Definability of choice over scattered trees in MSO

Michał Skrzypczak

Highlights 2015
Prague

Logic:

Logic: **Monadic Second-Order** (MSO) logic

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ $\quad$ (x — node)

Logic: **Monadic Second-Order** ($\textsc{mso}$) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ $(x$ — node$)$
- $\exists_X$, $\forall_X$ $(X$ — set of nodes$)$
- $x \in X$, $x = y$, predicates

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

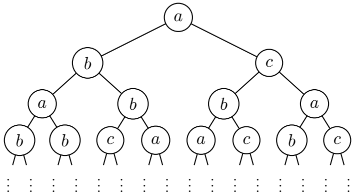
words

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ $\big(x$ — node$\big)$
- $\exists_X$, $\forall_X$ $\big(X$ — set of nodes$\big)$
- $x \in X$, $x = y$, predicates

Structures:

words and **trees**

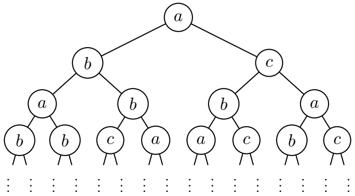


Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

words and **trees**



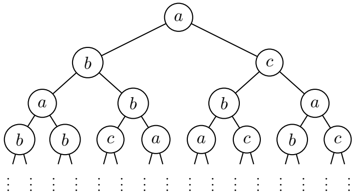
Scattered trees:

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

words and **trees**



Scattered trees:

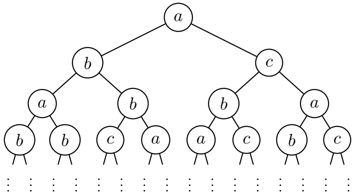
Partial trees with only **countably many** branches

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

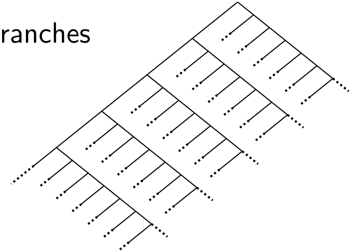
Structures:

words and **trees**



Scattered trees:

Partial trees with only **countably many** branches

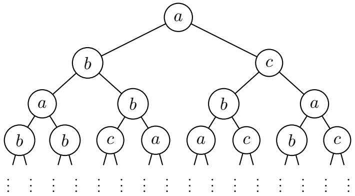


Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

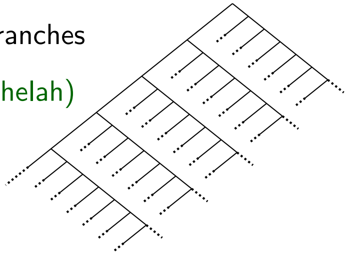
words and **trees**



Scattered trees:

Partial trees with only **countably many** branches

- uniformisation with parameters (Lifsches, Shelah)

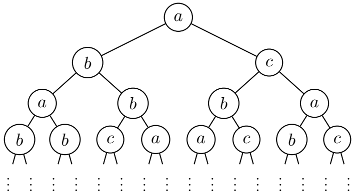


Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

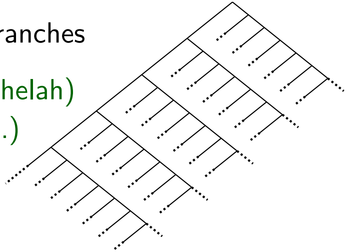
words and **trees**



Scattered trees:

Partial trees with only **countably many** branches

- uniformisation with parameters (Lifsches, Shelah)
- algebraic characterisations (Bojańczyk et al.)

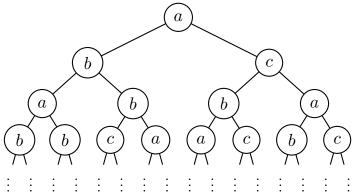


Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

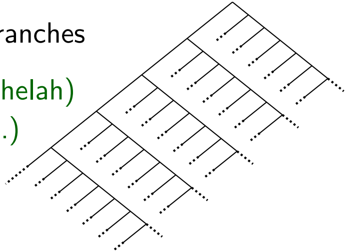
Structures:

words and **trees**



Scattered trees:

Partial trees with only **countably many** branches



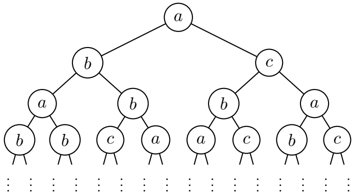
- uniformisation with parameters (Lifsches, Shelah)
- algebraic characterisations (Bojańczyk et al.)
- logical interpretations (Rabinovich et al.)

Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

Structures:

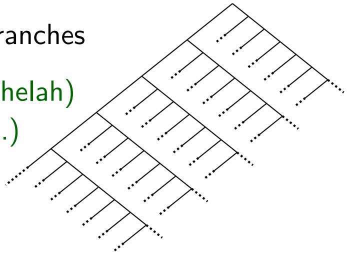
words and **trees**



Scattered trees:

Partial trees with only **countably many** branches

- uniformisation with parameters (Lifsches, Shelah)
- algebraic characterisations (Bojańczyk et al.)
- logical interpretations (Rabinovich et al.)
- descriptive properties (S.)

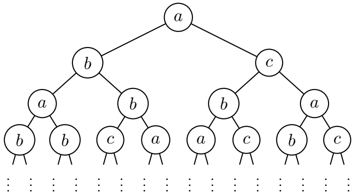


Logic: **Monadic Second-Order** (MSO) logic

- $\exists_x$, $\forall_x$ (x — node)
- $\exists_X$, $\forall_X$ (X — set of nodes)
- $x \in X$, $x = y$, predicates

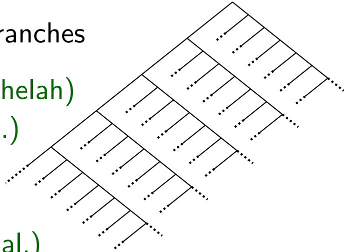
Structures:

words and **trees**



Scattered trees:

Partial trees with only **countably many** branches



- uniformisation with parameters (Lifsches, Shelah)
- algebraic characterisations (Bojańczyk et al.)
- logical interpretations (Rabinovich et al.)
- descriptive properties (S.)
- boundedness and determinacy (Fijalkow et al.)

Choice

Choice

$\varphi(x, X)$ defines choice if

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \emptyset.\ \exists! x \in X. \quad \varphi(x, X)$$

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \emptyset.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words:

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X"

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \emptyset.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])

There is **no** MSO-definable choice over infinite trees.

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X.\quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])

There is **no** MSO-definable choice over infinite trees. ✗

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \emptyset.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])

There is **no** MSO-definable choice over infinite trees. ✗

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])

There is **no** MSO-definable choice over infinite trees. ✗

Conjecture

There is **no** MSO-definable choice over **scattered** trees. ???

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \emptyset. \ \exists! x \in X. \quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])

There is **no** MSO-definable choice over infinite trees. ✗

Conjecture

There is **no** MSO-definable choice over **scattered** trees. **???**

Theorem (Bilkowski, S. [2013])

The above conjecture implies

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \varnothing.\ \exists! x \in X.\quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])

There is **no** MSO-definable choice over infinite trees. ✗

Conjecture

There is **no** MSO-definable choice over **scattered** trees. ???

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation

Choice

$\varphi(x, X)$ defines choice if

$$\forall X \neq \emptyset.\ \exists! x \in X.\quad \varphi(x, X)$$

I.e. from every non-empty set of nodes X

$\varphi(x, X)$ chooses a unique element $x \in X$

Words: $\varphi(x, X) \equiv$ "x is the $<$-minimal element of X" ✓

Theorem (Gurevich, Shelah [1983], Carayol, Löding [2007])
There is **no** MSO-definable choice over infinite trees. ✗

Conjecture
There is **no** MSO-definable choice over **scattered** trees. ???

Theorem (Bilkowski, S. [2013])
The above conjecture implies an effective characterisation of bi-unambiguous languages of complete trees.

Consistent markings

Consistent markings

An algebra H

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

Consistent markings

An algebra H

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

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

Consistent markings

An 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

An 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

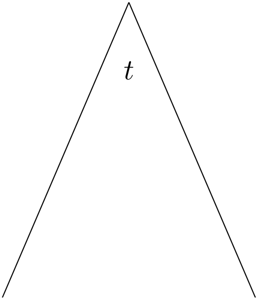
Consistent markings

An 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

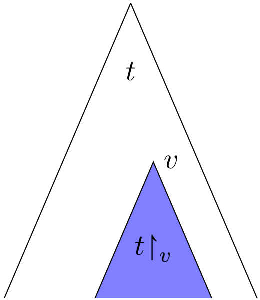
An algebra H

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 $t{\restriction}_v$



Consistent markings

An algebra H

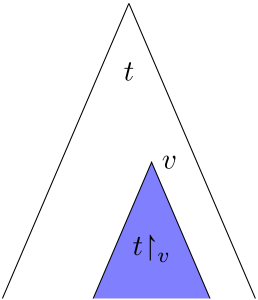
i.e. monoid, forest algebra, thin algebra,...

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

An 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

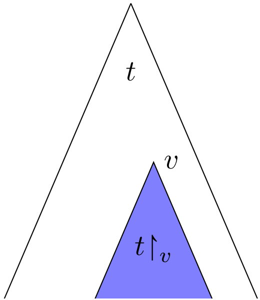
homomorphism $\alpha\colon \mathrm{Struct} \to H$ $\sim$ assignment of actual types

Marking : a labelling τ of a tree t by H

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

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

[if there exists $\alpha\colon \mathrm{Trees} \to H\ldots$]



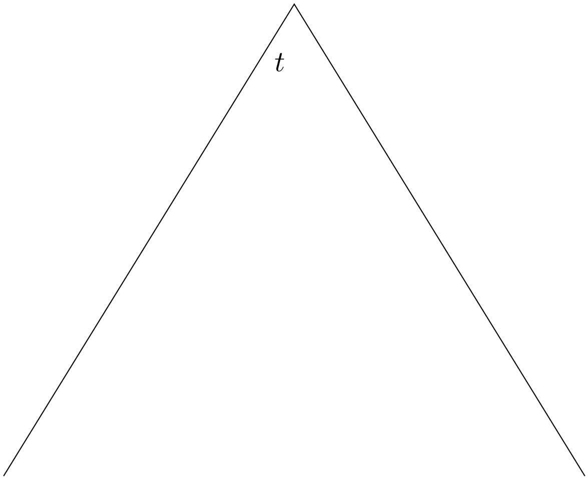
Consistent markings

Consistent markings

τ is consistent if the declarations are consistent along branches

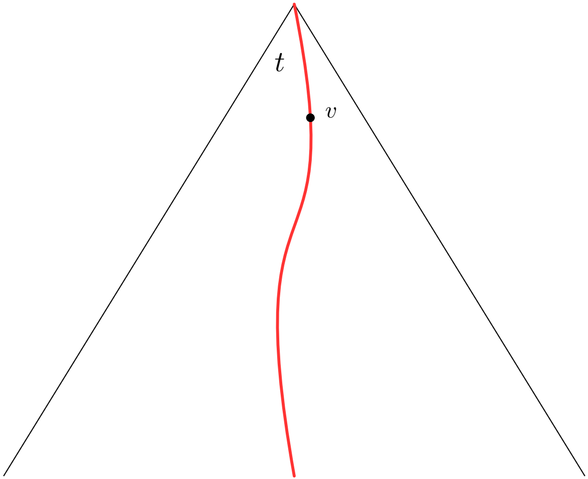
Consistent markings

τ is consistent if the declarations are consistent along branches



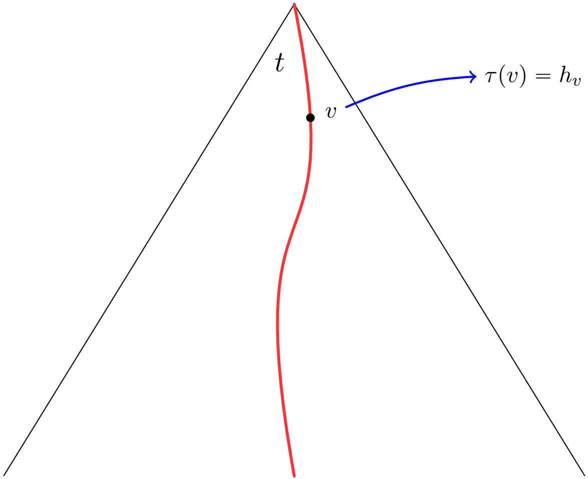
Consistent markings

τ is consistent if the declarations are consistent along branches



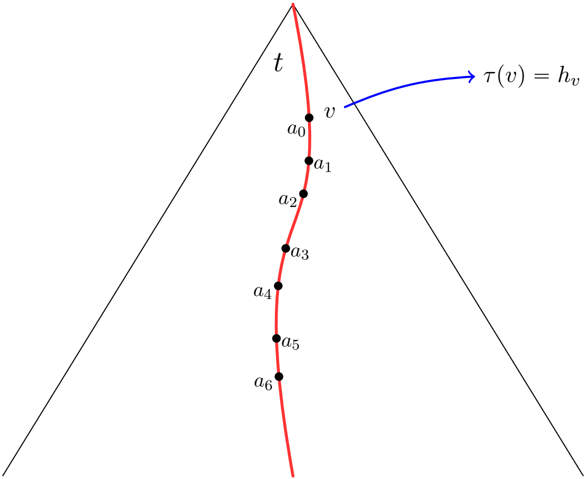
Consistent markings

τ is consistent if the declarations are consistent along branches



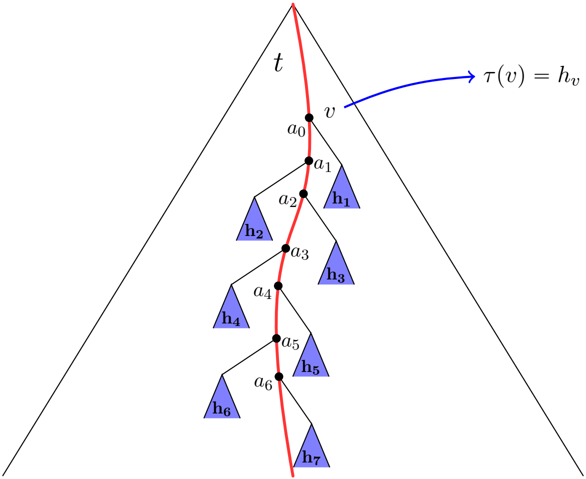
Consistent markings

τ is consistent if the declarations are consistent along branches



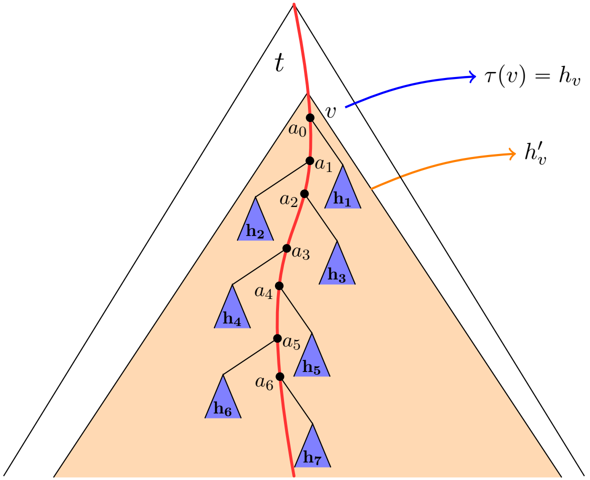
Consistent markings

τ is consistent if the declarations are consistent along branches



Consistent markings

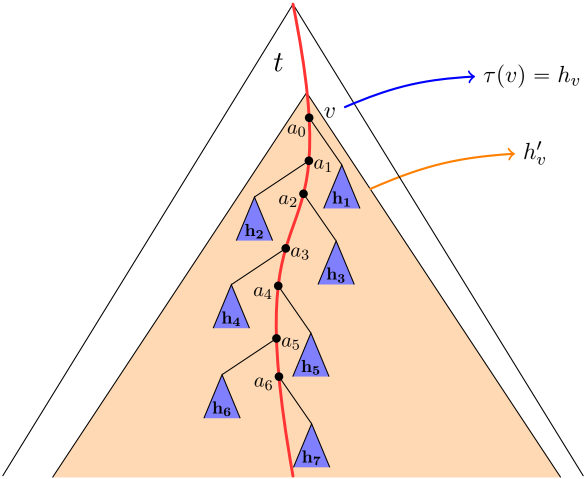
τ is consistent if the declarations are consistent along branches



Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]



Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no 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 $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no 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 $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!

Consistent markings

τ is consistent if the declarations are consistent along branches

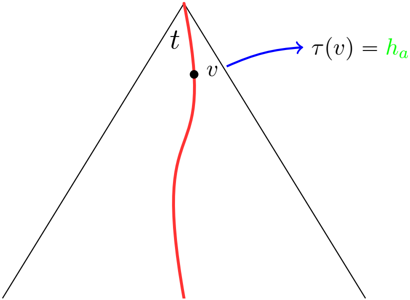
[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!



Consistent markings

τ is consistent if the declarations are consistent along branches

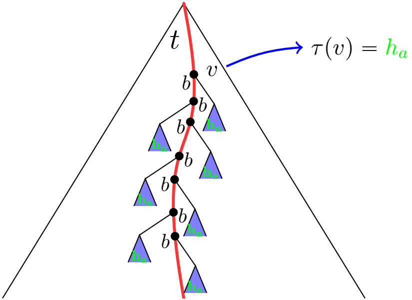
[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!



Consistent markings

τ is consistent if the declarations are consistent along branches

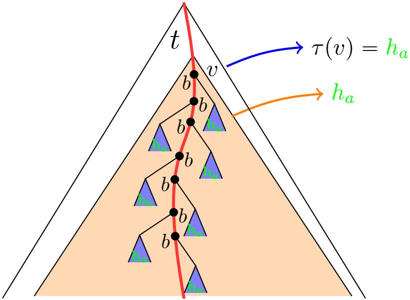
[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_a$ ("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 $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

For all v let: $t(v) = b$ and $\tau(v) = h_a$ ("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 $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

For every finite thin algebra H and every complete tree t

Consistent markings

τ is consistent if the declarations are consistent along branches

[it is enough to use thin algebra to check if $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

For every finite thin algebra H and every complete tree t

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 $h_v = h'_v$]

Example

$H = \{h_a, h_b\}$, $h_a \equiv$ "exists letter a", $h_b \equiv$ "no letter a"

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

τ is consistent!

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

For every finite thin algebra H and every complete tree t

there exists a consistent marking of t by H.

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

Summary

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation of bi-unambiguous languages of complete trees.

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation of bi-unambiguous languages of complete trees.

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation of bi-unambiguous languages of complete trees.

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

For every finite thin algebra H and every tree t (scattered or not)

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation of bi-unambiguous languages of complete trees.

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

For every finite thin algebra H and every tree t (scattered or not) there exists a consistent marking of t by H.

Summary

Conjecture

There is **no** MSO-definable choice over **scattered** trees.

Theorem (Bilkowski, S. [2013])

The above conjecture implies an effective characterisation of bi-unambiguous languages of complete trees.

Theorem (S. [2013])

There is **no** MSO-definable choice over **scattered** trees

iff

For every finite thin algebra H and every tree t (scattered or not) there exists a consistent marking of t by H.

$\rightsquigarrow$ connections with path continuous hyper-clones of Blumensath