On uniformisation in MSO over infinite trees

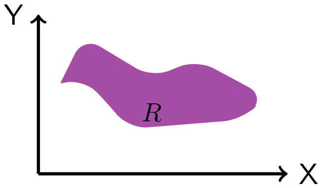
Michał Skrzypczak

LIAFA, University of Warsaw

GT-ALGA 2015
Paris

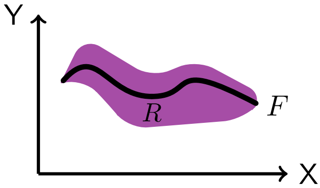
Uniformisation

Uniformisation



Relation $R \subseteq X \times Y$

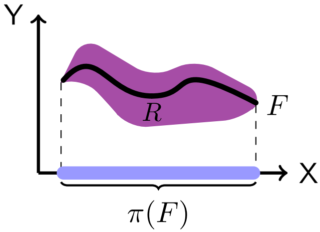
Uniformisation



Relation $R \subseteq X \times Y$

Uniformisation $F \subseteq R$

Uniformisation

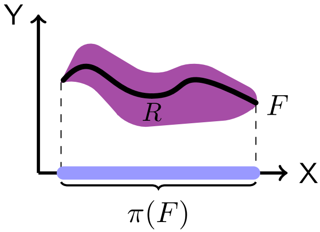


Relation $R \subseteq X \times Y$

Uniformisation $F \subseteq R$

Projection $\pi(F) = \pi(R)$

Uniformisation



Relation $R \subseteq X \times Y$

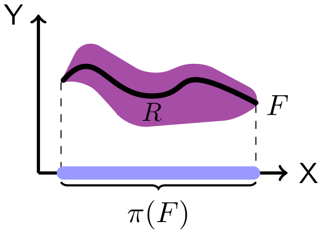
Uniformisation $F \subseteq R$

Projection $\pi(F) = \pi(R)$

Theorem [Axiom of Choice]

Every relation admits a uniformisation.

Uniformisation



Relation $R \subseteq X \times Y$

Uniformisation $F \subseteq R$

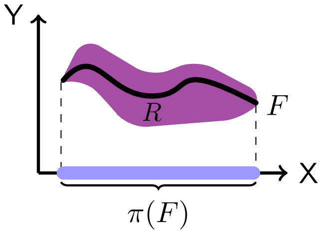
Projection $\pi(F) = \pi(R)$

Theorem [Axiom of Choice]

Every relation admits a uniformisation.

What about definability?

Uniformisation



Relation $R \subseteq X \times Y$

Uniformisation $F \subseteq R$

Projection $\pi(F) = \pi(R)$

Theorem [Axiom of Choice]

Every relation admits a uniformisation.

What about definability?

Theorem (Novikov, Kondô [1938])

Every co-analytic ($\mathbf{\Pi}^1_1$) relation admits a co-analytic uniformisation.

In monadic second-order (MSO) logic?

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓
- finite trees ✓

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓
- finite trees ✓
- infinite words ✓

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓
- finite trees ✓
- infinite words ✓

Rabin's Uniformisation Problem

What about infinite trees?

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓
- finite trees ✓
- infinite words ✓

Rabin's Uniformisation Problem

What about infinite trees?

Theorem (Gurevich, Shelah [1983])

The relation $\mathbf{y} \in \mathbf{X}$ does not admit MSO-def. uniformisation of $\mathbf{y}$.

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓ - finite trees ✓ - infinite words ✓

Rabin's Uniformisation Problem

What about infinite trees?

Theorem (Gurevich, Shelah [1983])

The relation $\mathbf{y} \in \mathbf{X}$ does not admit MSO-def. uniformisation of $\mathbf{y}$.

$\rightsquigarrow$ no MSO-def. choice function over infinite trees

In monadic second-order (MSO) logic?

Structure s over A

Structure s' over B

Pair $(s, s') \sim s \otimes s'$ over $A \times B$

Does every MSO-def. relation admit MSO-def. uniformisation?

- finite words ✓ - finite trees ✓ - infinite words ✓

Rabin's Uniformisation Problem

What about infinite trees?

Theorem (Gurevich, Shelah [1983])

The relation $\mathbf{y} \in \mathbf{X}$ does not admit MSO-def. uniformisation of $\mathbf{y}$.

$\rightsquigarrow$ no MSO-def. choice function over infinite trees

[a model-theoretic argument with some subtleties]

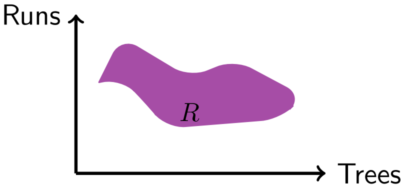
Unambiguity

Unambiguity

$\mathcal{A}$ — non-deterministic tree automaton

Unambiguity

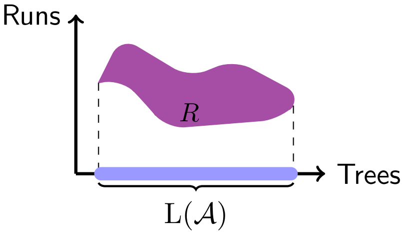
$\mathcal{A}$ — non-deterministic tree automaton



$R = \{(t, \rho) \mid \rho$ is an acc. run over $t\}$

Unambiguity

$\mathcal{A}$ — non-deterministic tree automaton

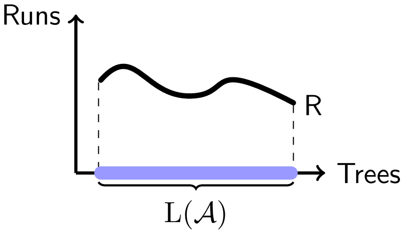


$R = \{(t, \rho) \mid \rho \text{ is an acc. run over } t\}$

$\mathrm{L}(\mathcal{A}) = \{t \mid \exists_\rho\ (t, \rho) \in R\}$

Unambiguity

$\mathcal{A}$ — non-deterministic tree automaton



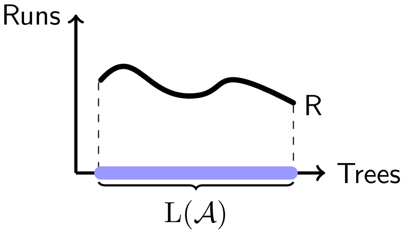
$R = \{(t, \rho) \mid \rho$ is an acc. run over $t\}$

$\mathrm{L}(\mathcal{A}) = \{t \mid \exists_\rho\ (t, \rho) \in R\}$

$\mathcal{A}$ is unambiguous if R is uniformised

Unambiguity

$\mathcal{A}$ — non-deterministic tree automaton



$R = \{(t, \rho) \mid \rho$ is an acc. run over $t\}$

$\mathrm{L}(\mathcal{A}) = \{t \mid \exists_\rho \ (t, \rho) \in R\}$

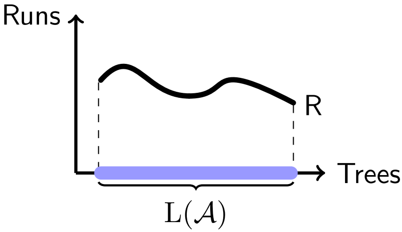
$\mathcal{A}$ is unambiguous if R is uniformised

Theorem (Niwiński, Walukiewicz [1996])

The language $\exists_y \ a(y)$ cannot be recognised by any unambiguous automaton.

Unambiguity

$\mathcal{A}$ — non-deterministic tree automaton



$R = \{(t, \rho) \mid \rho$ is an acc. run over $t\}$

$\mathrm{L}(\mathcal{A}) = \{t \mid \exists_\rho\ (t, \rho) \in R\}$

$\mathcal{A}$ is unambiguous if R is uniformised

Theorem (Niwiński, Walukiewicz [1996])

The language $\exists_y\ a(y)$ cannot be recognised by any unambiguous automaton.

Proof

Any unambiguous automaton for $\exists_y\ a(y)$ induces an MSO-definable choice function.

Gurevich & Shelah ++

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Proof

Pumping of runs of a marking automaton.

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Proof

Pumping of runs of a marking automaton.

State of the art

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Proof

Pumping of runs of a marking automaton.

State of the art

- no uniformisation in MSO

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Proof

Pumping of runs of a marking automaton.

State of the art

- no uniformisation in MSO [$y \in X$ and its variants]

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Proof

Pumping of runs of a marking automaton.

State of the art

- no uniformisation in MSO [$y \in X$ and its variants]
- no unambiguous automata for MSO

Gurevich & Shelah ++

Theorem (Carayol, Löding [2007])

There is no MSO-def. choice function over infinite trees.

Proof

Pumping of runs of a marking automaton.

State of the art

- no uniformisation in MSO [$y \in X$ and its variants]
- no unambiguous automata for MSO [$\exists_y\ a(y)$ and its variants]

Thin trees

Thin trees [also: scattered or tame trees]

Thin trees [also: scattered or tame trees]

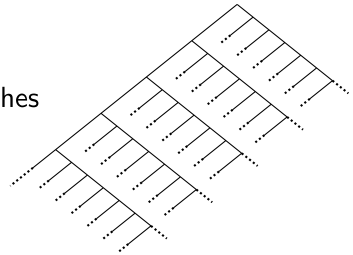
- partial, infinite, labelled, binary trees

Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches

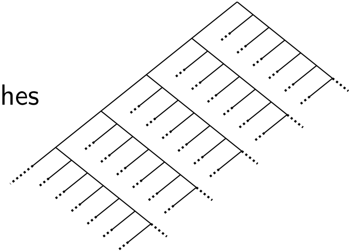
Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees [also: scattered or tame trees]

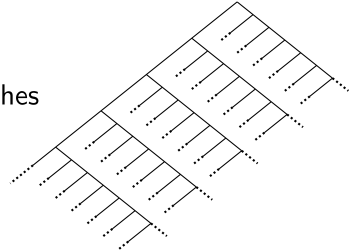
- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees = Finite trees ⊗ Infinite words

Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees = Finite trees ⊗ Infinite words

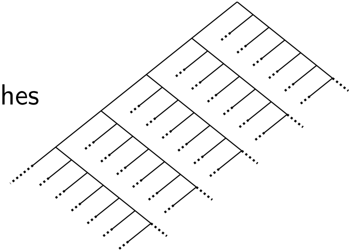
Theorem (Bojańczyk, Idziaszek [2013])

MSO over thin trees is equivalent with thin algebra.

$\rightsquigarrow$ effective characterisations

Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees = Finite trees ⊗ Infinite words

Theorem (Bojańczyk, Idziaszek [2013])

MSO over thin trees is equivalent with thin algebra.

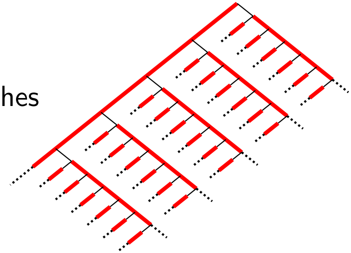
⇝ effective characterisations

Proof

Induction over the structure of a decomposition of a given tree.

Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees = Finite trees ⊗ Infinite words

Theorem (Bojańczyk, Idziaszek [2013])

MSO over thin trees is equivalent with thin algebra.

$\rightsquigarrow$ effective characterisations

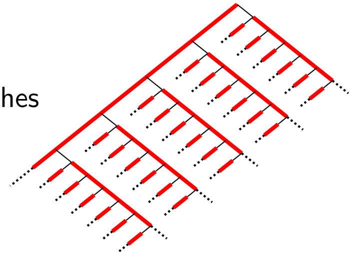
Proof

Induction over the structure of a decomposition of a given tree.

Skeleton — a decomposition of t into separate branches

Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees = Finite trees ⊗ Infinite words

Theorem (Bojańczyk, Idziaszek [2013])

MSO over thin trees is equivalent with thin algebra.

⇝ effective characterisations

Proof

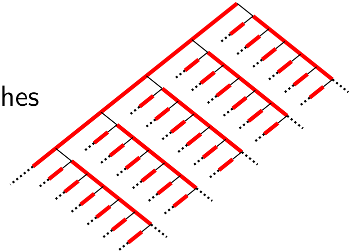
Induction over the structure of a decomposition of a given tree.

Skeleton — a decomposition of t into separate branches

[formally a set of nodes of t]

Thin trees [also: scattered or tame trees]

- partial, infinite, labelled, binary trees
- with only countably many infinite branches



Thin trees = Finite trees ⊗ Infinite words

Theorem (Bojańczyk, Idziaszek [2013])

MSO over thin trees is equivalent with thin algebra.

⇝ effective characterisations

Proof

Induction over the structure of a decomposition of a given tree.

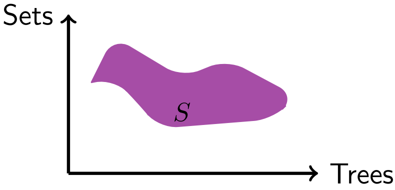
Skeleton — a decomposition of t into separate branches

[formally a set of nodes of t]

t has a skeleton ⇔ t is thin

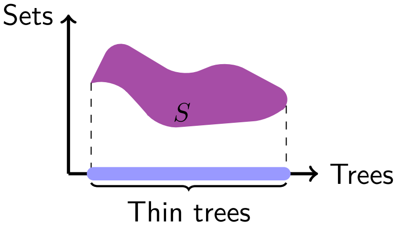
Uniformisation on thin trees

Uniformisation on thin trees



$S = \{(t, \sigma) \mid \sigma$ is a skeleton of $t\}$

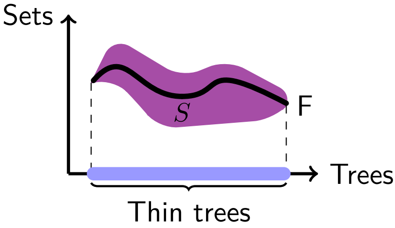
Uniformisation on thin trees



$S = \{(t, \sigma) \mid \sigma$ is a skeleton of $t\}$

t is thin iff t has a skeleton

Uniformisation on thin trees

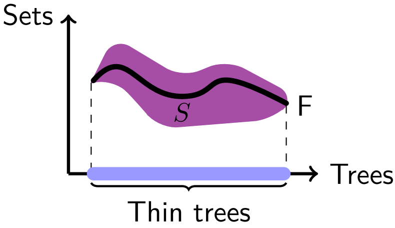


$S = \{(t, \sigma) \mid \sigma$ is a skeleton of $t\}$

t is thin iff t has a skeleton

Is there MSO-def. F?

Uniformisation on thin trees



$S = \{(t,\sigma) \mid \sigma$ is a skeleton of $t\}$

t is thin iff t has a skeleton

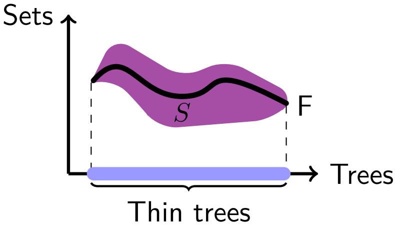
Is there MSO-def. F?

Theorem (S. [2013])

There is **no** MSO-def. uniformisation of skeletons.

Thin trees cannot be recognised by unambiguous automaton.

Uniformisation on thin trees



$S = \{(t, \sigma) \mid \sigma$ is a skeleton of $t\}$

t is thin iff t has a skeleton

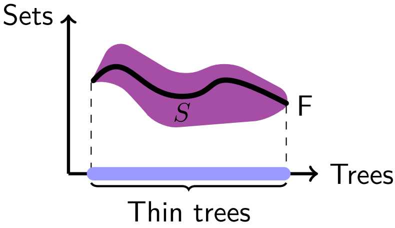
Is there MSO-def. F?

Theorem (S. [2013])

There is **no** MSO-def. uniformisation of skeletons.
Thin trees cannot be recognised by unambiguous automaton.

⇝ new examples of non-uniformisable / ambiguous languages

Uniformisation on thin trees



$S = \{(t, \sigma) \mid \sigma$ is a skeleton of $t\}$

t is thin iff t has a skeleton

Is there MSO-def. F?

Theorem (S. [2013])

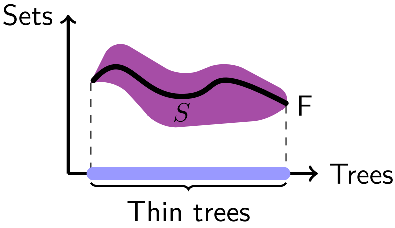
There is **no** MSO-def. uniformisation of skeletons.

Thin trees cannot be recognised by unambiguous automaton.

Proof

A method of proving non-uniformisability via consistent markings.

Uniformisation on thin trees



$S = \{(t, \sigma) \mid \sigma$ is a skeleton of $t\}$

t is thin iff t has a skeleton

Is there MSO-def. F?

Theorem (S. [2013])

There is **no** MSO-def. uniformisation of skeletons.
Thin trees cannot be recognised by unambiguous automaton.

Proof

A method of proving non-uniformisability via consistent markings.

Conjecture (S. [2013])

There is **no** MSO-def. choice function over thin trees.

Consistent markings

Consistent markings

A finite algebra H

i.e. monoid, forest algebra, thin algebra,. . .

Consistent markings

A finite algebra H

i.e. monoid, forest algebra, thin algebra,...

element $h \in H$ $\sim$ type of structures

Consistent markings

A finite algebra H

i.e. monoid, forest algebra, thin algebra,...

element $h \in H$	$\sim$	type of structures
operation $\cdot$ in H	$\sim$	composition of structures

Consistent markings

A finite algebra H

i.e. monoid, forest algebra, thin algebra,...

element $h \in H$	$\sim$	type of structures
operation $\cdot$ in H	$\sim$	composition of structures
homomorphism $\alpha\colon \mathrm{Struct} \to H$	$\sim$	assignment of actual types

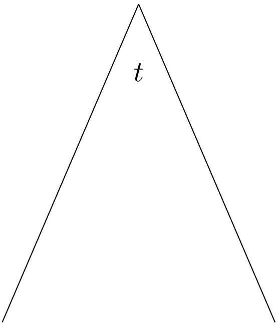
Consistent markings

A finite algebra H

i.e. monoid, forest algebra, thin algebra,. . .

element $h \in H$	$\sim$	type of structures
operation $\cdot$ in H	$\sim$	composition of structures
homomorphism $\alpha\colon \mathrm{Struct} \to H$	$\sim$	assignment of actual types

Marking : a labelling τ of a tree t by H



Consistent markings

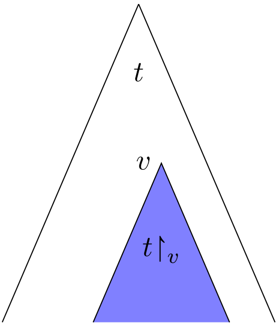
A finite algebra H

i.e. monoid, forest algebra, thin algebra,...

element $h \in H$	$\sim$	type of structures
operation $\cdot$ in H	$\sim$	composition of structures
homomorphism $\alpha\colon \mathrm{Struct} \to H$	$\sim$	assignment of actual types

Marking : a labelling τ of a tree t by H

$\tau(v) \equiv$ declared type of $t{\upharpoonright}_v$



Consistent markings

A finite algebra H

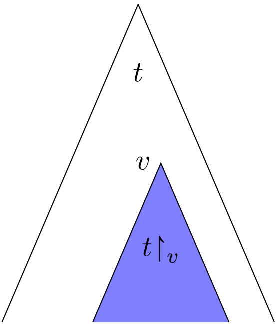
i.e. monoid, forest algebra, thin algebra,...

element $h \in H$	$\sim$	type of structures
operation $\cdot$ in H	$\sim$	composition of structures
homomorphism $\alpha\colon \mathrm{Struct} \to H$	$\sim$	assignment of actual types

Marking : a labelling τ of a tree t by H

$\tau(v) \equiv$ declared type of $t{\restriction}_v$

Actual marking : $\tau(v) = \alpha(t{\restriction}_v)$



Consistent markings

A finite algebra H

i.e. monoid, forest algebra, thin algebra,...

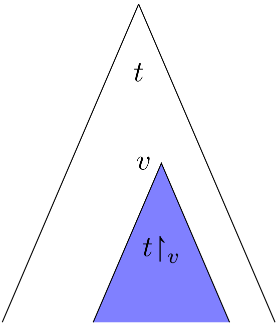
element $h \in H$	$\sim$	type of structures
operation $\cdot$ in H	$\sim$	composition of structures
homomorphism $\alpha\colon \mathrm{Struct} \to H$	$\sim$	assignment of actual types

Marking : a labelling τ of a tree t by H

$\tau(v) \equiv$ declared type of $t{\upharpoonright}_v$

Actual marking : $\tau(v) = \alpha(t{\upharpoonright}_v)$

if there exists $\alpha\colon \mathrm{Trees} \to H$...



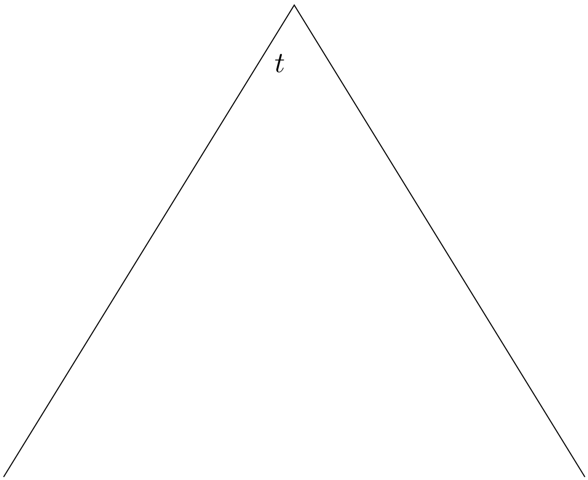
Consistent markings

Consistent markings

τ is consistent if the declarations are consistent along branches

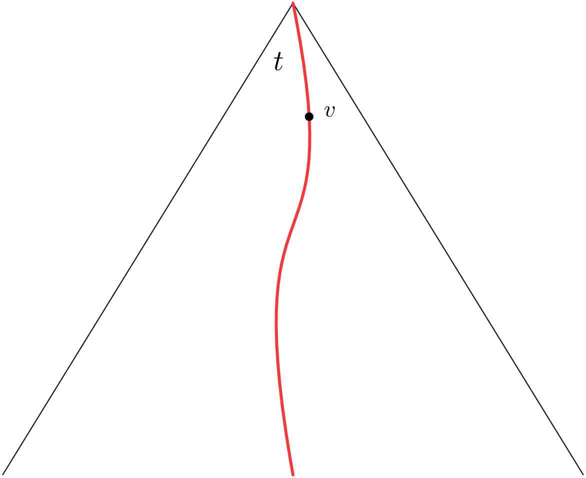
Consistent markings

τ is consistent if the declarations are consistent along branches



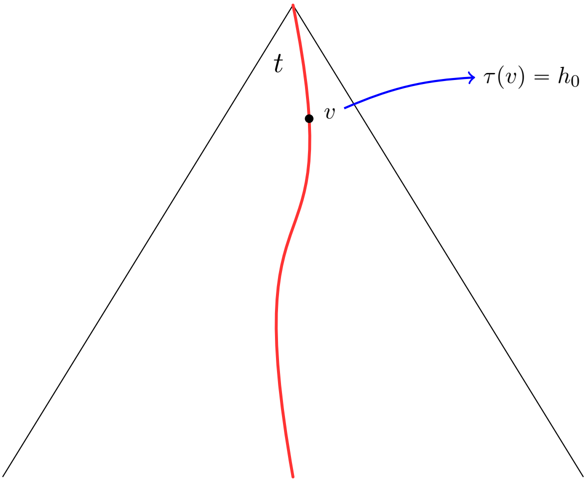
Consistent markings

τ is consistent if the declarations are consistent along branches



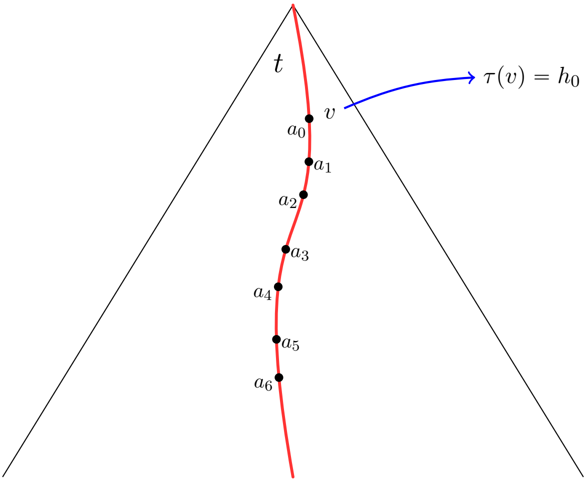
Consistent markings

τ is consistent if the declarations are consistent along branches



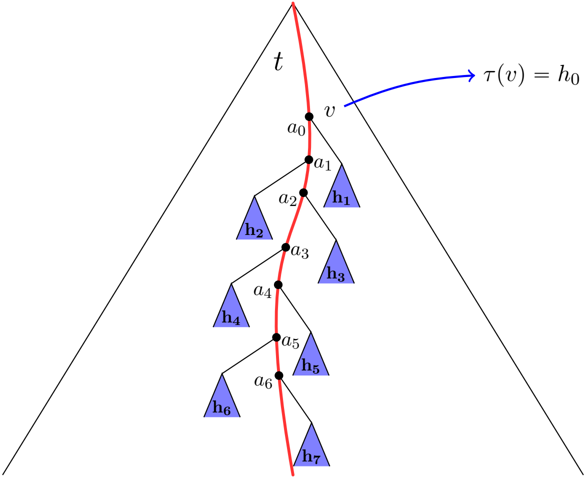
Consistent markings

τ is consistent if the declarations are consistent along branches



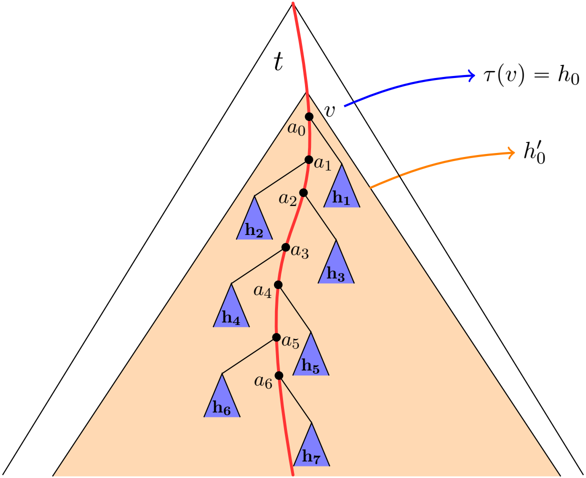
Consistent markings

τ is consistent if the declarations are consistent along branches



Consistent markings

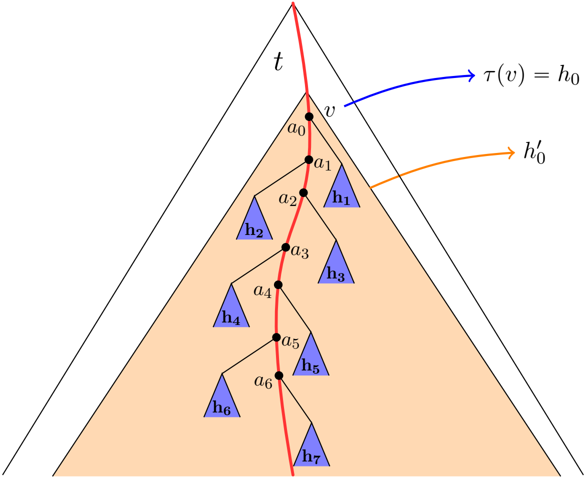
τ is consistent if the declarations are consistent along branches



Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]



Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

$$H = \{h_0, h_1\}, \quad h_0 \equiv \text{"no letter } a\text{"}, \quad h_1 \equiv \text{"exists letter } a\text{"}$$

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h_0'$]

Example

$H = \{h_0, h_1\}, \quad h_0 \equiv$ "no letter a", $\quad h_1 \equiv$ "exists letter a"

For all v let:

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

$H = \{h_0, h_1\}$, $h_0 \equiv$ "no letter a", $h_1 \equiv$ "exists letter a"

For all v let: $t(v) = b$

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

$H = \{h_0, h_1\}$, $h_0 \equiv$ "no letter a", $h_1 \equiv$ "exists letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_1$ ("exists a")

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h_0'$]

Example

$H = \{h_0, h_1\}$, $h_0 \equiv$ "no letter a", $h_1 \equiv$ "exists letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_1$ ("exists a")

τ is consistent!

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

$H = \{h_0, h_1\}, \quad h_0 \equiv$ "no letter a", $\quad h_1 \equiv$ "exists letter a"

For all v let: $\quad t(v) = b \quad$ and $\quad \tau(v) = h_1$ ("exists a")

τ is consistent!

Theorem (S. [2013])

There is no MSO-def choice function on thin trees

iff

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

$H = \{h_0, h_1\}$, $h_0 \equiv$ "no letter a", $h_1 \equiv$ "exists letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_1$ ("exists a")

τ is consistent!

Theorem (S. [2013])

There is no MSO-def choice function on thin trees

iff

For every finite thin algebra H and every tree t (thin or non-thin)

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h'_0$]

Example

$H = \{h_0, h_1\}$, $h_0 \equiv$ "no letter a", $h_1 \equiv$ "exists letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_1$ ("exists a")

τ is consistent!

Theorem (S. [2013])

There is no MSO-def choice function on thin trees

iff

For every finite thin algebra H and every tree t (thin or non-thin)

there exists a consistent marking of t by H.

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_0 = h_0'$]

Example

$H = \{h_0, h_1\}$, $h_0 \equiv$ "no letter a", $h_1 \equiv$ "exists letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_1$ ("exists a")

τ is consistent!

Theorem (S. [2013])

There is no MSO-def choice function on thin trees

iff

For every finite thin algebra H and every tree t (thin or non-thin) there exists a consistent marking of t by H.

[no actual marking because $\alpha\colon \mathrm{Thin} \to H$ (not $\alpha\colon \mathrm{Trees} \to H$)]

Summary

Summary

New examples of non-uniformisability / ambiguity in MSO

Summary

New examples of non-uniformisability / ambiguity in MSO

New conjecture: **no** MSO-def. choice function over thin trees

Summary

New examples of non-uniformisability / ambiguity in MSO

New conjecture: **no** MSO-def. choice function over thin trees

Equivalently: every tree admits a consistent marking by every thin algebra

Summary

New examples of non-uniformisability / ambiguity in MSO

New conjecture: **no** MSO-def. choice function over thin trees

Equivalently: every tree admits a consistent marking by every thin algebra

Example (Bojańczyk, Kuperberg [2014])

Even if $t \equiv a_0$ there may be no consistent $\tau \equiv h_0$ (i.e. constant).

Summary

New examples of non-uniformisability / ambiguity in MSO

New conjecture: **no** MSO-def. choice function over thin trees

Equivalently: every tree admits a consistent marking by every thin algebra

Example (Bojańczyk, Kuperberg [2014])

Even if $t \equiv a_0$ there may be no consistent $\tau \equiv h_0$ (i.e. constant).

Theorem (Bilkowski, S. [2013])

If there is **no** MSO-def. choice function over thin trees

Summary

New examples of non-uniformisability / ambiguity in MSO

New conjecture: **no** MSO-def. choice function over thin trees

Equivalently: every tree admits a consistent marking by every thin algebra

Example (Bojańczyk, Kuperberg [2014])

Even if $t \equiv a_0$ there may be no consistent $\tau \equiv h_0$ (i.e. constant).

Theorem (Bilkowski, S. [2013])

If there is **no** MSO-def. choice function over thin trees

then it is decidable if a regular language of complete trees

Summary

New examples of non-uniformisability / ambiguity in MSO

New conjecture: **no** MSO-def. choice function over thin trees

Equivalently: every tree admits a consistent marking by every thin algebra

Example (Bojańczyk, Kuperberg [2014])

Even if $t \equiv a_0$ there may be no consistent $\tau \equiv h_0$ (i.e. constant).

Theorem (Bilkowski, S. [2013])

If there is **no** MSO-def. choice function over thin trees
then it is decidable if a regular language of complete trees
is bi-unambiguous [both L and $\mathrm{Trees}{-}L$ are unambiguous].