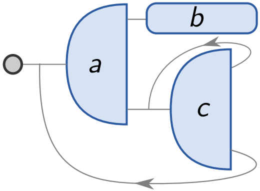
Tree Algebras and Bisimulation-Invariant MSO on Finite Graphs

Thomas Colcombet, Amina Doumane, Denis Kuperberg

CNRS, LIP, ENS Lyon

MOVE Team Seminar

Transitions systems



Specifying properties

MSO formulae:

$$\varphi := a(x) \mid E(x,y) \mid x \in X \mid \exists X.\varphi \mid \neg\varphi \mid \varphi \vee \varphi$$

Example: $\varphi(r)$ for "$\exists\ \infty$ path from the root r":
$\exists X.$
$r \in X \wedge$
$\forall x. x \in X \Rightarrow \exists y. E(x,y) \wedge y \in X$

Specifying properties

MSO formulae:

$$\varphi := a(x) \mid E(x,y) \mid x \in X \mid \exists X.\varphi \mid \neg\varphi \mid \varphi \vee \varphi$$

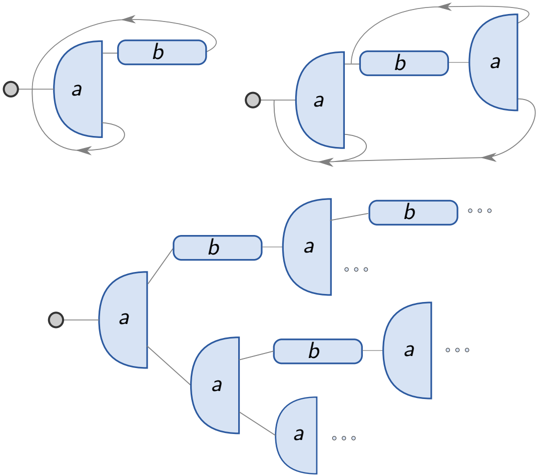
Example: $\varphi(r)$ for "$\exists$ ∞ path from the root r":
$\exists X.$
$r \in X \wedge$
$\forall x.x \in X \Rightarrow \exists y.E(x,y) \wedge y \in X$

μ-calculus formulae:

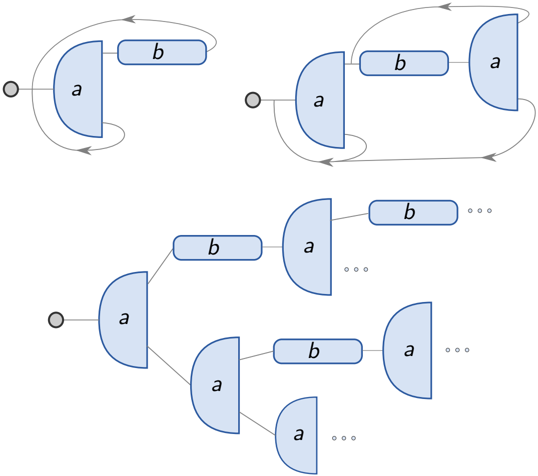
$$\psi := a \mid \psi \vee \psi \mid \neg\psi \mid \diamond\psi \mid \Box\psi \mid \mu X.\psi \mid \nu X.\psi$$

Example: ψ for "$\exists$ ∞ path from the root": $\nu X. \diamond X$

Unfold and Bisimulation equivalence

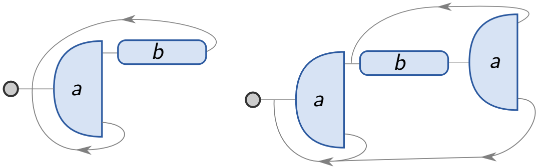


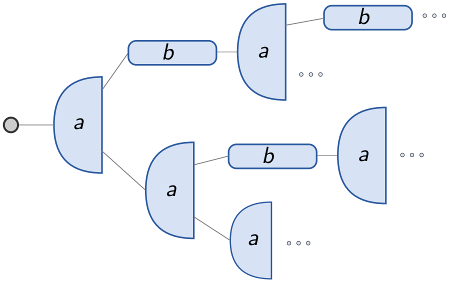
Unfold and Bisimulation equivalence



Bisimulation = unfold + children duplication

Unfold and Bisimulation equivalence





Bisimulation = unfold + children duplication

Fact: μ-calculus is bisimulation-invariant.

Starting point

Theorem (Janin and Walukiewicz 1996)

For properties of systems, the following are equivalent:

1. Being MSO-definable and bisimulation-invariant.
2. Being μ-calculus-definable.

Starting point

Theorem (Janin and Walukiewicz 1996)

For properties of systems, the following are equivalent:

1. Being MSO-definable and bisimulation-invariant.
2. Being μ-calculus-definable.

μ-calculus $\rightarrow$ bisim-inv MSO : Easy

Bisim-inv MSO $\rightarrow$ μ-calculus : Hard

Proof sketch for bisim-inv $MSO \to \mu$-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.

Proof sketch for bisim-inv *MSO* $\to$ μ-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]

Proof sketch for bisim-inv $MSO \to \mu$-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]
- $\mathcal{A} \rightsquigarrow \mu$-calculus formula ψ. [Janin-Walukiewicz 1996]

Proof sketch for bisim-inv $MSO \to \mu$-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]
- $\mathcal{A} \rightsquigarrow \mu$-calculus formula ψ. [Janin-Walukiewicz 1996]

Correctness:

Proof sketch for bisim-inv *MSO* $\rightarrow$ μ-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]
- $\mathcal{A} \rightsquigarrow$ μ-calculus formula ψ. [Janin-Walukiewicz 1996]

Correctness:

- φ and ψ are equivalent on infinite trees

Proof sketch for bisim-inv $MSO \to \mu$-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]
- $\mathcal{A} \rightsquigarrow \mu$-calculus formula ψ. [Janin-Walukiewicz 1996]

Correctness:

- φ and ψ are equivalent on infinite trees
- Every system is bisimilar to an infinite tree

Proof sketch for bisim-inv $MSO \to \mu$-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]
- $\mathcal{A} \rightsquigarrow \mu$-calculus formula ψ. [Janin-Walukiewicz 1996]

Correctness:

- φ and ψ are equivalent on infinite trees
- Every system is bisimilar to an infinite tree
- φ and ψ are bisim-invariant

Proof sketch for bisim-inv $MSO \to \mu$-calculus

Let $\varphi \in$ bisim-inv MSO:

- φ is in particular a formula on infinite trees.
- $\varphi \rightsquigarrow$ automaton $\mathcal{A}$ on infinite trees. [Rabin 1968]
- $\mathcal{A} \rightsquigarrow \mu$-calculus formula ψ. [Janin-Walukiewicz 1996]

Correctness:

- φ and ψ are equivalent on infinite trees
- Every system is bisimilar to an infinite tree
- φ and ψ are bisim-invariant
- $\implies \varphi$ and ψ are equivalent on all systems

Finite systems

The proof of Janin-Walukiewicz needs bisim-inv on infinite systems.

Finite systems

The proof of Janin-Walukiewicz needs bisim-inv on infinite systems.

Finite model property for μ-calculus:
If ψ has a model then it has a finite one.

Finite systems

The proof of Janin-Walukiewicz needs bisim-inv on infinite systems.

Finite model property for μ-calculus:
If ψ has a model then it has a finite one.

Can we restrict the theorem to finite systems ?

Finite systems

The proof of Janin-Walukiewicz needs bisim-inv on infinite systems.

Finite model property for μ-calculus:
If ψ has a model then it has a finite one.

Can we restrict the theorem to finite systems ?

Main Contribution

For properties of **finite** systems, the following are equivalent:

1. Being MSO-definable and bisimulation-invariant.
2. Being μ-calculus-definable.

Examples of the difference

MSO formula φ for "$\exists$ cycle":

- φ is not bisim-invariant on all systems.
- φ is bisim-invariant on **finite** systems.
- Equivalent to $\psi = \nu X. \diamond X$ on finite systems.

Examples of the difference

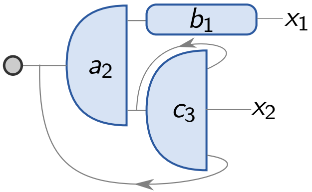
MSO formula φ for "$\exists$ cycle":

- φ is not bisim-invariant on all systems.
- φ is bisim-invariant on **finite** systems.
- Equivalent to $\psi = \nu X. \diamond X$ on finite systems.

$\Longrightarrow$ using Janin-Walukiewicz does not work for finite systems.

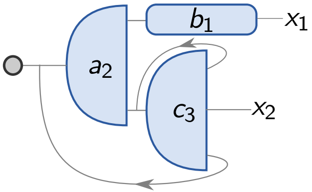
Algebra of systems

Systems have open ports and arities:



Algebra of systems

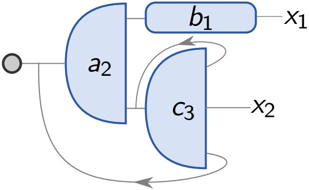
Systems have open ports and arities:



Algebra: Remember only relevant information about a system.
Arity stratification $\rightsquigarrow$ Algebra $\mathcal{A} = (A_n)_{n\in\mathbb{N}}$.

Algebra of systems

Systems have open ports and arities:

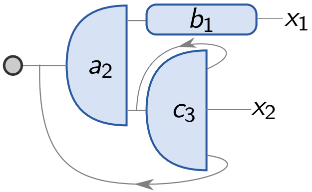


Algebra: Remember only relevant information about a system.
Arity stratification $\rightsquigarrow$ Algebra $\mathcal{A} = (A_n)_{n\in\mathbb{N}}$.

Example: $L = \{\text{Systems with an } a\}$. $A_n = \{a_n, b_n\}$

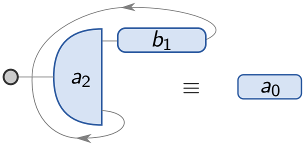
Algebra of systems

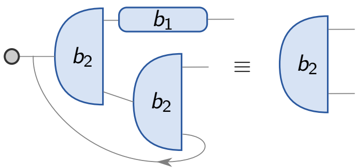
Systems have open ports and arities:



Algebra: Remember only relevant information about a system.
Arity stratification $\rightsquigarrow$ Algebra $\mathcal{A} = (A_n)_{n \in \mathbb{N}}$.

Example: $L = \{$Systems with an $a\}$. $A_n = \{a_n, b_n\}$



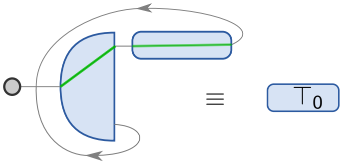


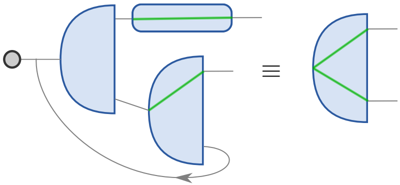
Then $L = \{$Systems evaluating to $a_0\}$, via h :Systems$\to \mathcal{A}$.

Another example of algebra

Language $L = \{\exists \text{ branch with } \infty \text{ many } a\text{'s}\}$.

Then $A_n = 2^{\{1,\dots,n\}} \cup \{\top_n\}$, and $L = h^{-1}(\top_0)$

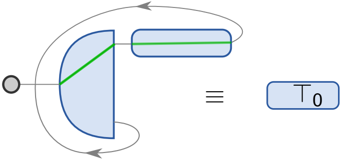


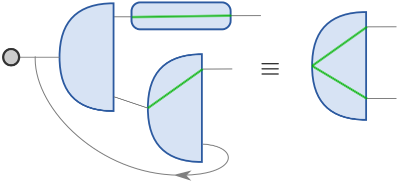


Another example of algebra

Language $L = \{\exists \text{ branch with } \infty \text{ many } a\text{'s}\}$.

Then $A_n = 2^{\{1,\dots,n\}} \cup \{\top_n\}$, and $L = h^{-1}(\top_0)$





$\mathcal{A}$ is sortwise-finite but not sortwise-bounded.
Intuition: Enough for regularity.

Recognizability

Main Contribution 2

If L is recognized by a sortwise-finite algebra, then L is recognized by some automaton model.

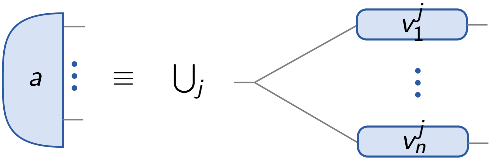
Recognizability

Main Contribution 2

If L is recognized by a sortwise-finite algebra, then L is recognized by some automaton model.

Key Lemma

$\forall a \in A_n$, $\exists (v_i^j)_{i,j}$ from A_1 such that:



With new operators in the algebras.

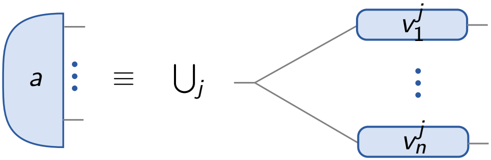
Recognizability

Main Contribution 2

If L is recognized by a sortwise-finite algebra, then L is recognized by some automaton model.

Key Lemma

$\forall a \in A_n$, $\exists (v_i^j)_{i,j}$ from A_1 such that:



With new operators in the algebras.

Consequences

- A_1 actually contains all the information about A_n.
- Algebras can be turned into automata.

Proof of Main Theorem

μ-**calculus** $\rightarrow$ **bisim-inv MSO** is easy, same as before.

Proof of Main Theorem

μ-**calculus** $\rightarrow$ **bisim-inv MSO** is easy, same as before.

Bisim-inv MSO $\rightarrow$ μ-**calculus**:

- MSO $\rightarrow$ algebra by standard compositional methods [Feferman-Vaught 1959, Shelah 1975].

Proof of Main Theorem

μ-**calculus** $\rightarrow$ **bisim-inv MSO** is easy, same as before.

Bisim-inv MSO $\rightarrow$ μ-**calculus**:

- MSO $\rightarrow$ algebra by standard compositional methods [Feferman-Vaught 1959, Shelah 1975].
- Algebra $\rightarrow$ unfold-invariant automata by the key lemma.

Proof of Main Theorem

μ-**calculus** $\rightarrow$ **bisim-inv MSO** is easy, same as before.

Bisim-inv MSO $\rightarrow$ μ-**calculus**:

- MSO $\rightarrow$ algebra by standard compositional methods [Feferman-Vaught 1959, Shelah 1975].
- Algebra $\rightarrow$ unfold-invariant automata by the key lemma.
- Unfold-invariant $\rightarrow$ bisimulation-invariant automata by adding duplication as in [Janin-Walukiewicz 1996].

Proof of Main Theorem

μ-**calculus** $\rightarrow$ **bisim-inv MSO** is easy, same as before.

Bisim-inv MSO $\rightarrow$ μ-**calculus**:

- MSO $\rightarrow$ algebra by standard compositional methods [Feferman-Vaught 1959, Shelah 1975].
- Algebra $\rightarrow$ unfold-invariant automata by the key lemma.
- Unfold-invariant $\rightarrow$ bisimulation-invariant automata by adding duplication as in [Janin-Walukiewicz 1996].
- Bisim-invariant automata $\rightarrow$ μ-calculus as in [Janin-Walukiewicz 1996].

Proof of Main Theorem

μ-**calculus** $\rightarrow$ **bisim-inv MSO** is easy, same as before.

Bisim-inv MSO $\rightarrow$ μ-**calculus**:

- MSO $\rightarrow$ algebra by standard compositional methods [Feferman-Vaught 1959, Shelah 1975].
- Algebra $\rightarrow$ unfold-invariant automata by the key lemma.
- Unfold-invariant $\rightarrow$ bisimulation-invariant automata by adding duplication as in [Janin-Walukiewicz 1996].
- Bisim-invariant automata $\rightarrow$ μ-calculus as in [Janin-Walukiewicz 1996].

Thanks for your attention!