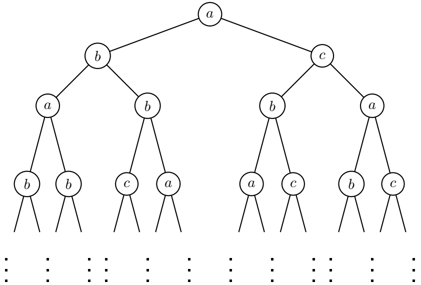
Deciding the index hierarchies of game automata

Alessandro Facchini Filip Murlak Michał Skrzypczak

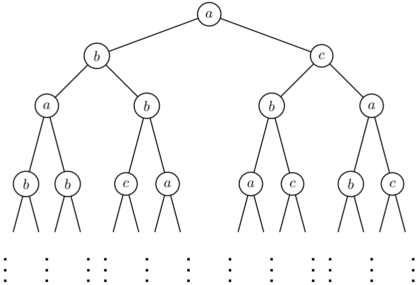
University of Warsaw

LICS 2013
New Orleans

Infinite trees are very rich

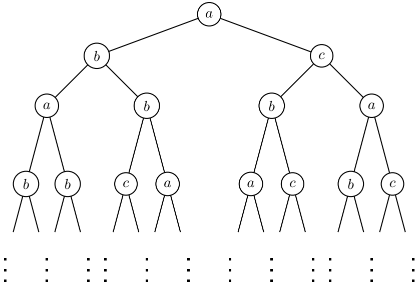


Infinite trees are very rich



One infinite tree can encode:

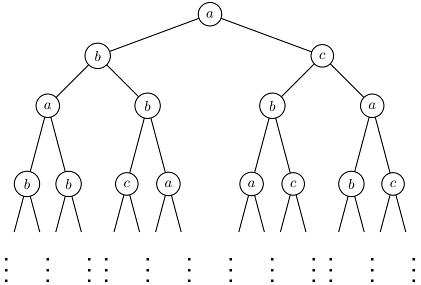
Infinite trees are very rich



One infinite tree can encode:

- an arbitrary set of finite words

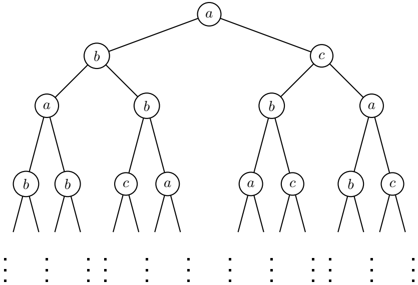
Infinite trees are very rich



One infinite tree can encode:

- an arbitrary set of finite words
- all the possible futures of a nondeterministic system

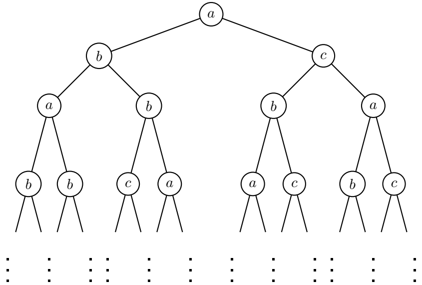
Infinite trees are very rich



One infinite tree can encode:

- an arbitrary set of finite words
- all the possible futures of a nondeterministic system
- a strategy in an infinite-duration game

Infinite trees are very rich



One infinite tree can encode:

- an arbitrary set of finite words
- all the possible futures of a nondeterministic system
- a strategy in an infinite-duration game

$\rightsquigarrow$ Application in verification and model-checking

Specifying properties of trees

- FO, CTL*, . . .
- modal μ-calculus
- finite automata
- Monadic Second-Order logic

Specifying properties of trees

- FO, CTL*, . . .
- modal μ-calculus
- finite automata
- Monadic Second-Order logic

Theorem (Rabin 1969)

The satisfiability problem of Monadic Second-Order (MSO) logic is decidable over infinite trees.

Specifying properties of trees

- FO, CTL*, ...
- modal μ-calculus
- finite automata
- Monadic Second-Order logic

Theorem (Rabin 1969)

The satisfiability problem of Monadic Second-Order (MSO) logic is decidable over infinite trees.

Translate formulæ into automata:

Specifying properties of trees

- FO, CTL*, ...
- modal μ-calculus
- finite automata
- Monadic Second-Order logic

Theorem (Rabin 1969)

The satisfiability problem of Monadic Second-Order (MSO) logic is decidable over infinite trees.

Translate formulæ into automata:
nondeterministic and alternating (deterministic are **not** enough)

How complex is a given property?

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How many set quantifiers a language requires?

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How many set quantifiers a language requires?
$\rightsquigarrow$ at most two

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How many set quantifiers a language requires?
$\rightsquigarrow$ at most two

When only one is enough?
???

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How many set quantifiers a language requires?
$\rightsquigarrow$ at most two

When only one is enough?
???

Is a given language Borel?
???

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How many set quantifiers a language requires?
$\rightsquigarrow$ at most two

When only one is enough?
???

Is a given language Borel?
???

Number of alternations of μ / ν operators?

$$\mu X.\ \nu Y.\ (a \wedge \Box X) \vee (\neg a \wedge \Box Y)$$

How complex is a given property?

Empty?
$\rightsquigarrow$ Rabin's theorem

How many set quantifiers a language requires?
$\rightsquigarrow$ at most two

When only one is enough?
???

Is a given language Borel?
???

Number of alternations of μ / ν operators?

$$\mu X.\ \nu Y.\ (a \wedge \Box X) \vee (\neg a \wedge \Box Y)$$

???

Rabin-Mostowski index problem

Parity automata

Priorities $\Omega\colon Q \to \mathbb{N}$

Rabin-Mostowski index problem

Parity automata

Priorities $\Omega \colon Q \to \mathbb{N}$

A sequence $q_0, q_1, q_2, \ldots$ is accepting if

the highest value $\Omega(q_i)$ occurring infinitely often is even

Rabin-Mostowski index problem

Parity automata

Priorities $\Omega\colon Q \to \mathbb{N}$

A sequence $q_0, q_1, q_2, \ldots$ is accepting if

the highest value $\Omega(q_i)$ occurring infinitely often is even

index of $\mathcal{A} = (\min \Omega, \max \Omega)$ — range of priorities

Rabin-Mostowski index problem

Parity automata

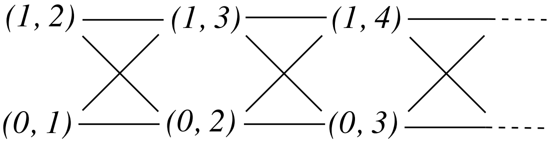
Priorities $\Omega\colon Q \to \mathbb{N}$

A sequence $q_0, q_1, q_2, \ldots$ is accepting if

the highest value $\Omega(q_i)$ occurring infinitely often is even

index of $\mathcal{A} = (\min\Omega, \max\Omega)$ — range of priorities

Rabin-Mostowski index hierarchy



Rabin-Mostowski index problem

Parity automata

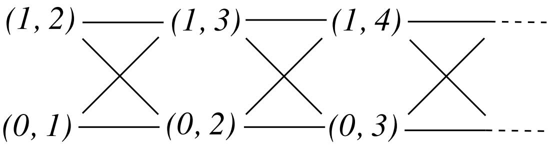
Priorities $\Omega\colon Q \to \mathbb{N}$

A sequence $q_0, q_1, q_2, \ldots$ is accepting if

the highest value $\Omega(q_i)$ occurring infinitely often is even

index of $\mathcal{A} = (\min \Omega, \max \Omega)$ — range of priorities

Rabin-Mostowski index hierarchy



$\rightsquigarrow$ alternation of fixpoints in modal μ-calculus

Infinite trees are difficult

Only few effective characterizations:

- Boolean combinations of open sets [Bojańczyk, Place '12]
- nondetereministic $(0, 1)$-automata [Colcombet, Löding]
- weakness for $(1, 2)$-automata [Kuperberg, Vanden Boom '11]

Infinite trees are difficult

Only few effective characterizations:

- Boolean combinations of open sets [Bojańczyk, Place '12]
- nondetereministic $(0, 1)$-automata [Colcombet, Löding]
- weakness for $(1, 2)$-automata [Kuperberg, Vanden Boom '11]

Approach: solve problems for "easier" subclasses

- all hierarchies decidable for deterministic automata [Niwiński, Walukiewicz '03, '05; M '05, '06, '08]
- weak index decidable for weak game automata [Duparc, F, M '11]

Infinite trees are difficult

Only few effective characterizations:

- Boolean combinations of open sets [Bojańczyk, Place '12]
- nondetereministic $(0, 1)$-automata [Colcombet, Löding]
- weakness for $(1, 2)$-automata [Kuperberg, Vanden Boom '11]

Approach: solve problems for “easier” subclasses

- all hierarchies decidable for deterministic automata [Niwiński, Walukiewicz '03, '05; M '05, '06, '08]
- weak index decidable for weak game automata [Duparc, F, M '11]

This work:
Rabin-Mostowski index problem for game automata

Index problem(s)

Given an automaton $\mathcal{A}$,
what is the minimal index of an automaton recognizng $\mathrm{L}(\mathcal{A})$?

Index problem(s)

Given an automaton $\mathcal{A}$,
what is the minimal index of an automaton recognizng $\mathrm{L}(\mathcal{A})$?

nondeterministic or alternating

Index problem(s)

Given an automaton $\mathcal{A}$,
what is the minimal index of an automaton recognizng $\mathrm{L}(\mathcal{A})$?

input	**nondeterministic index**	**alternating index**
nondet.	?	?

Index problem(s)

Given an automaton $\mathcal{A}$,
what is the minimal index of an automaton recognizng $\mathrm{L}(\mathcal{A})$?

input	**nondeterministic index**	**alternating index**
nondet.	?	?
deterministic	arbitrarily high, decidable [Niwiński, Walukiewicz '04]	always $(0, 1)$

Index problem(s)

Given an automaton $\mathcal{A}$,
what is the minimal index of an automaton recognizng $\mathrm{L}(\mathcal{A})$?

input	**nondeterministic index**	**alternating index**
nondet.	?	?
game	arb. high, **decidable**	arb. high, **decidable**
deterministic	arbitrarily high, decidable [Niwiński, Walukiewicz '04]	always $(0,1)$

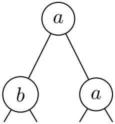
Alternating tree automata

Alternating tree automata

Semantics via games: two opponents $\exists$ and $\forall$

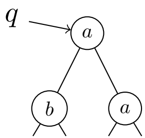
Alternating tree automata

Semantics via games: two opponents $\exists$ and $\forall$



Alternating tree automata

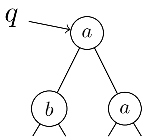
Semantics via games: two opponents $\exists$ and $\forall$



Alternating tree automata

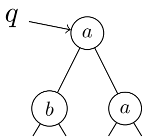
Semantics via games: two opponents $\exists$ and $\forall$

$$q \xrightarrow{a} (s, \mathbf{L}) \;\vee\; (p, \mathbf{L}) \wedge (r, \mathbf{R})$$



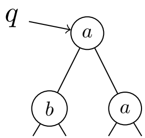
Semantics via games: two opponents ∃ and ∀

$$q \xrightarrow{a} \underbrace{(s, \mathbf{L}) \ \vee \ (p, \mathbf{L}) \wedge (r, \mathbf{R})}_{\text{positive boolean combination of } (q,\mathbf{L}) \text{ and } (q,\mathbf{R})}$$



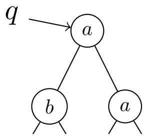
Semantics via games: two opponents $\exists$ and $\forall$

$$q \xrightarrow{a} \underbrace{(s, \mathbf{L}) \;\vee\; (p, \mathbf{L}) \wedge (r, \mathbf{R})}_{\exists \text{ chooses}}$$



Semantics via games: two opponents $\exists$ and $\forall$

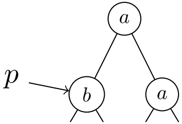
$$q \xrightarrow{a} (s, \mathbf{L}) \;\vee\; \underbrace{(p, \mathbf{L}) \wedge (r, \mathbf{R})}_{\forall \text{ chooses}}$$



Alternating tree automata

Semantics via games: two opponents $\exists$ and $\forall$

$$q \xrightarrow{a} (s, \mathbf{L}) \;\vee\; (p, \mathbf{L}) \wedge (r, \mathbf{R})$$



Alternating tree automata

Semantics via games: two opponents $\exists$ and $\forall$

$$q \xrightarrow{a} \underbrace{(s,\mathbf{L}) \;\vee\; (p,\mathbf{L}) \wedge (r,\mathbf{R})}_{\text{alternating transitions}}$$

$\exists$ wins an infinite play if the parity condition is satisfied

Alternating tree automata

Semantics via games: two opponents $\exists$ and $\forall$

$$q \xrightarrow{a} \underbrace{(s, \mathbf{L}) \;\vee\; (p, \mathbf{L}) \wedge (r, \mathbf{R})}_{\text{alternating transitions}}$$

$\exists$ wins an infinite play if the parity condition is satisfied

A tree t is accepted if $\exists$ has a winning strategy

Game automata

Deterministic automata made symmetric with respect to alternation:

$$\frac{\text{game}}{\text{alternating}} = \frac{\text{deterministic}}{\text{nondeterminsitic}}$$

Deterministic automata made symmetric with respect to alternation:

$$\frac{\text{game}}{\text{alternating}} = \frac{\text{deterministic}}{\text{nondeterminsitic}}$$

Deterministic automata:

$$q \xrightarrow{a} (q_L, \mathbf{L}) \wedge (q_R, \mathbf{R})$$

Game automata

Deterministic automata made symmetric with respect to alternation:

$$\frac{\text{game}}{\text{alternating}} = \frac{\text{deterministic}}{\text{nondeterminsitic}}$$

Deterministic automata:

$$q \xrightarrow{a} (q_L, \mathbf{L}) \wedge (q_R, \mathbf{R})$$

Game automata:

$$q \xrightarrow{a} (q_L, \mathbf{L}) \wedge (q_R, \mathbf{R}) \quad \text{— } \forall\text{'s choice}$$

$$q \xrightarrow{a} (q_L, \mathbf{L}) \vee (q_R, \mathbf{R}) \quad \text{— } \exists\text{'s choice}$$

Game automata

Deterministic automata made symmetric with respect to alternation:

$$\frac{\text{game}}{\text{alternating}} = \frac{\text{deterministic}}{\text{nondeterminsitic}}$$

Deterministic automata:

$$q \xrightarrow{a} (q_L, \mathbf{L}) \wedge (q_R, \mathbf{R})$$

Game automata:

$$q \xrightarrow{a} (q_L, \mathbf{L}) \wedge (q_R, \mathbf{R}) \quad \text{— } \forall\text{'s choice}$$

$$q \xrightarrow{a} (q_L, \mathbf{L}) \vee (q_R, \mathbf{R}) \quad \text{— } \exists\text{'s choice}$$

$\rightsquigarrow$ every node of a tree can be reached in exactly one state

Properties of game automata

Properties of game automata

- Contain deterministic automata and their complements;

Properties of game automata

- Contain deterministic automata and their complements;
- closed under complementation;

Properties of game automata

- Contain deterministic automata and their complements;
- closed under complementation;
- **not** closed under union nor intersection;

Properties of game automata

- Contain deterministic automata and their complements;
- closed under complementation;
- **not** closed under union nor intersection;
- recognize languages arbitrarily high in both hierarchies.

Our results

Our results

- Effective characterization of game languages

Our results

- Effective characterization of game languages
- Nondeterministic index problem for game languages

Our results

- Effective characterization of game languages
- Nondeterministic index problem for game languages
- Alternating index problem for game languages

Theorem (Characterisation of game languages)

Given a tree automaton $\mathcal{A}$ we can decide if $\mathrm{L}(\mathcal{A})$ is recognised by any game automaton.

Theorem (Characterisation of game languages)

Given a tree automaton $\mathcal{A}$ we can decide if $\mathrm{L}(\mathcal{A})$ is recognised by any game automaton.

Following ideas of Niwiński and Walukiewicz.

Theorem (Characterisation of game languages)

Given a tree automaton $\mathcal{A}$ we can decide if $\mathrm{L}(\mathcal{A})$ is recognised by any game automaton.

Following ideas of Niwiński and Walukiewicz.

A game language has to be:

Theorem (Characterisation of game languages)

Given a tree automaton $\mathcal{A}$ we can decide if $\mathrm{L}(\mathcal{A})$ is recognised by any game automaton.

Following ideas of Niwiński and Walukiewicz.

A game language has to be:

- locally game — locally it looks like a disjunction ($\vee$) or a conjunction ($\wedge$) of a pair of languages

Theorem (Characterisation of game languages)

Given a tree automaton $\mathcal{A}$ we can decide if $\mathrm{L}(\mathcal{A})$ is recognised by any game automaton.

Following ideas of Niwiński and Walukiewicz.

A game language has to be:

- locally game — locally it looks like a disjunction ($\vee$) or a conjunction ($\wedge$) of a pair of languages
- pathwise game — every infinite branch is itself winning or loosing

Theorem (Characterisation of game languages)

Given a tree automaton $\mathcal{A}$ we can decide if $\mathrm{L}(\mathcal{A})$ is recognised by any game automaton.

Following ideas of Niwiński and Walukiewicz.

A game language has to be:

- locally game — locally it looks like a disjunction ($\vee$) or a conjunction ($\wedge$) of a pair of languages
- pathwise game — every infinite branch is itself winning or loosing

(Composition method)

Theorem (Nondeterministic index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of a nondeterministic automaton recognising $\mathrm{L}(\mathcal{B})$.

Theorem (Nondeterministic index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of a nondeterministic automaton recognising $\mathrm{L}(\mathcal{B})$.

Proof.

By a reduction to the deterministic case.

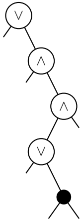
Theorem (Nondeterministic index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of a nondeterministic automaton recognising $\mathrm{L}(\mathcal{B})$.

Proof.

By a reduction to the deterministic case.

We can enforce players to pick particular directions in a tree.



■

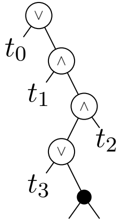
Theorem (Nondeterministic index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of a nondeterministic automaton recognising $\mathrm{L}(\mathcal{B})$.

Proof.

By a reduction to the deterministic case.

We can enforce players to pick particular directions in a tree.



■

Theorem (Alternating index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of an alternating automaton recognising $\mathrm{L}(\mathcal{B})$.

Theorem (Alternating index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of an alternating automaton recognising $\mathrm{L}(\mathcal{B})$.

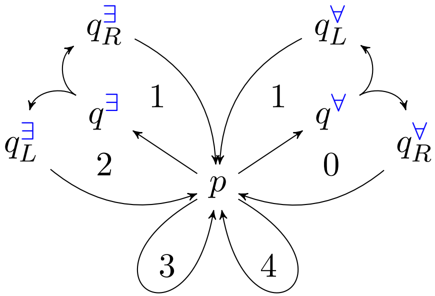
The most involved reasoning

Theorem (Alternating index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of an alternating automaton recognising $\mathrm{L}(\mathcal{B})$.

The most involved reasoning

Determining hard gadgets



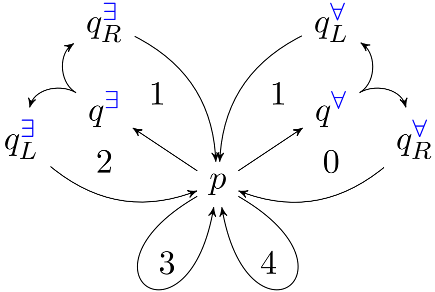
Theorem (Alternating index problem)

Given a game automaton $\mathcal{B}$ we can compute the minimal index of an alternating automaton recognising $\mathrm{L}(\mathcal{B})$.

The most involved reasoning

Determining hard gadgets

Using tools of topology



Two bounds

Lower bound

If $\mathcal{A}$ contains a hard gadget then $\mathrm{L}(\mathcal{A})$ is hard.

Two bounds

Lower bound

If $\mathcal{A}$ contains a hard gadget then $\mathrm{L}(\mathcal{A})$ is hard.

Use topological methods and enforcing.

Two bounds

Lower bound

If $\mathcal{A}$ contains a hard gadget then $\mathrm{L}(\mathcal{A})$ is hard.

Use topological methods and enforcing.

Upper bound

If $\mathcal{A}$ does **not** contain a hard gadget then show that $\mathrm{L}(\mathcal{A})$ is **not** hard.

Two bounds

Lower bound

If $\mathcal{A}$ contains a hard gadget then $\mathrm{L}(\mathcal{A})$ is hard.

Use topological methods and enforcing.

Upper bound

If $\mathcal{A}$ does **not** contain a hard gadget then show that $\mathrm{L}(\mathcal{A})$ is **not** hard.

Inductively construct a *simple* automaton recognising $\mathrm{L}(\mathcal{A})$.

Rabin-Mostowski index problem

input	nondeterministic index	alternating index
nondet.	?	?
game	arb. high, **decidable**	arb. high, **decidable**
deterministic	arbitrarily high, decidable [Niwiński, Walukiewicz '04]	always $(0, 1)$

Rabin-Mostowski index problem

input	**nondeterministic index**	**alternating index**
nondet.	?	?
game	arb. high, **decidable**	arb. high, **decidable**
deterministic	arbitrarily high, decidable [Niwiński, Walukiewicz '04]	always $(0, 1)$

Game automata are as manageable as deterministic ones.