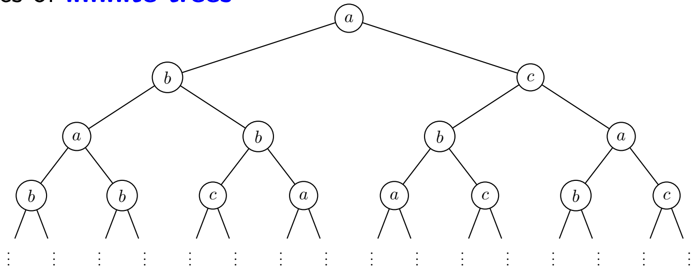
Unambiguous Büchi is weak

Henryk Michalewski Michał Skrzypczak

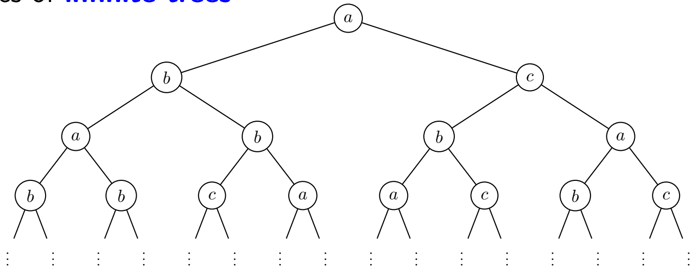
DLT 2016
Montreal

Languages of **infinite trees**

Languages of **infinite trees**

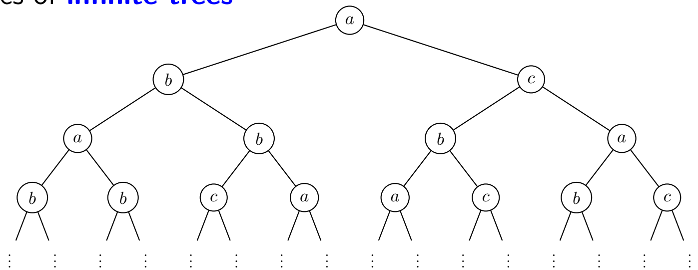


Languages of **infinite trees**



Definable in **Monadic Second-Order logic** (MSO)

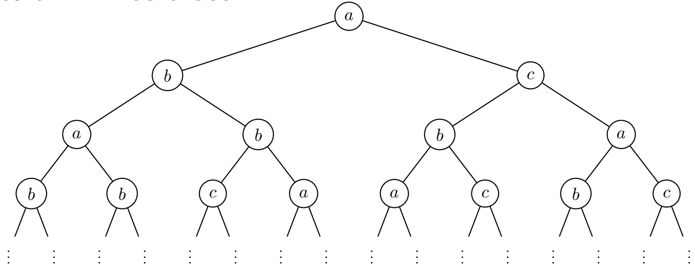
Languages of **infinite trees**



Definable in **Monadic Second-Order logic** (MSO)

$\neg, \ \vee, \ \exists_x, \ \exists_X, \ x \in X, \ x = y$

Languages of **infinite trees**



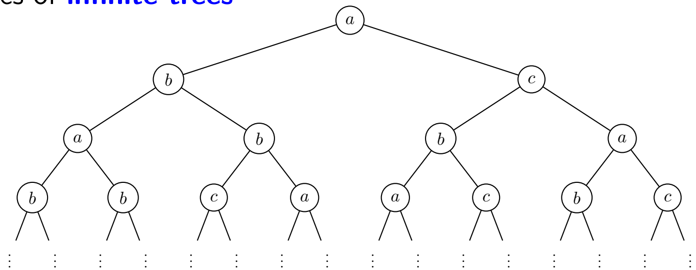
Definable in **Monadic Second-Order logic** (MSO)

$\neg,\ \vee,\ \exists_x,\ \exists_X,\ x \in X,\ x = y$

Theorem (Rabin 1969)

The MSO theory of $\big(\{\mathtt{L},\mathtt{R}\}^*, \mathrm{s}_{\mathtt{L}}, \mathrm{s}_{\mathtt{R}}\big)$ is decidable.

Languages of **infinite trees**



Definable in **Monadic Second-Order logic** (MSO)

$\neg,\ \vee,\ \exists_x,\ \exists_X,\ x \in X,\ x = y$

Theorem (Rabin 1969)

The MSO theory of $\left(\{\mathtt{L},\mathtt{R}\}^*, \mathrm{s}_{\mathtt{L}}, \mathrm{s}_{\mathtt{R}}\right)$ is decidable.

Proof

Automata...

Parity automata

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

$$\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$$

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

$$\limsup_{n \to \infty} \Omega(q_n) \equiv 0 \pmod 2$$

Translations

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

$$\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$$

Translations

Monadic Second-order logic

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

$$\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$$

Translations

Monadic Second-order logic

$$\cong$$

alternating parity automata

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i{+}1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

$$\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$$

Translations

Monadic Second-order logic

$\wr\!\!\|$

alternating parity automata

$\wr\!\!\|$

non-deterministic parity automata

Parity automata

Every state $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$

Sequence $q_0, q_1, q_2 \ldots$ is accepting if

$$\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$$

Translations

Monadic Second-order logic

$\cong$

alternating parity automata

$\cong$

non-deterministic parity automata

$\supsetneq$

deterministic parity automata

Non-deterministic automata

Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

the **run** of $\mathcal{A}$ on t is accepting

Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

the **run** of $\mathcal{A}$ on t is accepting

A non-deterministic automaton $\mathcal{A}$ accepts a tree t if

Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

the **run** of $\mathcal{A}$ on t is accepting

A non-deterministic automaton $\mathcal{A}$ accepts a tree t if

there exists a **run** of $\mathcal{A}$ on t that is accepting

Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

the **run** of $\mathcal{A}$ on t is accepting

A non-deterministic automaton $\mathcal{A}$ accepts a tree t if

there exists a **run** of $\mathcal{A}$ on t that is accepting

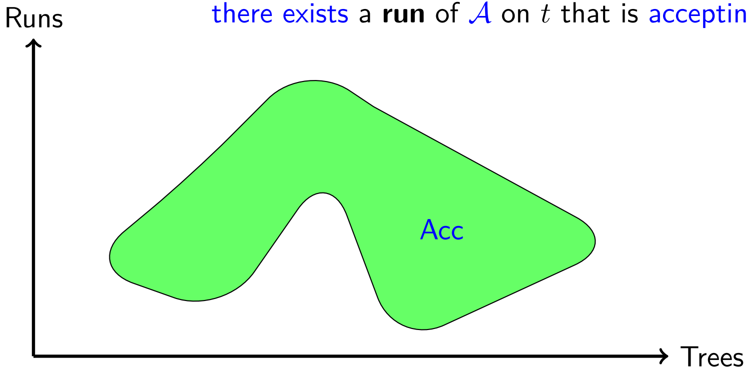
Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

the **run** of $\mathcal{A}$ on t is accepting

A non-deterministic automaton $\mathcal{A}$ accepts a tree t if

there exists a **run** of $\mathcal{A}$ on t that is accepting



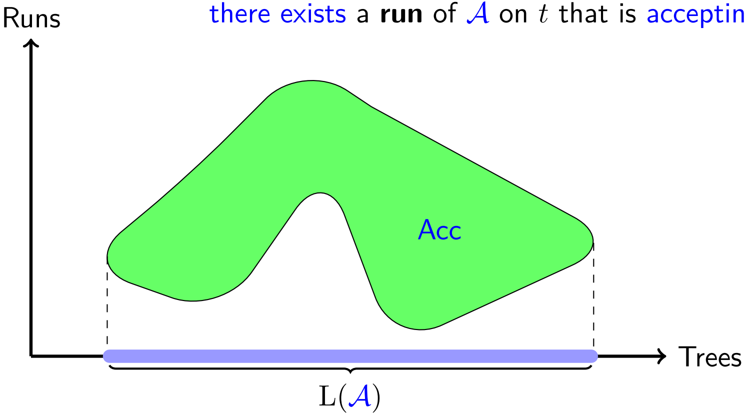
Non-deterministic automata

A deterministic automaton $\mathcal{A}$ accepts a tree t if

the **run** of $\mathcal{A}$ on t is accepting

A non-deterministic automaton $\mathcal{A}$ accepts a tree t if

there exists a **run** of $\mathcal{A}$ on t that is accepting



Unambiguous automata

Unambiguous automata

A non-deterministic automaton $\mathcal{A}$ is unambiguous if

Unambiguous automata

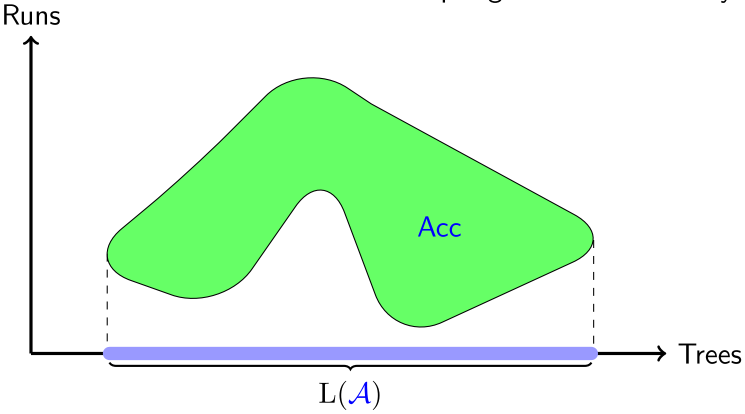
A non-deterministic automaton $\mathcal{A}$ is unambiguous if

there is at most one accepting **run** of $\mathcal{A}$ on every input

Unambiguous automata

A non-deterministic automaton $\mathcal{A}$ is unambiguous if

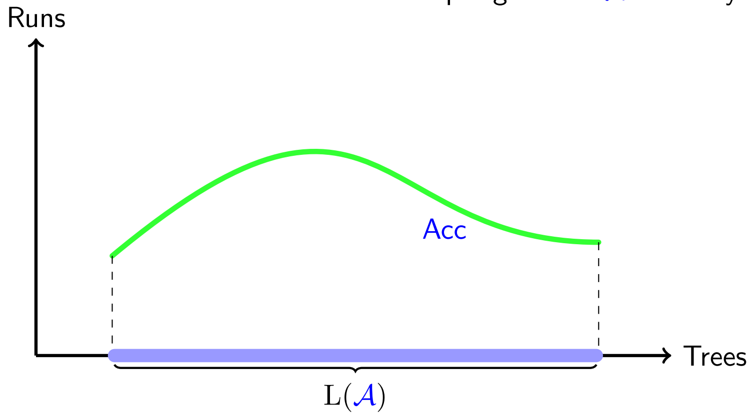
there is at most one accepting **run** of $\mathcal{A}$ on every input



Unambiguous automata

A non-deterministic automaton $\mathcal{A}$ is unambiguous if

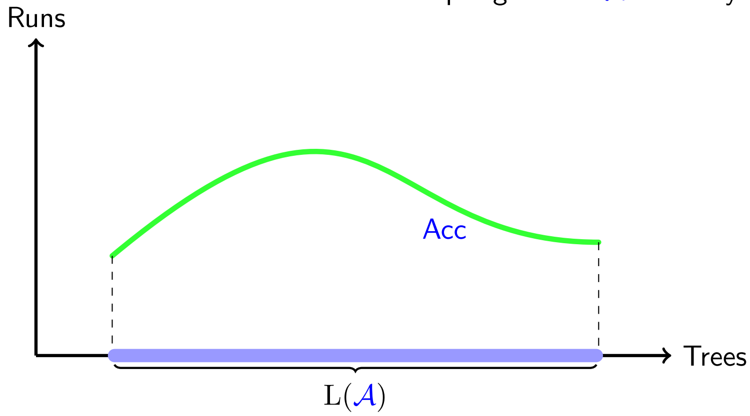
there is at most one accepting **run** of $\mathcal{A}$ on every input



Unambiguous automata

A non-deterministic automaton $\mathcal{A}$ is unambiguous if

there is at most one accepting **run** of $\mathcal{A}$ on every input

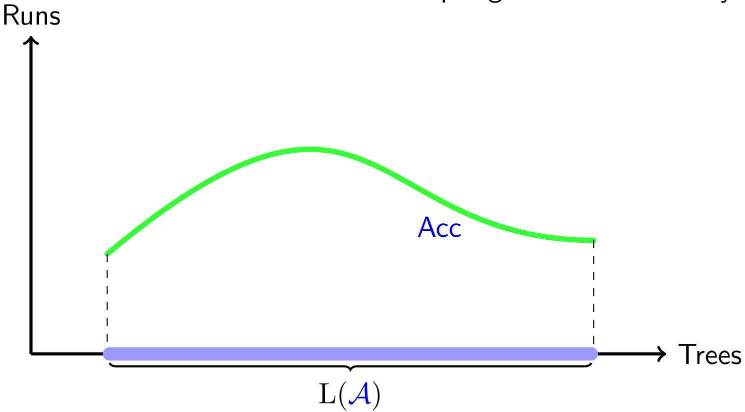


⇝ efficient algorithms for unambiguous automata

Unambiguous automata

A non-deterministic automaton $\mathcal{A}$ is unambiguous if

there is at most one accepting **run** of $\mathcal{A}$ on every input



⇝ efficient algorithms for unambiguous automata

(e.g. Stearns, Hunt [1985])

Expressive power

Expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a

Expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a

cannot be recognised by an unambiguous automaton.

Expressive power

Theorem (Niwiński, Walukiewicz [1996])
The language of trees that contain a letter a **cannot** be recognised by an unambiguous automaton.

non-deterministic automata

$\not\supseteq$

unambiguous automata

$\not\supseteq$

deterministic automata

Expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a

cannot be recognised by an unambiguous automaton.

non-deterministic automata

$\not\supseteq$

unambiguous automata

$\not\supseteq$

deterministic automata

Expressive power

Theorem (Niwiński, Walukiewicz [1996])
The language of trees that contain a letter a
cannot be recognised by an unambiguous automaton.

non-deterministic automata

$\not\supseteq$

unambiguous automata

$\not\supseteq$

deterministic automata

Expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a

cannot be recognised by an unambiguous automaton.

non-deterministic automata

? $\not\supseteq$ **?**

unambiguous automata

$\not\supseteq$

deterministic automata

Expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a

cannot be recognised by an unambiguous automaton.

non-deterministic automata

? $\subsetneq$ **?**

unambiguous automata

$\subsetneq$

deterministic automata

$\rightsquigarrow$ **Goal**: understand unambiguous languages of infinite trees

Questions about unambiguity

Questions about unambiguity

— which languages are unambiguous **?**

Questions about unambiguity

— which languages are unambiguous ? (i.e. effective characterisation)

Questions about unambiguity

— which languages are unambiguous **?** (i.e. effective characterisation)

— are there unambiguous languages that are complex **?**

Questions about unambiguity

— which languages are unambiguous ? (i.e. effective characterisation)

— are there unambiguous languages that are complex ?

— how to use the assumption that $\mathcal{A}$ is unambiguous ?

Questions about unambiguity

— which languages are unambiguous ? (i.e. effective characterisation)

$\rightsquigarrow$ OPEN

— are there unambiguous languages that are complex ?

— how to use the assumption that $\mathcal{A}$ is unambiguous ?

Questions about unambiguity

— which languages are unambiguous ? (i.e. effective characterisation)

$\rightsquigarrow$ OPEN

— are there unambiguous languages that are complex ?

$\rightsquigarrow$ (Hummel [2012]): there are unambiguous languages outside $\sigma(\mathbf{\Sigma}_1^1)$

— how to use the assumption that $\mathcal{A}$ is unambiguous ?

Questions about unambiguity

— which languages are unambiguous **?** (i.e. effective characterisation)

$\rightsquigarrow$ OPEN

— are there unambiguous languages that are complex **?**

$\rightsquigarrow$ (Hummel [2012]): there are unambiguous languages outside $\sigma(\mathbf{\Sigma}^1_1)$

— how to use the assumption that $\mathcal{A}$ is unambiguous **?**

$\rightsquigarrow$ **this work**

Index hierarchy (alternating one)

Index hierarchy (alternating one)

$$\left[\begin{array}{c}\text{parity automaton: every } q \in Q \text{ has a priority } \Omega(q) \in \{i, i{+}1, \ldots, j\} \\ \text{sequence } q_0, q_1, \ldots \text{ is accepting if } \limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2\end{array}\right]$$

Index hierarchy (alternating one)

$$\left[\begin{array}{c}\text{parity automaton: every } q \in Q \text{ has a priority } \Omega(q) \in \{i, i{+}1, \ldots, j\} \\ \text{sequence } q_0, q_1, \ldots \text{ is accepting if } \limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2\end{array}\right]$$

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

Index hierarchy (alternating one)

$$\left[\begin{array}{c}\text{parity automaton: every } q\in Q \text{ has a priority } \Omega(q)\in\{i,i{+}1,\ldots,j\}\\ \text{sequence } q_0,q_1,\ldots \text{ is accepting if } \limsup_{n\to\infty}\Omega(q_n)\equiv 0 \pmod 2\end{array}\right]$$

$\mathcal{A}$ has index (i,j) if $\Omega\colon Q\to\{i,\ldots,j\}$

L has index (i,j) if $L=\mathrm{L}(\mathcal{A})$ of index $(i.j)$

Index hierarchy (alternating one)

[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i{+}1, \dots, j\}$
sequence $q_0, q_1, \dots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \dots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

[w.l.o.g. we can assume that $i \in \{0, 1\}$]

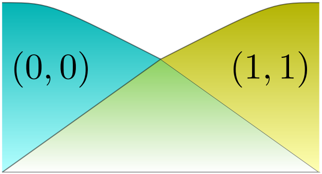
Index hierarchy (alternating one)

$$\left[\begin{array}{c}\text{parity automaton: every } q \in Q \text{ has a priority } \Omega(q) \in \{i, i{+}1, \ldots, j\} \\ \text{sequence } q_0, q_1, \ldots \text{ is accepting if } \limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2\end{array}\right]$$

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

[w.l.o.g. we can assume that $i \in \{0, 1\}$]



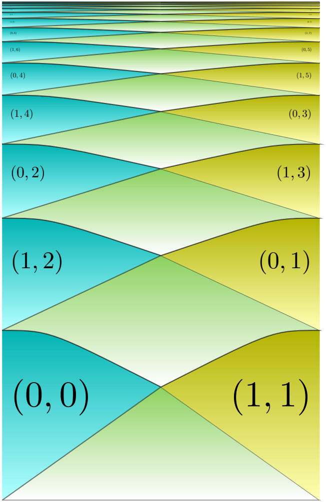
Index hierarchy (alternating one)

[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i{+}1, \ldots, j\}$
sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

[w.l.o.g. we can assume that $i \in \{0, 1\}$]



Index hierarchy (alternating one)

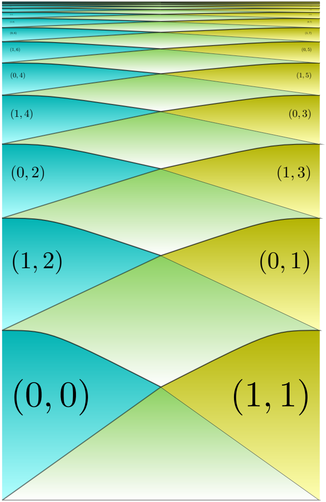
[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i{+}1, \ldots, j\}$
sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

[w.l.o.g. we can assume that $i \in \{0, 1\}$]

Theorem For modal μ-calculus:



Index hierarchy (alternating one)

[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i{+}1, \ldots, j\}$
sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

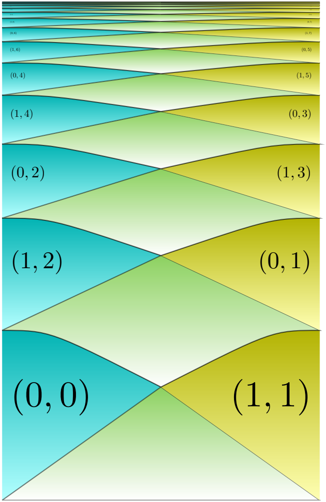
$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

[w.l.o.g. we can assume that $i \in \{0, 1\}$]

Theorem For modal μ-calculus:

Index hierarchy ≅ Alternation depth hierarchy



Index hierarchy (alternating one)

[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$
sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

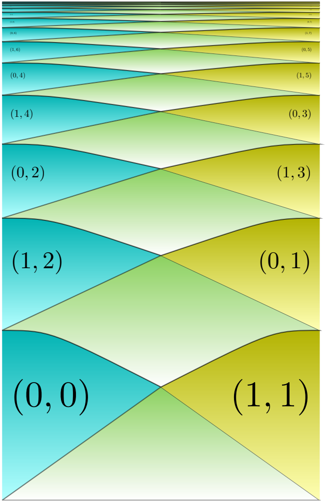
[w.l.o.g. we can assume that $i \in \{0, 1\}$]

Theorem For modal μ-calculus:

Index hierarchy ≅ Alternation depth hierarchy

Theorem (Bradfield [1998], Arnold [1999])

There are languages requiring **big** indices.



Index hierarchy (alternating one)

[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i{+}1, \ldots, j\}$
sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

L has index (i, j) if $L = \mathrm{L}(\mathcal{A})$ of index $(i.j)$

[w.l.o.g. we can assume that $i \in \{0, 1\}$]

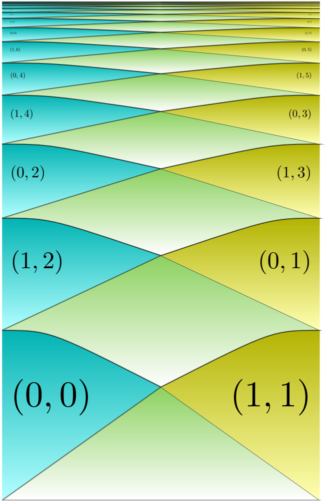
Theorem For modal μ-calculus:

Index hierarchy ≅ Alternation depth hierarchy

Theorem (Bradfield [1998], Arnold [1999])

There are languages requiring **big** indices.

(i.e. the hierarchies are strict)



Index hierarchy — Comp classes

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

strongly connected components of $\mathcal{A}$

have indices (i, j) and $(i{+}1, j{+}1)$

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

strongly connected components of $\mathcal{A}$

have indices (i, j) and $(i{+}1, j{+}1)$

$\rightsquigarrow$ corresponds to composition of formulæ

in modal μ-calculus

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i,j)$ if

strongly connected components of $\mathcal{A}$

have indices (i,j) and $(i{+}1,j{+}1)$

$\rightsquigarrow$ corresponds to composition of formulæ

in modal μ-calculus

Theorem (Rabin [1970])

$\mathrm{Comp}(0,0) \cong (0,1) \cap (1,2) \cong$ weak MSO

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

strongly connected components of $\mathcal{A}$

have indices (i, j) and $(i{+}1, j{+}1)$

$\rightsquigarrow$ corresponds to composition of formulæ

in modal μ-calculus

Theorem (Rabin [1970])

$\mathrm{Comp}(0, 0) \cong (0, 1) \cap (1, 2) \cong$ weak MSO

Theorem (Arnold, Santocanale [2005])

$\mathrm{Comp}(i, j) \subsetneq (i, j{+}1) \cap (i{+}1, j{+}2)$ for $i < j$

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

strongly connected components of $\mathcal{A}$

have indices (i, j) and $(i{+}1, j{+}1)$

$\rightsquigarrow$ corresponds to composition of formulæ

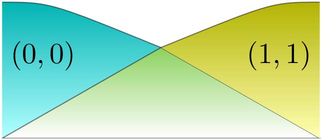
in modal μ-calculus

Theorem (Rabin [1970])

$\mathrm{Comp}(0, 0) \cong (0, 1) \cap (1, 2) \cong$ weak MSO

Theorem (Arnold, Santocanale [2005])

$\mathrm{Comp}(i, j) \subsetneq (i, j{+}1) \cap (i{+}1, j{+}2)$ for $i < j$



Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i,j)$ if

strongly connected components of $\mathcal{A}$

have indices (i,j) and $(i{+}1, j{+}1)$

$\rightsquigarrow$ corresponds to composition of formulæ

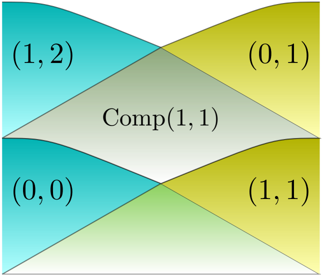
in modal μ-calculus

Theorem (Rabin [1970])

$\mathrm{Comp}(0,0) \cong (0,1) \cap (1,2) \cong$ weak MSO

Theorem (Arnold, Santocanale [2005])

$\mathrm{Comp}(i,j) \subsetneq (i,j{+}1) \cap (i{+}1, j{+}2)$ for $i < j$



Index hierarchy — Comp classes

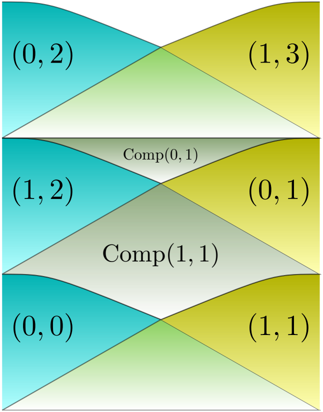
$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

strongly connected components of $\mathcal{A}$

have indices (i, j) and $(i+1, j+1)$

$\rightsquigarrow$ corresponds to composition of formulæ

in modal μ-calculus



Theorem (Rabin [1970])

$\mathrm{Comp}(0, 0) \cong (0, 1) \cap (1, 2) \cong$ weak MSO

Theorem (Arnold, Santocanale [2005])

$\mathrm{Comp}(i, j) \subsetneq (i, j+1) \cap (i+1, j+2)$ for $i < j$

Index hierarchy — Comp classes

$\mathcal{A}$ is $\mathrm{Comp}(i, j)$ if

strongly connected components of $\mathcal{A}$

have indices (i, j) and $(i+1, j+1)$

$\rightsquigarrow$ corresponds to composition of formulæ

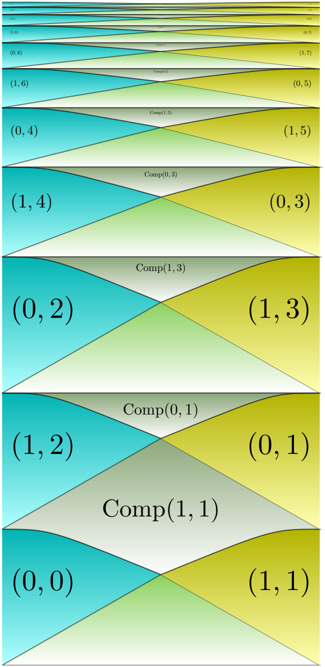
in modal μ-calculus

Theorem (Rabin [1970])

$\mathrm{Comp}(0, 0) \cong (0, 1) \cap (1, 2) \cong$ weak MSO

Theorem (Arnold, Santocanale [2005])

$\mathrm{Comp}(i, j) \subsetneq (i, j+1) \cap (i+1, j+2)$ for $i < j$



Our result

Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

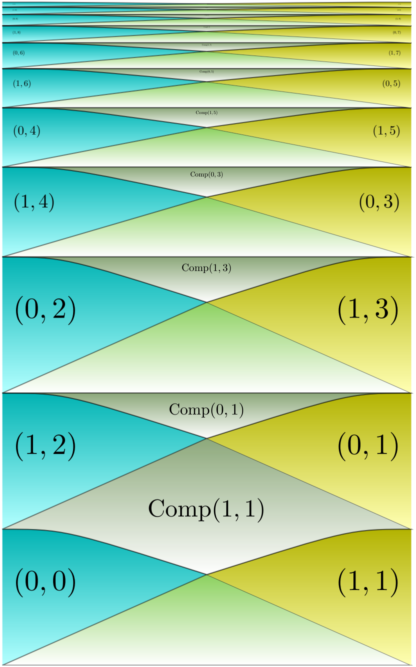
then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.

Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.



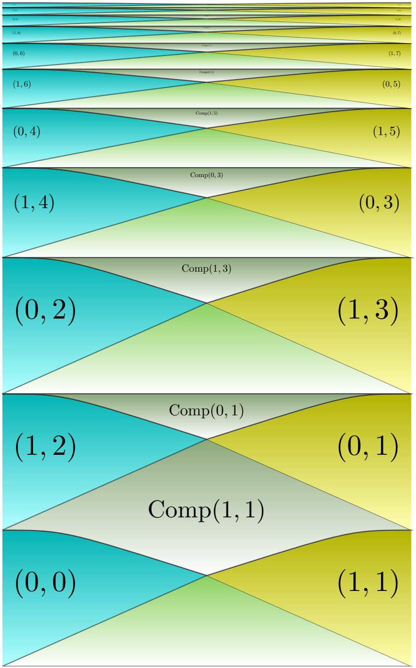
Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.

Büchi ⟶



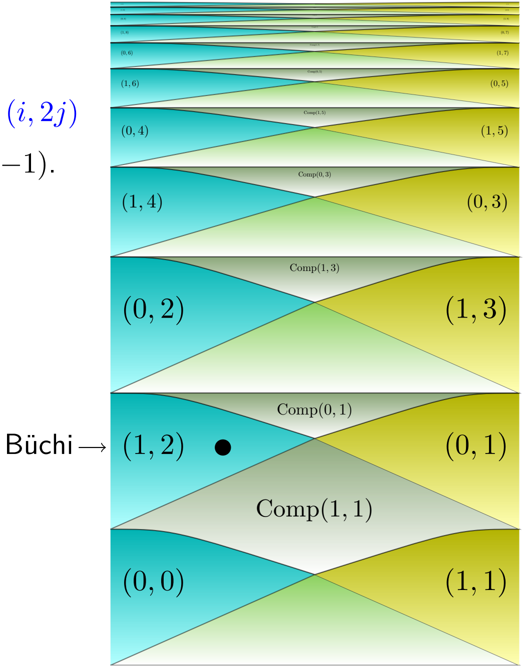
Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j{-}1)$.

I.e. if $\mathcal{A}$ is Büchi



Our result

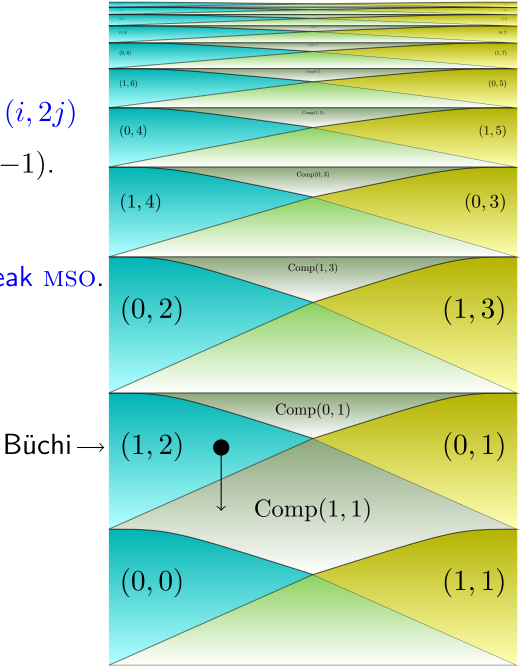
Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.

I.e. if $\mathcal{A}$ is Büchi

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(1, 1) \cong$ weak MSO.



Our result

Theorem (Michalewski, S.)

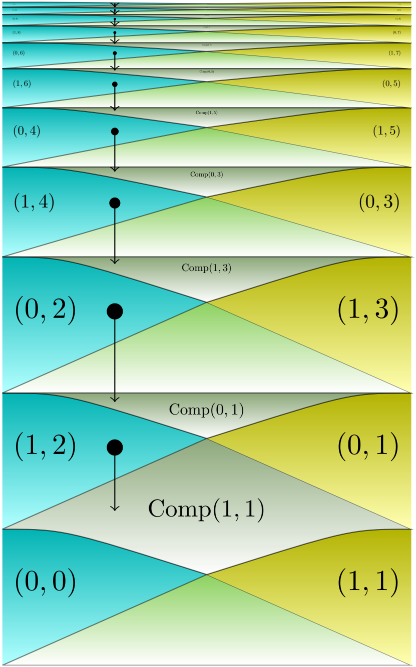
If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.

I.e. if $\mathcal{A}$ is Büchi

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(1, 1) \cong$ weak MSO.

Büchi ⟶



Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

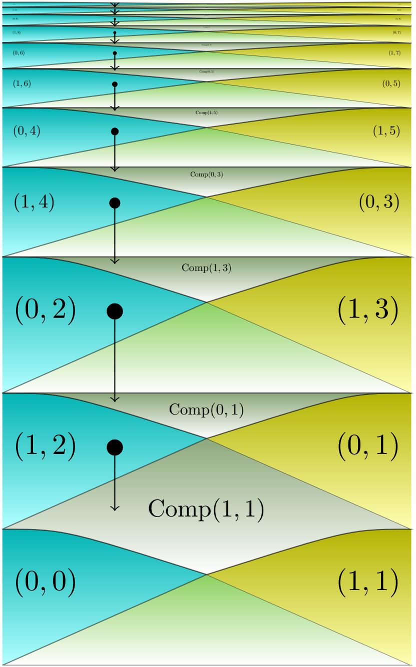
then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j{-}1)$.

I.e. if $\mathcal{A}$ is Büchi

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(1,1) \cong$ weak MSO.

Moreover, we can construct

Büchi ⟶



Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

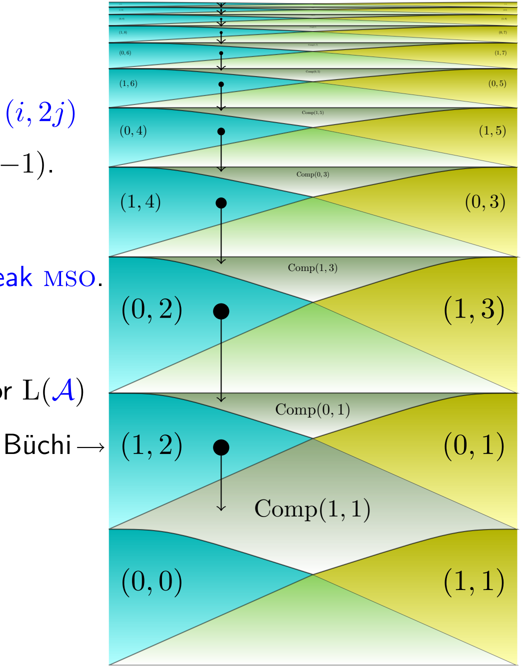
then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.

I.e. if $\mathcal{A}$ is Büchi

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(1, 1) \cong$ weak MSO.

Moreover, we can construct

$\mathrm{Comp}(i, 2j-1)$ automaton for $\mathrm{L}(\mathcal{A})$



Our result

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j-1)$.

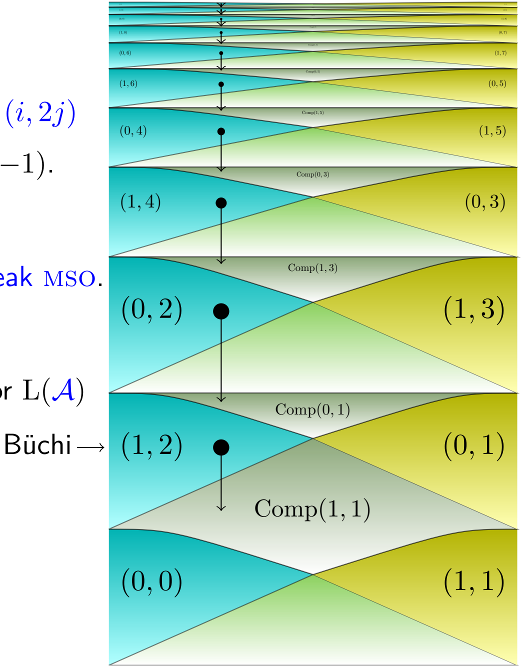
I.e. if $\mathcal{A}$ is Büchi

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(1, 1) \cong$ weak MSO.

Moreover, we can construct

$\mathrm{Comp}(i, 2j-1)$ automaton for $\mathrm{L}(\mathcal{A})$

of size polynomial in $\mathcal{A}$.



Summary

Summary

— the first parity index collapse

based on unambiguity

Summary

— the first parity index collapse

based on unambiguity

— efficient transformation algorithm

Summary

— the first parity index collapse

based on unambiguity

— efficient transformation algorithm

— insights into expressive power

of unambiguous automata

Summary

— the first parity index collapse

based on unambiguity

— efficient transformation algorithm

— insights into expressive power

of unambiguous automata

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

Summary

— the first parity index collapse
based on unambiguity

— efficient transformation algorithm

— insights into expressive power
of unambiguous automata

Theorem (Michalewski, S.)
If $\mathcal{A}$ is unambiguous of index $(i, 2j)$
then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j{-}1)$.

Summary

— the first parity index collapse

based on unambiguity

— efficient transformation algorithm

— insights into expressive power

of unambiguous automata

Theorem (Michalewski, S.)

If $\mathcal{A}$ is unambiguous of index $(i, 2j)$

then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j{-}1)$.

And we can construct

$\mathrm{Comp}(i, 2j{-}1)$ automaton for $\mathrm{L}(\mathcal{A})$

of size polynomial in $\mathcal{A}$.

Summary

— the first parity index collapse
based on unambiguity

— efficient transformation algorithm

— insights into expressive power
of unambiguous automata

Theorem (Michalewski, S.)
If $\mathcal{A}$ is unambiguous of index $(i, 2j)$
then $\mathrm{L}(\mathcal{A}) \in \mathrm{Comp}(i, 2j{-}1)$.

And we can construct
$\mathrm{Comp}(i, 2j{-}1)$ automaton for $\mathrm{L}(\mathcal{A})$
of size polynomial in $\mathcal{A}$.