem Description

Macro Tree Transducers (MTTs) are functional programs
that work on trees and have pattern matching and tree construction as only operations. (→ they always terminate)

EXAMPLE

For a tree s = a(b, c), preorder(s, ⊥) is the tree a(b (c (⊥)))

```
MTT preorder( s : Tree, t: Tree ) : tree
{
  match s {
    a(x1, x2 ) ->  a(preorder(x1, preorder(x2, t ))
    b          ->  b(t)
    c          ->  b(t)
}
```

The imagination driving Australia's ICT future.

Problem Description

Macro Tree Transducers (MTTs) are functional programs
that work on trees and have pattern matching and tree construction as only operations. (→ they always terminate)

EXAMPLE

For a tree s = a(b, c), preorder(s, ⊥) is the tree a(b (c (⊥)))

```
MTT preorder( s : InTree, t: OutTree ) : OutTree
{
  match s {
    a(x1, x2) -> a(preorder(x1, preorder(x2, t ))
    b         -> b(t)
    c         -> b(t)
}
```

The imagination driving Australia's ICT future.

Problem Description

Macro Tree Transducers (MTTs) are functional programs
that work on trees and have pattern matching and tree construction as only operations. (→ they always terminate)

MTTs:

• well studied since the 80's in Formal Language Theory
they are "finite-state top-down tree transducers with parameters":

(Q, Σ, Δ, q0, R)

• can model *XML query and transformation languages* such as XQuery and XSLT [Milo/Suciu/Vianu02] + [Engelfriet/M.03]:

Pebble Tr. Tr's ⊆ MTT^*

The imagination driving Australia's ICT future.

Problem Description

Equivalence Problem

Given MTTs M1 and M2, ∀s: M1(s) = M2(s)?

Is this problem decidable??

Unfortunately, this remains open, BUT:

MAIN THEOREM
The Equivalence Problem for finite-copying MTTs is decidable.

The imagination driving Australia's ICT future.

Problem Description

Equivalence Problem

Given MTTs M1 and M2, ∀s: M1(s) = M2(s)?

Is this problem decidable??

Unfortunately, this remains open, BUT:

MAIN THEOREM
The Equivalence Problem for finite-copying MTTs is decidable.

Beautiful class = single-use Attribute Grammars
= tree translations definable in MSO Logic (MSOTT)

[Bloem/Engelfriet02] + [Engelfriet/M.99]

Outline

- Equivalence Problems in FLT
- Parikh's Theorem and Semilinear Sets
- MSO Tree Transducers (MSOTT)
- Deciding Equivalence of MSOTTs

Equivalence Problems in FLT

- *nondeterministic (one-way) finite state transducers* **undecidable** [Griffiths68]
  (→ reduction from PCP, use complement and union)

Equivalence Problems in FLT

- *nondeterministic (one-way) finite state transducers* **undecidable** [Griffiths68]
  (→ reduction from PCP, use complement and union)
- *deterministic (one-way) finite state transducers* **decidable** [Gurari82]
  (→ use Parikh property)

Equivalence Problems in FLT

- *nondeterministic (one-way) finite state transducers* **undecidable** [Griffiths68]
  (→ reduction from PCP, use complement and union)
- *deterministic (one-way) finite state transducers* **decidable** [Gurari82]
  (→ use Parikh property)
- *deterministic top-down tree transducers* **decidable** [Esik80]

Equivalence Problems in FLT

- *nondeterministic (one-way) finite state transducers* **undecidable** [Griffiths68]
  (→ reduction from PCP, use complement and union)
- *deterministic (one-way) finite state transducers* **decidable** [Gurari82]
  (→ use Parikh property)
- *deterministic top-down tree transducers* **decidable** [Esik80]
- *nonnested, seperated attributed/marco tree transducers* **decidable** [Courcelle/Franchi-Zannetacci82]

(→ seperated = can be evaluated in two phases,
(1) only inherited, over Δ_{inh}
(2) synthsized, over Δ_{syn})

Parikh's Theorem and Semilinear Sets

Let $\Sigma = \{ \sigma_1, \ldots, \sigma_k \}$.
For a discrete graph / string / tree g over Σ, define

$$\text{Par}(g) = (n_1, \ldots, n_k) \subseteq \mathbf{N}^k$$

such that n_i is the number of σ_i's in g.

A set $P \subseteq \mathbf{N}^k$ is semilinear if there exists a *regular langage* R such that P = Par(R).

A set S (of graphs/stings) is **Parikh**, if Par(S) is semilinear.

Parikh's Theorem and Semilinear Sets

Let $\Sigma = \{ \sigma_1, \ldots, \sigma_k \}$.
For a discrete graph / string / tree g over Σ, define

Par(g) = $(n_1, \ldots, n_k) \subseteq \mathbf{N}^k$

such that n_i is the number of σ_i's in g.

A set $P \subseteq \mathbf{N}^k$ is semilinear if there exists a *regular langage* R such that P = Par(R).

A set S (of graphs/stings) is **Parikh**, if Par(S) is semilinear.

PARIKH's THEOREM
Every context-free language is Parikh.

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Context-free languages are **Parikh**

L1 = $\{ a^n b^n \mid n \geq 0 \}$

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Context-free languages are **Parikh**

L1 = $\{ a^n b^n \mid n \geq 0 \}$

Par(L1) = $\{ (n, n) \mid n \geq 0 \}$

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Context-free languages are **Parikh**

L1 = $\{ a^n b^n \mid n \geq 0 \}$

Par(L1) = $\{ (n, n) \mid n \geq 0 \}$ = Par((ab)*)

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Context-free languages are **Parikh**

L1 = $\{ a^n b^n \mid n \geq 0 \}$

Par(L1) = $\{ (n, n) \mid n \geq 0 \}$ = Par((ab)*)

L2 = $\{ ww \mid w \in \{ a, b \}^* \}$

Par(L2) =

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Context-free languages are **Parikh**

L1 = $\{ a^n b^n \mid n \geq 0 \}$

Par(L1) = $\{ (n, n) \mid n \geq 0 \}$ = Par((ab)*)

L2 = $\{ ww \mid w \in \{ a, b \}^* \}$

Par(L2) = Par((aa)*(bb)*)

L3 = $\{ w \in \{ a, b \}^* \mid \#a(w) = \#b(w) \}$

→ Is L3 context-free?? SURE! (even deterministic)

And, again, Par(L3) = Par(L1)

Parikh's Theorem and Semilinear Sets

Lemma It is decidable for a semilinear set $S \subseteq N^2$,
whether there exists $n \in N$ such that $(n, n) \in S$.

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Lemma It is decidable for a semilinear set $S \subseteq N^2$,
whether there exists $n \in N$ such that $(n, n) \in S$.

Proof. Let $P = \{ (n, n) \mid n \in N \} = Par((ab)^*)$

Then $S \cap P$ is semilinear [GinsburgSpanier64, difficult!],
and semilinear sets have decidable emptiness.

Alternative Proof??

The imagination driving Australia's ICT future.

Parikh's Theorem and Semilinear Sets

Lemma It is decidable for a semilinear set $S \subseteq N^2$,
whether there exists $n \in N$ such that $(n, n) \in S$.

Proof. Let $P = \{ (n, n) \mid n \in N \} = Par((ab)^*)$

Then $S \cap P$ is semilinear [GinsburgSpanier64, difficult!],
and semilinear sets have decidable emptiness.

Alternative Proof??

$L3 = \{ w \in \{a, b\}^* \mid \#a(w) = \#b(w) \}$

Let R be reg. language s.t. Par(S) = R.

Then $R \cap L3$ is context-free ("triple-construction")
and cf. languages have decidable emptiness.

The imagination driving Australia's ICT future.

MSO Tree Transducers

FACTS

MSOTT

→ Closed under composition
→ inverses preserve REGT

MSOTT(REGT)
= MSOTT(REGT)

→ Closed under MSOTT
→ path languages(MSOTT(REGT))
= string languages(MSOTT(REGT))
= MSOTS(REGT)

→ are Parikh

The imagination driving Australia's ICT future.

MSO Tree Transducers

FACTS

MSOTT

➔ Closed under composition ←
→ inverses preserve REGT

MSOTT(REGT)
= MSOTT(REGT)

➔ Closed under MSOTT ←
→ path languages(MSOTT(REGT))
= string languages(MSOTT(REGT))
= MSOTS(REGT)

➔ are Parikh ←

The imagination driving Australia's ICT future.

Deciding Equivalence of MSOTTs

Let's first do string output.
The Equivalence problem for MSOTS is decidable.

Deciding Equivalence of MSOTTs

Let's first do string output.
The Equivalence problem for MSOTS is decidable.

Given MSOTS M1 and M2:
Add an end marker \$ (gives MSOTS N1/N2)
N1(s) = M1(s) \$
N2(s) = M2(s) \$

M1 *is equiv. to* M2

iff $\neg(\exists s \exists n \;\; N1(s)/n \neq N2(s)/n)$

The imagination driving Australia's ICT future.

Deciding Equivalence of MSOTTs

Let's first do string output.
The Equivalence problem for MSOTS is decidable.

Given MSOTS M1 and M2:
Add an end marker \$ (gives MSOTS N1/N2)
N1(s) = M1(s) \$
N2(s) = M2(s) \$

M1 *is equiv. to* M2

iff $\neg(\exists s \exists n \;\; N1(s)/n \neq N2(s)/n)$

iff $\neg(\exists a, b\colon a \neq b \;\wedge\; \exists s \exists n \; N1(s)/n = a \wedge N2(s)/n = b)$

The imagination driving Australia's ICT future.

Deciding Equivalence of MSOTTs

Let's first do string output.
The Equivalence problem for MSOTS is decidable.

Given MSOTS M1 and M2:
Add an end marker \$ (gives MSOTS N1/N2)
N1(s) = M1(s) \$
N2(s) = M2(s) \$

M1 *is equiv. to* M2 (gives $N1^a(s) = \{ a^n \mid N1(s)/n = a \}$ and $N2^b$)

iff $\neg(\exists s \exists n \;\; N1(s)/n \neq N2(s)/n)$

iff $\neg(\exists a, b\colon a \neq b \;\wedge\; \exists s \exists n \; N1(s)/n = a \wedge N2(s)/n = b)$

iff $\neg($ .... $\wedge \;\exists s \exists n \;\; a^n \in N1^a(s) \;\wedge\; b^n \in N2^b(s))$

The imagination driving Australia's ICT future.

Deciding Equivalence of MSOTTs

Let's first do string output.
The Equivalence problem for MSOTS is decidable.

Given MSOTS M1 and M2:
Add an end marker \$ (gives MSOTS N1/N2)
N1(s) = M1(s) \$
N2(s) = M2(s) \$

M1 *is equiv. to* M2 (gives $N12^{a,b}(s) = \{ a^n b^m \mid N1(s)/n = a, N2(s)/m = b \}$)

iff $\neg(\exists s \exists n \;\; N1(s)/n \neq N2(s)/n)$

iff $\neg(\exists a, b\colon a \neq b \;\wedge\; \exists s \exists n \; N1(s)/n = a \wedge N2(s)/n = b)$

iff $\neg($ .... $\wedge \;\exists s \exists n \;\; a^n \in N1^a(s) \;\wedge\; b^n \in N2^b(s))$

iff $\neg($ .... $\wedge \; Par(Out(N12^{a,b})) \cap \{(n,n) \mid n \geq 1\} \neq \varnothing$ **?**)

The imagination driving Australia's ICT future.

Deciding Equivalence of MSOTTs

Let's first do string output.
The Equivalence problem for MSOTS is decidable.

Given MSOTS M1 and M2:
Add an end marker \$ (gives MSOTS N1/N2)
N1(s) = M1(s) \$
N2(s) = M2(s) \$

M1 *is equiv. to* M2 (gives $N12^{a,b}(s) = \{ a^n b^m \mid N1(s)/n = a, N2(s)/m = b \}$)

iff $\neg(\exists s \exists n \;\; N1(s)/n \neq N2(s)/n)$

iff $\neg(\exists a, b\colon a \neq b \;\wedge\; \exists s \exists n \; N1(s)/n = a \wedge N2(s)/n = b)$

iff $\neg($ .... $\wedge \;\exists s \exists n \;\; a^n \in N1^a(s) \;\wedge\; b^n \in N2^b(s))$

iff $\neg($ .... $\wedge \; Par(Out(N12^{a,b})) \cap \{(n,n) \mid n \geq 1\} \neq \varnothing$ **?**) Decidable! By Lemma.

□

The imagination driving Australia's ICT future.

Deciding Equivalence of MSOTTs

The Equivalence problem for MSOTT is decidable.

Simply compose M1 and M2 w. the MSOTS `preorder`!
→ Gives MSOTS N1, N2.

□

COROLLARY (using [Maneth, FSTTCS'02])

(structural) XML queries of linear size increase have decidable equivalence.

THE END ☺

The Equivalence problem for MSOTT is decidable.

Simply compose M1 and M2 w. the MSOTS `preorder`!
→ Gives MSOTS N1, N2.
□

COROLLARY (using [Maneth, FSTTCS'02])

(structural) XML queries of linear size increase have decidable equivalence.