Unambiguous languages exhaust the index hierarchy

Michał Skrzypczak

ICALP 2018

Prague 12.07.2018

FNP
Foundation for Polish Science

UNIVERSITY OF WARSAW

NATIONAL SCIENCE CENTRE
POLAND

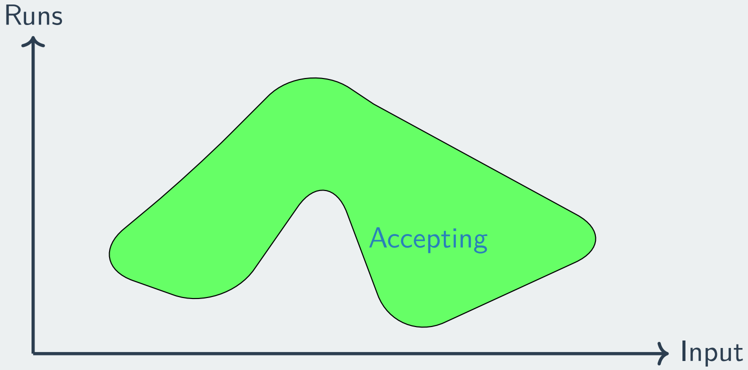
Non-determinism and unambiguity

Non-determinism and unambiguity

$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \Big\}$$

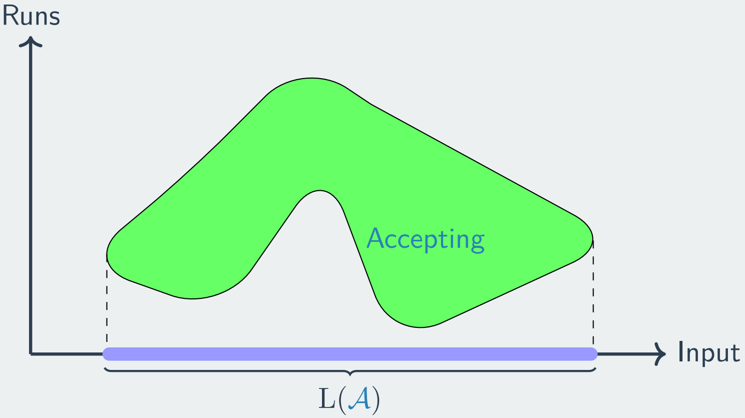
Non-determinism and **unambiguity**

$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \Big\}$$



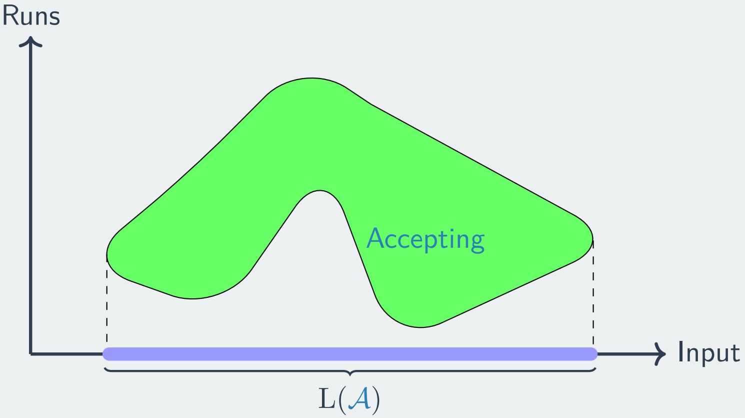
Non-determinism and unambiguity

$$\mathrm{L}(\mathcal{A}) = \left\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \right\}$$



Non-determinism and unambiguity

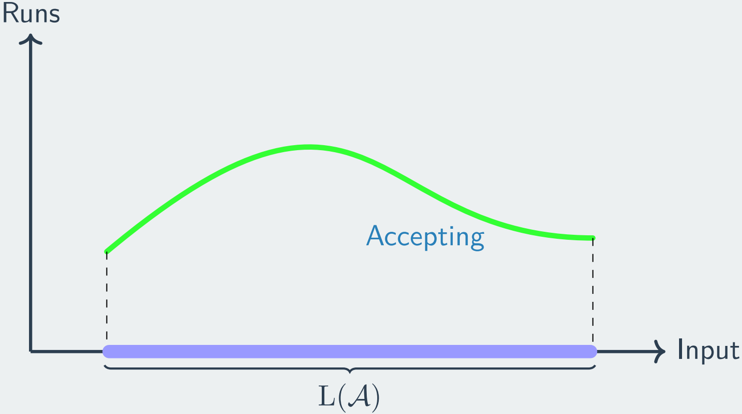
$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \Big\}$$



$\mathcal{A}$ is unambiguous
iff
$\forall w \in \mathrm{L}(\mathcal{A}).\ \exists! \rho.\ \rho$ is accepting over w

Non-determinism and unambiguity

$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \Big\}$$



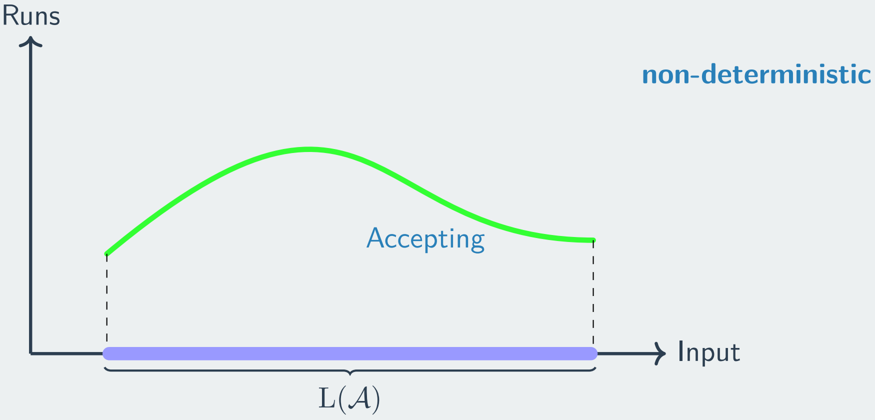
$\mathcal{A}$ is unambiguous

iff

$\forall w \in \mathrm{L}(\mathcal{A}).\ \exists! \rho.\ \rho$ is accepting over w

Non-determinism and **unambiguity**

$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \Big\}$$



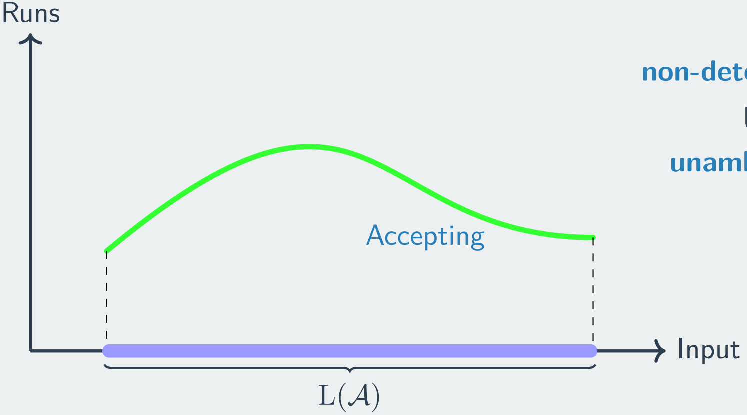
$\mathcal{A}$ is unambiguous

iff

$\forall w \in \mathrm{L}(\mathcal{A}).\ \exists! \rho.\ \rho$ is accepting over w

Non-determinism and **unambiguity**

$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists\rho.\ \rho \text{ is an accepting run over } w \Big\}$$



non-deterministic

⊇

unambiguous

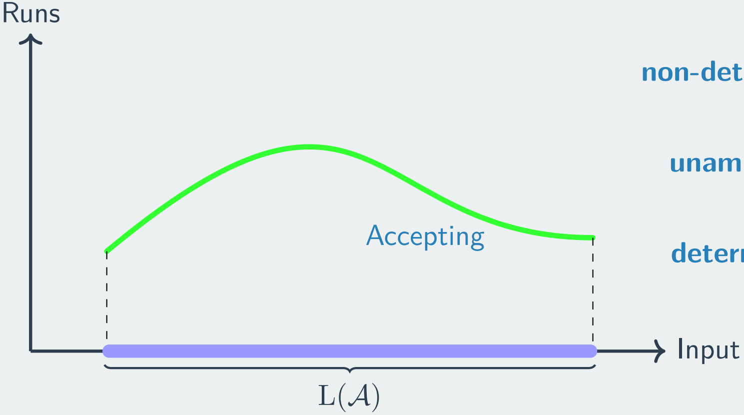
$\mathcal{A}$ is unambiguous

iff

$\forall w \in \mathrm{L}(\mathcal{A}).\ \exists!\rho.\ \rho$ is accepting over w

Non-determinism and **unambiguity**

$$\mathrm{L}(\mathcal{A}) = \Big\{ w \mid \exists \rho.\ \rho \text{ is an accepting run over } w \Big\}$$



non-deterministic

$\cup|$

unambiguous

$\cup|$

deterministic

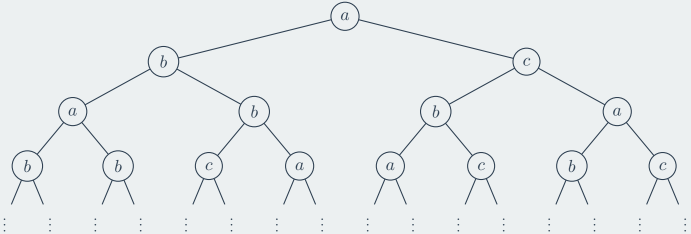
$\mathcal{A}$ is unambiguous

iff

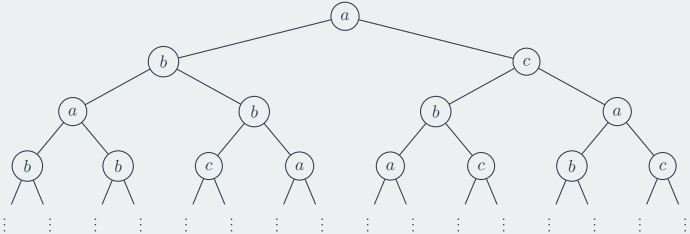
$\forall w \in \mathrm{L}(\mathcal{A}).\ \exists! \rho.\ \rho$ is accepting over w

Regular languages of **infinite trees**

Regular languages of **infinite trees**

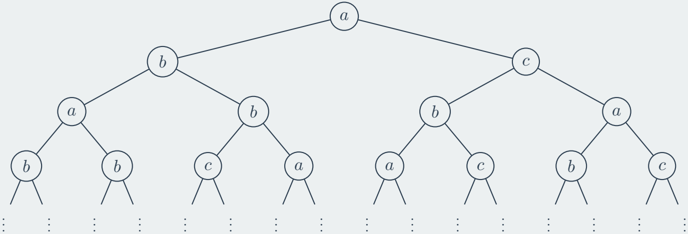


Regular languages of **infinite trees**



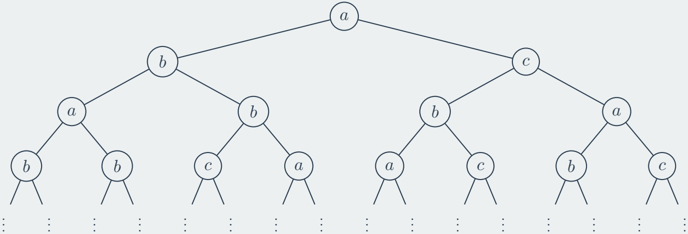
$$t\colon \{\mathtt{L}, \mathtt{R}\}^* \to A$$

Regular languages of **infinite trees**



$$t\colon \underbrace{\{\mathtt{L},\mathtt{R}\}^* \to A}_{\mathrm{Tr}_A}$$

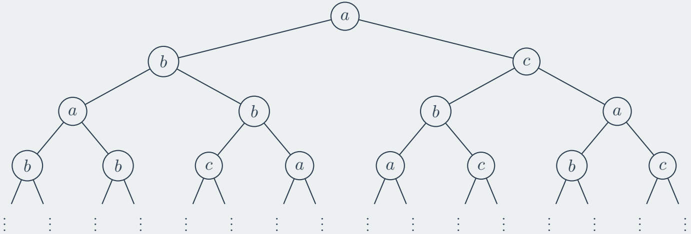
Regular languages of infinite trees



$$\underbrace{t\colon \{\mathtt{L},\mathtt{R}\}^* \to A}_{\mathrm{Tr}_A}$$

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

Regular languages of **infinite trees**

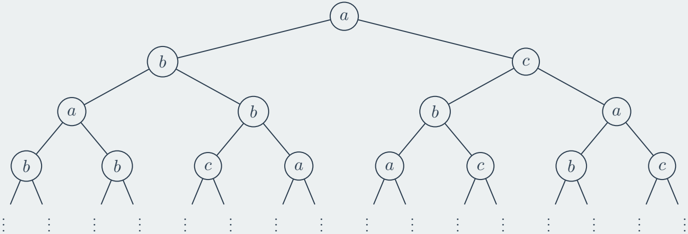


$$t\colon \underbrace{\{\mathtt{L},\mathtt{R}\}^* \to A}_{\mathrm{Tr}_A}$$

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

$\neg,\ \vee,\ \exists x,\ \exists X,\ x \in X,\ x = y$

Regular languages of **infinite trees**



$$t\colon \underbrace{\{\mathtt{L},\mathtt{R}\}^* \to A}_{\mathrm{Tr}_A}$$

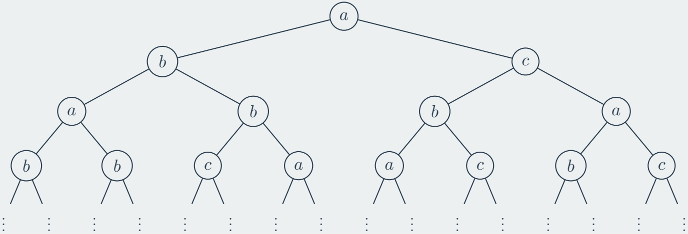
Definable in **Monadic Second-Order logic** (MSO)

$\neg,\ \vee,\ \exists x,\ \exists X,\ x \in X,\ x = y$

Theorem (Rabin [1969])

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

Regular languages of **infinite trees**



$$t\colon \underbrace{\{\mathtt{L},\mathtt{R}\}^* \to A}_{\mathrm{Tr}_A}$$

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

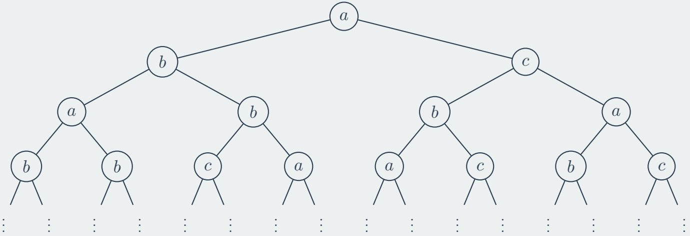
$\neg,\ \vee,\ \exists x,\ \exists X,\ x \in X,\ x = y$

Theorem (Rabin [1969])

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

Proof: Non-deterministic parity tree automata

Regular languages of **infinite trees**



$$t\colon \underbrace{\{\mathtt{L},\mathtt{R}\}^* \to A}_{\mathrm{Tr}_A}$$

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.

Proof: Non-deterministic parity tree automata (+ games)

Parity condition

Parity condition

$$\Omega\colon Q$$

$$\Uparrow$$

state q

Parity condition

$$\Omega\colon Q \to \{i, i{+}1, \ldots, j\}$$

$$\text{state} \quad q \in Q \longmapsto \Omega(q) \in \{i, \ldots, j\} \quad \text{priority}$$

Parity condition

$$\Omega\colon Q \to \{i, i+1, \ldots, j\}$$

$$\textbf{state}\ \ q \in Q \longmapsto \Omega(q) \in \{i,\ldots,j\}\ \ \textbf{priority}$$

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

Parity condition

$$\Omega\colon Q \to \{i, i+1, \ldots, j\}$$

$$\text{state}\quad q \in Q \longmapsto \Omega(q) \in \{i,\ldots,j\}\quad \text{priority}$$

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

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

Parity condition

$$\Omega\colon Q \to \{i, i{+}1, \ldots, j\}$$

$$\text{state} \quad q \in Q \longmapsto \Omega(q) \in \{i,\ldots,j\} \quad \text{priority}$$

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

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

"the highest priority seen infinitely often is even"

Parity condition

$$\Omega\colon Q \to \overbrace{\{i, i{+}1, \ldots, j\}}^{\text{index } (i,j)}$$

$$\text{state} \quad q \in Q \longmapsto \Omega(q) \in \{i,\ldots,j\} \quad \text{priority}$$

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

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

"the highest priority seen infinitely often is even"

Parity condition

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

$$\Omega\colon Q \to \overbrace{\{i, i{+}1, \ldots, j\}}^{\text{index } (i,j)}$$

$$\text{state } \; q \in Q \longmapsto \Omega(q) \in \{i, \ldots, j\} \; \text{priority}$$

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

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

"the highest priority seen infinitely often is even"

Non-deterministic parity tree automata

Non-deterministic parity tree automata

$$\mathrm{L}(\mathcal{A}) \stackrel{\text{def}}{=} \{t \in \mathrm{Tr}_A \mid$$

Non-deterministic parity tree automata

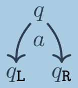
$\mathrm{L}(\mathcal{A}) \stackrel{\text{def}}{=} \{t \in \mathrm{Tr}_A \mid \exists \rho \in \mathrm{Tr}_Q.\ \rho$ is a run over t and

Non-deterministic parity tree automata

$$\mathrm{L}(\mathcal{A}) \stackrel{\text{def}}{=} \Big\{ t \in \mathrm{Tr}_A \mid \exists \rho \in \mathrm{Tr}_Q.\ \rho \text{ is a run over } t \quad \text{and} \quad \forall \pi \in \{\mathtt{L}, \mathtt{R}\}^\omega.\ \rho \text{ satisfies parity condition on } \pi \Big\}$$

Non-deterministic parity tree automata

local consistency

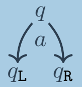


(transitions)

$$\mathrm{L}(\mathcal{A}) \overset{\text{def}}{=} \{ t \in \mathrm{Tr}_A \mid \exists \rho \in \mathrm{Tr}_Q.\ \rho \text{ is a run over } t \text{ and } \forall \pi \in \{\mathtt{L}, \mathtt{R}\}^\omega.\ \rho \text{ satisfies parity condition on } \pi \}$$

Non-deterministic parity tree automata

local consistency



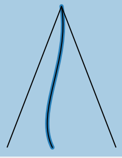
(transitions)

$\mathrm{L}(\mathcal{A}) \overset{\text{def}}{=} \{t \in \mathrm{Tr}_A \mid \exists \rho \in \mathrm{Tr}_Q.\ \rho$ is a run over t and

$\forall \pi \in \{\mathtt{L}, \mathtt{R}\}^{\omega}.\ \rho$ satisfies

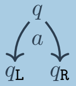
parity condition on $\pi\}$

path consistency



Non-deterministic parity tree automata

local consistency



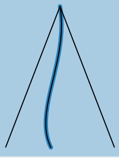
(transitions)

$$\mathrm{L}(\mathcal{A}) \stackrel{\text{def}}{=} \{t \in \mathrm{Tr}_A \mid \exists \rho \in \mathrm{Tr}_Q.\ \rho \text{ is a run over } t \quad \text{and}$$
$$\forall \pi \in \{\mathtt{L}, \mathtt{R}\}^{\omega}.\ \rho \text{ satisfies}$$
$$\text{parity condition on } \pi\}$$

non-determinism

!!!

path consistency



Unambiguous automata — expressive power

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a
cannot be recognised by any unambiguous automaton.

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

$\not\supseteq$

unambiguous

$\not\supseteq$

deterministic

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

$\supsetneq$

unambiguous

$\supsetneq$

deterministic

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a
cannot be recognised by any unambiguous automaton.

non-deterministic

$\supsetneq$

unambiguous

$\supsetneq$

deterministic

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

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

unambiguous

$\not\supseteq$

deterministic

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

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

unambiguous

$\not\supseteq$

deterministic

- which languages are unambiguous **?**

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

? $\supsetneq$ **?**

unambiguous

$\supsetneq$

deterministic

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

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

? $\subsetneq$ **?**

unambiguous

$\subsetneq$

deterministic

- which languages are unambiguous**?** (i.e. effective characterisation)
  ⇝ preliminary results (Bilkowski, S. [2013])

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

? $\not\supsetneq$ **?**

unambiguous

$\not\supsetneq$

deterministic

- which languages are unambiguous**?** (i.e. effective characterisation)
  $\rightsquigarrow$ preliminary results (Bilkowski, S. [2013])
- are there unambiguous languages that are complex**?**

Unambiguous automata — expressive power

Theorem (Niwiński, Walukiewicz [1996])

The language of trees that contain a letter a **cannot** be recognised by any unambiguous automaton.

non-deterministic

? $\subsetneq$?

unambiguous

$\subsetneq$

deterministic

- which languages are unambiguous**?** (i.e. effective characterisation)
  ⇝ preliminary results (Bilkowski, S. [2013])
- are there unambiguous languages that are complex**?**
  ⇝ Duparc + Fournier + Hummel + **this work**

Index hierarchy (alternating one)

Index hierarchy (alternating one)

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

Index hierarchy (alternating one)

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

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

Index hierarchy (alternating one)

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

$\mathcal{A}$ 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 $\Omega(q) \in \{i, i{+}1, \ldots, j\}$

a 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 $\Omega(q) \in \{i, i{+}1, \ldots, j\}$
a 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 $i \in \{0, 1\}$]

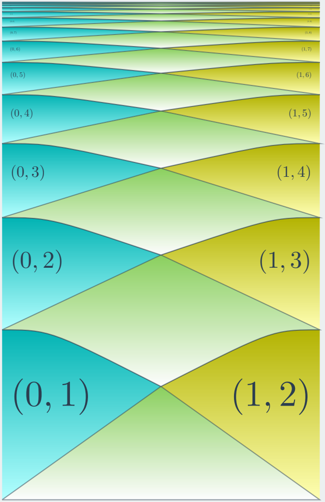
Index hierarchy (alternating one)

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

$\mathcal{A}$ 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)

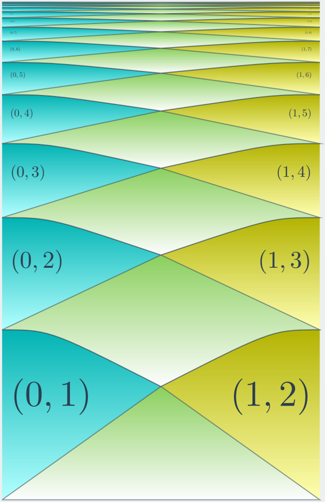
[parity automaton: every $q \in Q$ has a priority $\Omega(q) \in \{i, i+1, \ldots, j\}$
a sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

$\mathcal{A}$ 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 $\Omega(q) \in \{i, i+1, \ldots, j\}$
a sequence $q_0, q_1, \ldots$ is accepting if $\limsup_{n\to\infty} \Omega(q_n) \equiv 0 \pmod 2$]

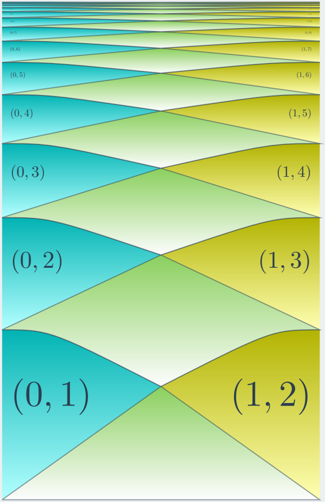
$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

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 $\cong$ Alternation depth hierarchy



Index hierarchy (alternating one)

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

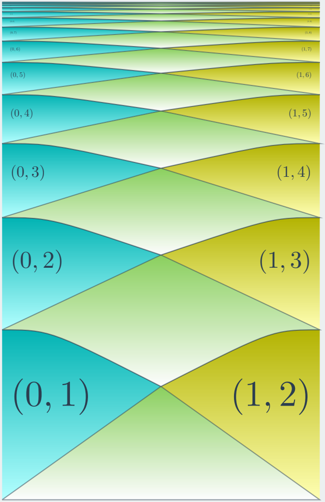
$\mathcal{A}$ has index (i, j) if $\Omega\colon Q \to \{i, \ldots, j\}$

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 $\cong$ 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 $\Omega(q) \in \{i, i+1, \ldots, j\}$
a 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 $i \in \{0, 1\}$]

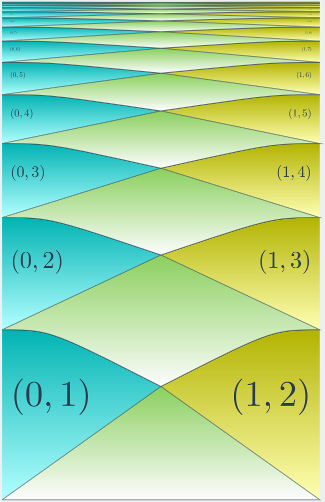
Theorem For modal μ-calculus:

Index hierarchy ≅ Alternation depth hierarchy

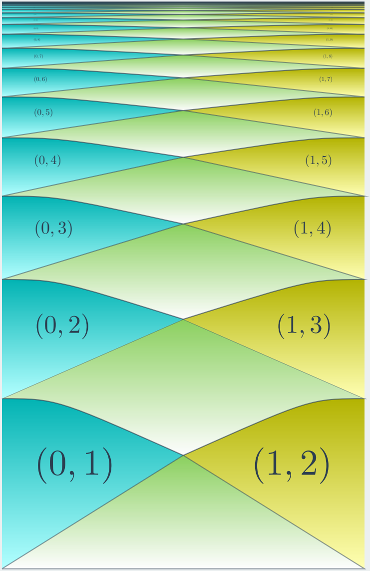
Theorem (Bradfield [1998], Arnold [1999])

There are languages requiring **big** indices.

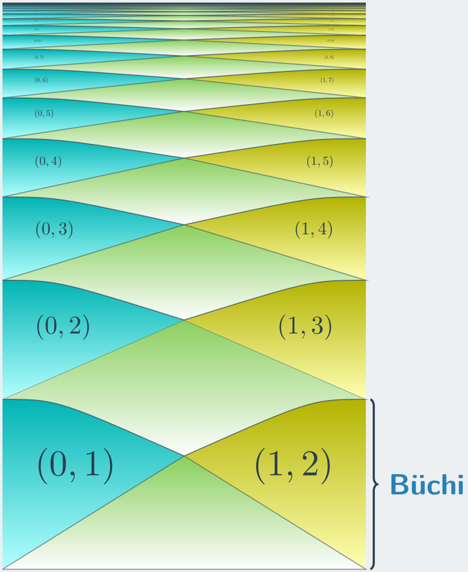
(i.e. the hierarchies are strict)



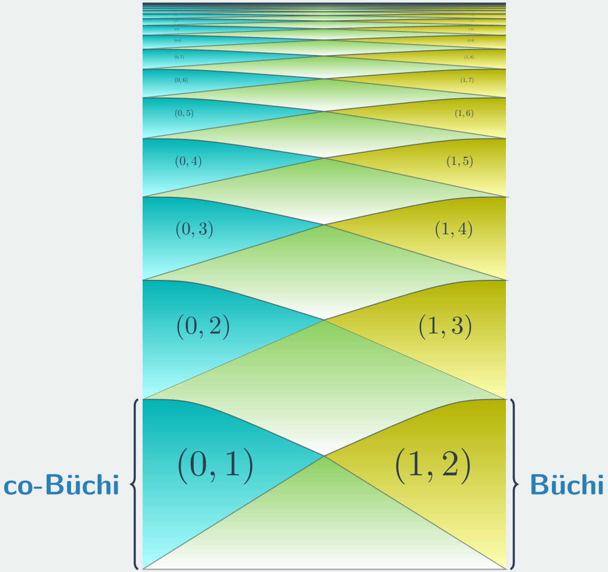
Alternating index hierarchy & unambiguous languages



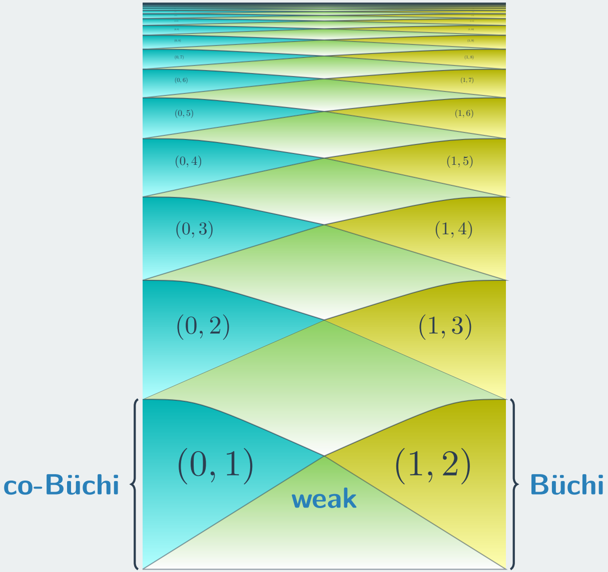
Alternating index hierarchy & unambiguous languages



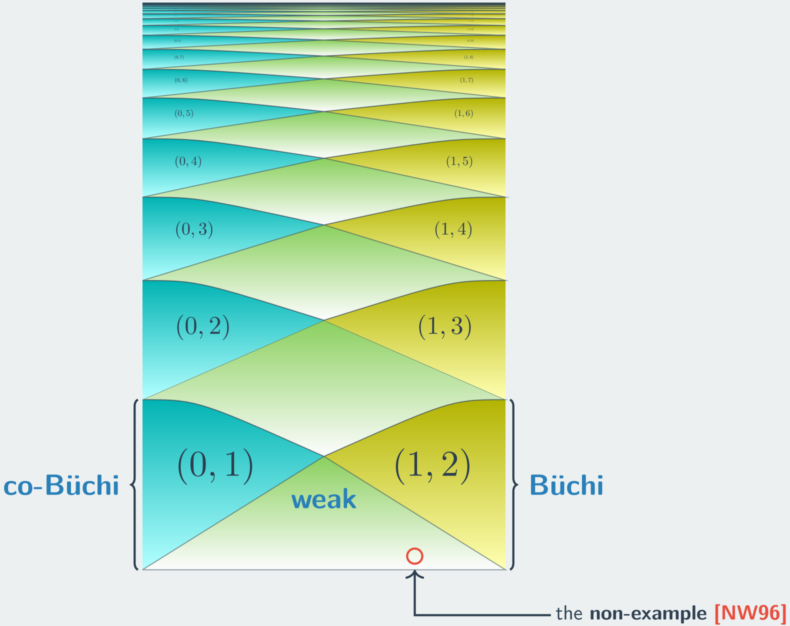
Alternating index hierarchy & unambiguous languages



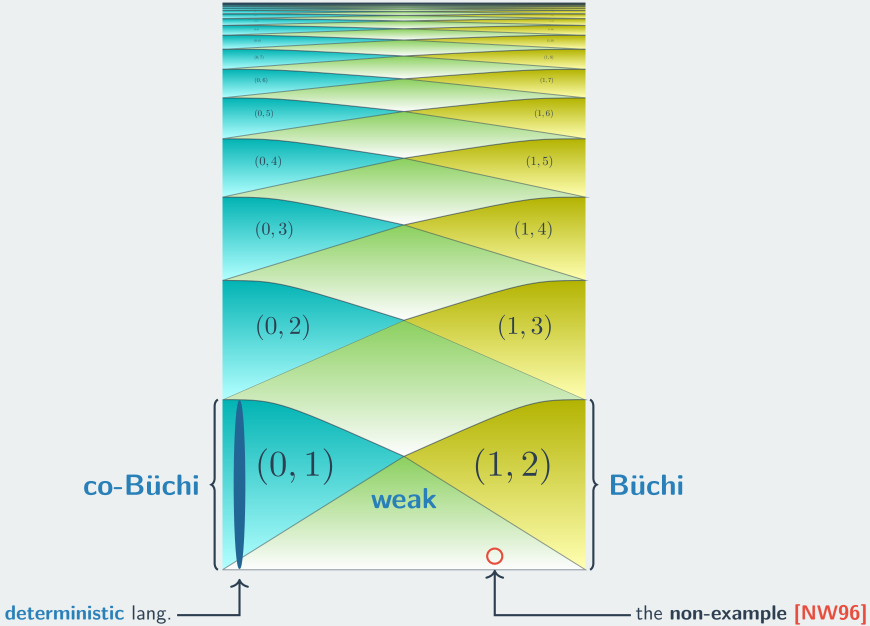
Alternating index hierarchy & unambiguous languages



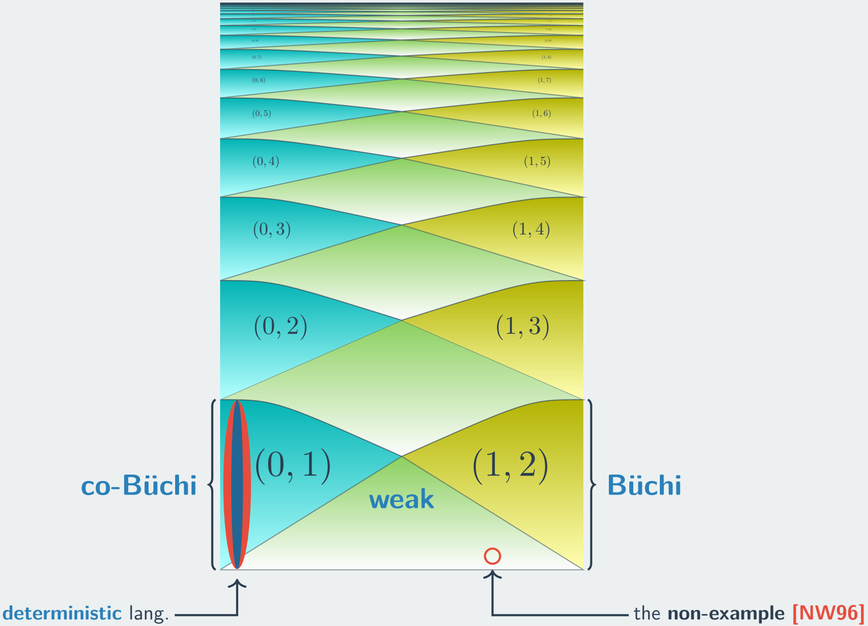
Alternating index hierarchy & unambiguous languages



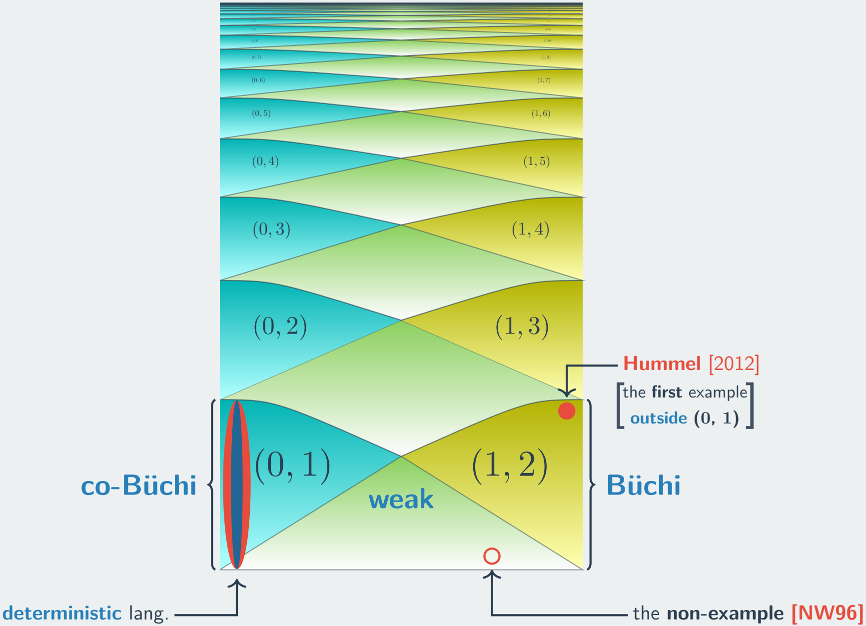
Alternating index hierarchy & unambiguous languages



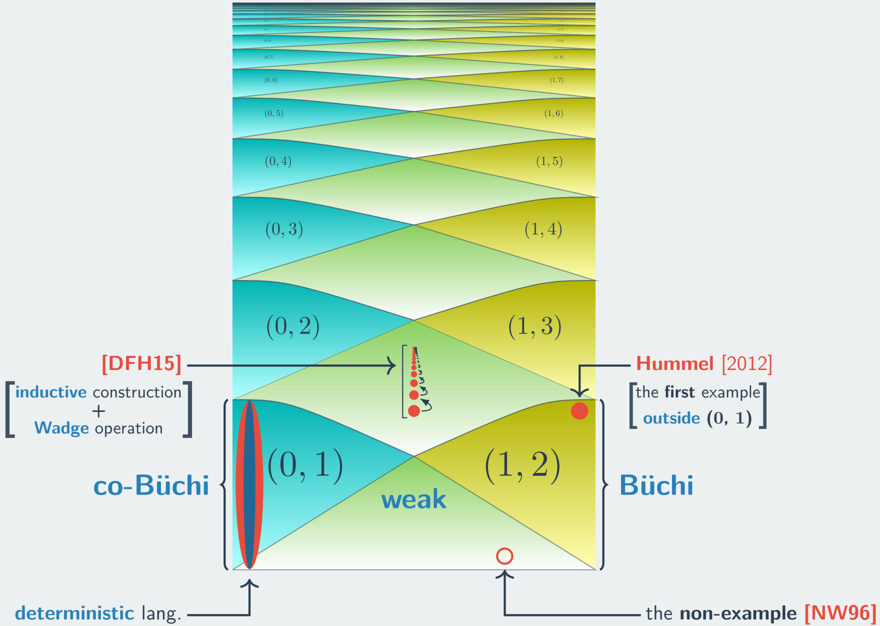
Alternating index hierarchy & unambiguous languages



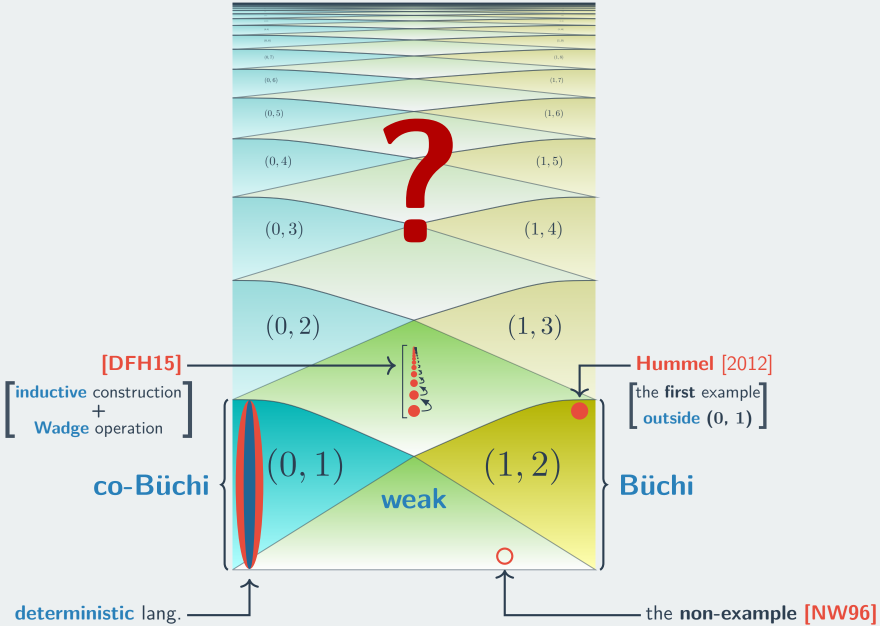
Alternating index hierarchy & unambiguous languages



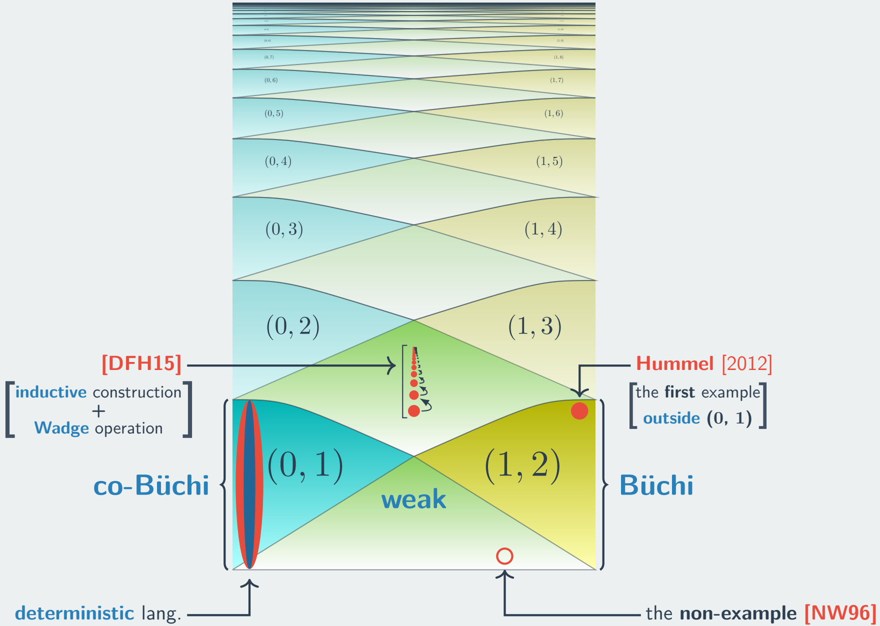
Alternating index hierarchy & unambiguous languages



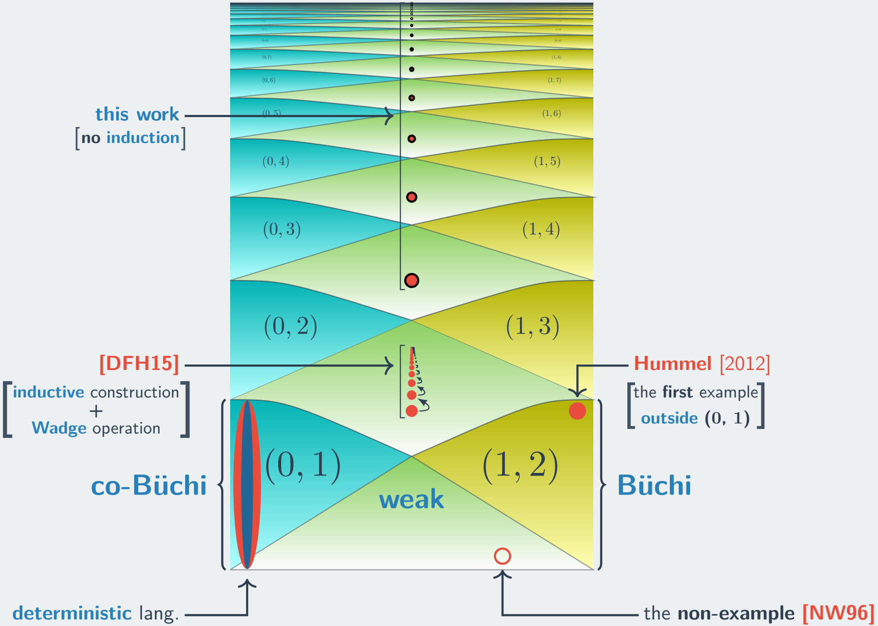
Alternating index hierarchy & unambiguous languages



Alternating index hierarchy & unambiguous languages



Alternating index hierarchy & unambiguous languages



Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

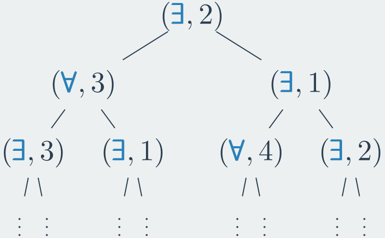
$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

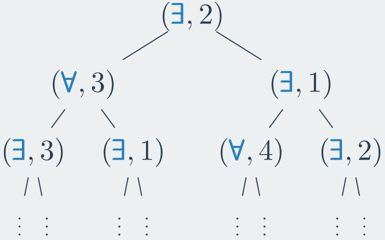
t over $A_{i,k}$



Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

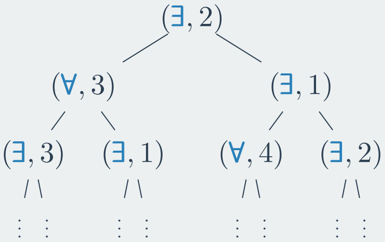
t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

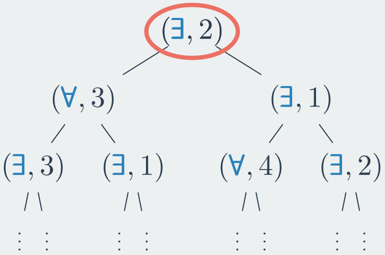


$\pi =$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

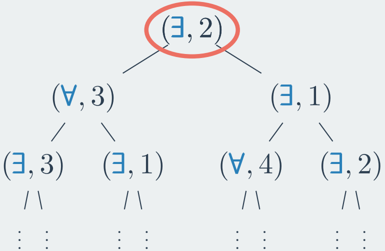


$\pi =$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

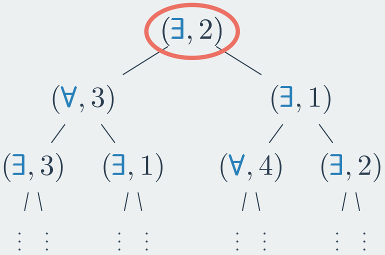


$\pi = 2$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



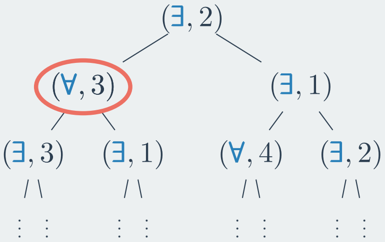
$\pi = 2$

$\exists \colon d_0 \quad (= \texttt{L})$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

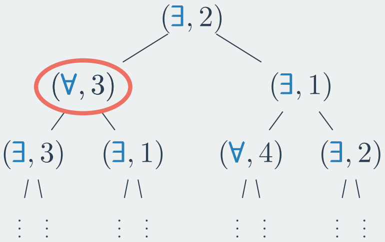


$\pi = 2$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i+1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

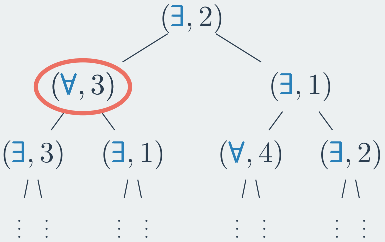


$\pi = 2,\ 3$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \dots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



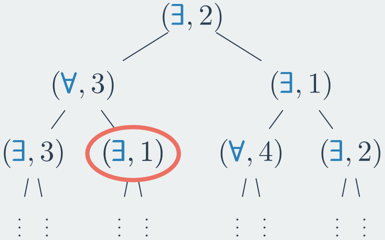
$\pi = 2,\ 3$

$\forall$: d_1 $(= \mathtt{R})$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

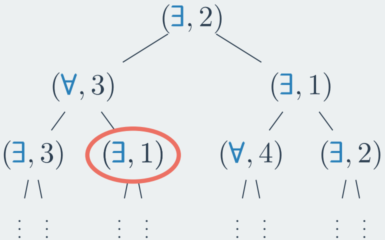


$\pi = 2,\ 3$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$

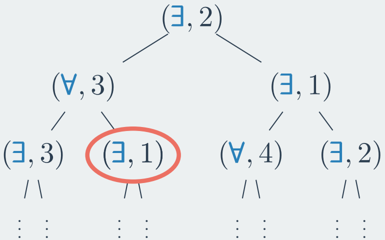


$\pi = 2,\ 3,\ 1$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i+1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



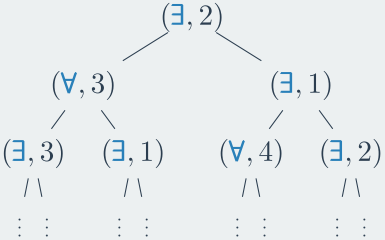
$\pi = 2,\ 3,\ 1$

$\exists, \forall$: d_n

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



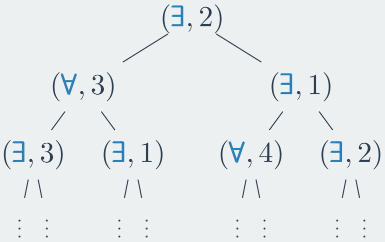
$\pi = 2,\ 3,\ 1,\ \ldots$

$\exists, \forall\colon d_n$

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \dots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



$\pi = 2,\ 3,\ 1,\ \dots$

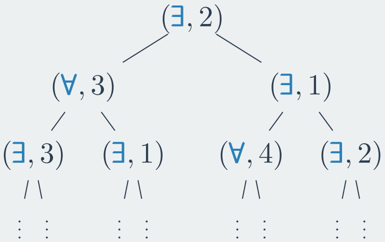
$\exists, \forall$: d_n

$\exists$ wins π if $\limsup$ is even

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



$\pi = 2,\ 3,\ 1,\ \ldots$

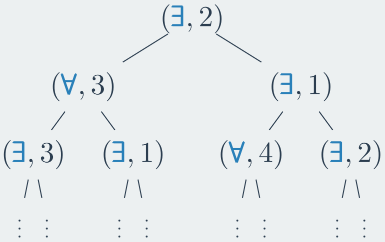
$\exists, \forall$: d_n

$\exists$ wins π if $\underbrace{\limsup}_{\text{parity condition}}$ is even

Languages $W_{i,j}$ (Walukiewicz [1996]) (also (Emerson, Jutla [1991]))

$$A_{i,j} = \{\exists, \forall\} \times \{i, i{+}1, \ldots, j\}$$

t over $A_{i,k}$ $\rightsquigarrow$ parity game $\mathcal{G}_t$



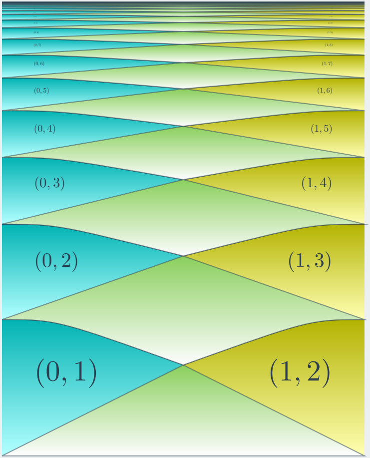
$\pi = 2,\ 3,\ 1,\ \ldots$

$\exists, \forall$: d_n

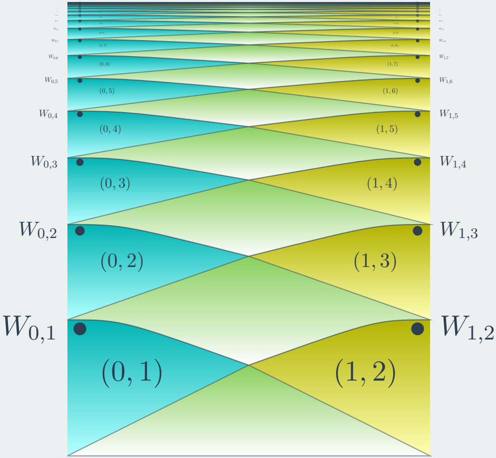
$\exists$ wins π if $\underbrace{\limsup}_{\text{parity condition}}$ is even

$t \in W_{i,j}$ **iff** $\exists$ has a winning strategy in $\mathcal{G}_t$

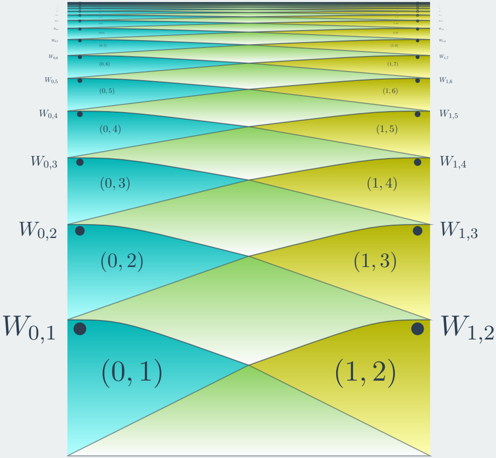
Hardness of the languages $W_{i,j}$



Hardness of the languages $W_{i,j}$

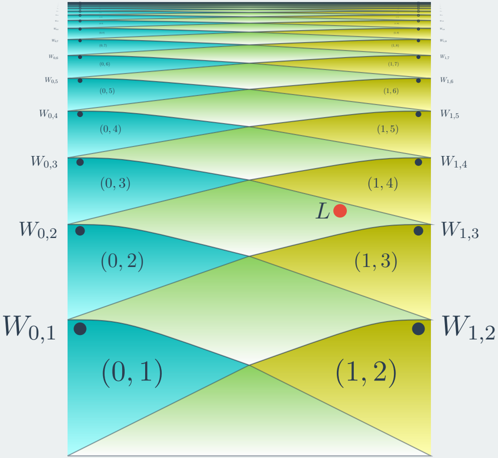


Hardness of the languages $W_{i,j}$



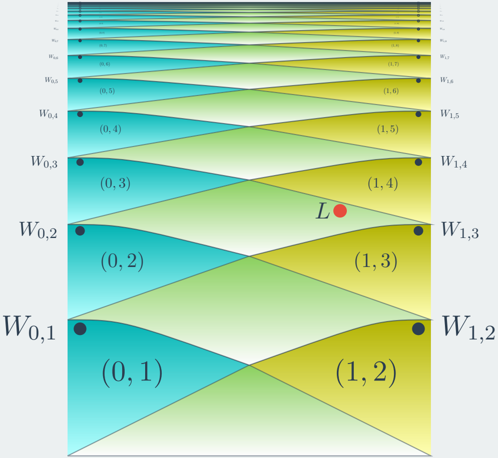
Theorem (Arnold [1999], Arnold, Niwiński [2007])

Hardness of the languages $W_{i,j}$



Theorem (Arnold [1999], Arnold, Niwiński [2007])

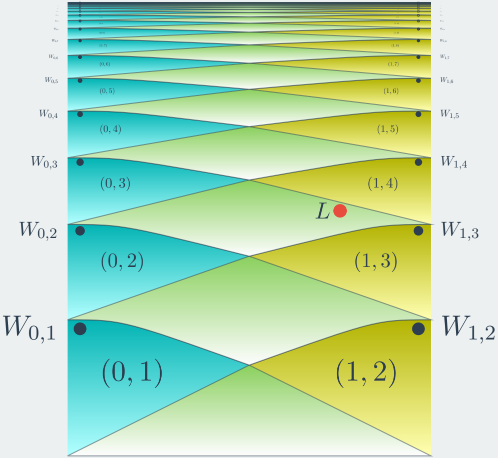
Hardness of the languages $W_{i,j}$



Theorem (Arnold [1999], Arnold, Niwiński [2007])

$$L \succ W_{i,j}$$

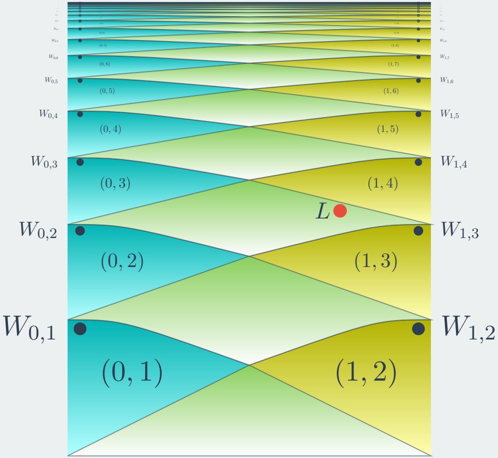
Hardness of the languages $W_{i,j}$



Theorem (Arnold [1999], Arnold, Niwiński [2007])

$$\underbrace{L \succ W_{i,j}}_{\text{Wadge order}}$$

Hardness of the languages $W_{i,j}$



Theorem (Arnold [1999], Arnold, Niwiński [2007])

$$\underbrace{L \succ W_{i,j}}_{\text{Wadge order}} \implies L \text{ is } \textbf{not } (i,j) \text{ recognisable}$$

Idea

Idea

1. Take $i < j$.

Idea

1. Take $i < j$.
2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

3.b and $L_{i,j} \geq W_{i,j}$.

Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

3.b and $L_{i,j} \geq W_{i,j}$.

} recognise unambiguously $W_{i,j}$

Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

3.b and $L_{i,j} \geq W_{i,j}$.

} recognise unambiguously $W_{i,j}$ with the help of an advice

Idea

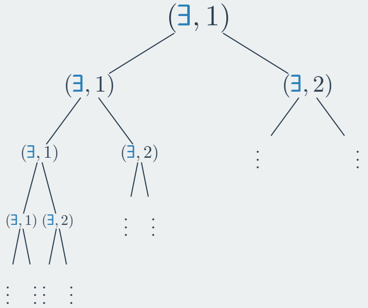
1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

3.b and $L_{i,j} \succeq W_{i,j}$.

} recognise unambiguously $W_{i,j}$ with the help of an advice



Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

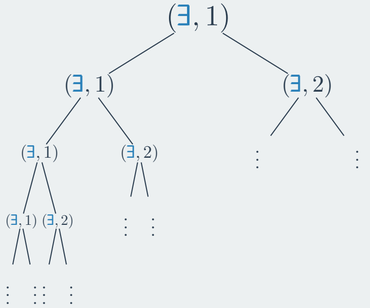
3.a such that $L_{i,j}$ is unambiguous

3.b and $L_{i,j} \geq W_{i,j}$.

} recognise unambiguously $W_{i,j}$ with the help of an advice

Problem

Staying within a winning region is **not** enough!



Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

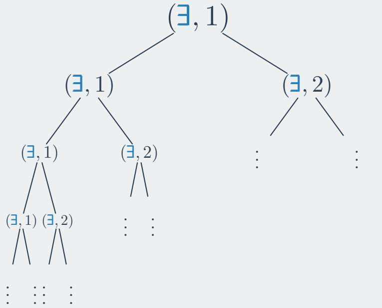
3.b and $L_{i,j} \geq W_{i,j}$.

} recognise unambiguously $W_{i,j}$ with the help of an advice

Problem

Staying within a winning region is **not** enough!

⇝ choose the **optimal** subtree



Idea

1. Take $i < j$.

2. Find a variant of $W_{i,j}$ (denoted $L_{i,j}$)

3.a such that $L_{i,j}$ is unambiguous

3.b and $L_{i,j} \geq W_{i,j}$.

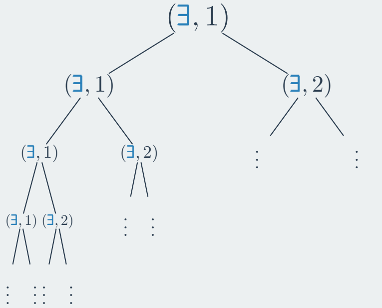
} recognise unambiguously $W_{i,j}$ with the help of an advice

Problem

Staying within a winning region is **not** enough!

⇝ choose the **optimal** subtree

⇝ appraise the subtrees by their **signatures**



Signatures

Signatures [$(0,2)$ case]

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$

Signatures [(0, 2) case]

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

Signatures [$(0,2)$ case]

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$

Signatures $[(0,2)$ case]

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s

Signatures $[(0,2)$ case]

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_\exists^{\min}$ is winning.

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_\exists^{\min}$ is winning. $\left[\rightsquigarrow \text{positional determinacy}\right]$

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_\exists^{\min}$ is winning. $[\rightsquigarrow$ positional determinacy$]$

Technical core

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_\exists^{\min}$ is winning. $[\rightsquigarrow$ positional determinacy$]$

Technical core

$$c\colon (t_1, t_2) \mapsto t$$

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_\exists^{\min}$ is winning. [$\rightsquigarrow$ positional determinacy]

Technical core

$$c\colon (t_1, t_2) \mapsto t$$

[continuous]

Signatures $[(0,2)$ case$]$

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_{\exists}$

Let $S_{\exists}^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_{\exists}^{\min}$ is winning. $[\rightsquigarrow$ positional determinacy$]$

Technical core

$$c\colon (t_1, t_2) \mapsto t$$

[continuous]

$$\sigma(t_1) \leqslant \sigma(t_2)$$

Signatures [$(0,2)$ case]

$\sigma\colon \mathrm{Tr}_{A_{0,2}} \to \mathrm{On} \sqcup \{\infty\}$ — ordinal numbers $< \omega_1$ or ∞

- If $t \notin W_{0,2}$ then $\sigma(t) = \infty$
- If $t \in W_{0,2}$ then $\sigma(t)$ counts the minimal number of 1s among all winning strategies $S_\exists$

Let $S_\exists^{\min}$ be the strategy minimising $\sigma(t)$.

Lemma (Walukiewicz)

If $t \in W_{0,2}$ then $S_\exists^{\min}$ is winning. [$\rightsquigarrow$ positional determinacy]

Technical core

$c\colon (t_1, t_2) \mapsto t$

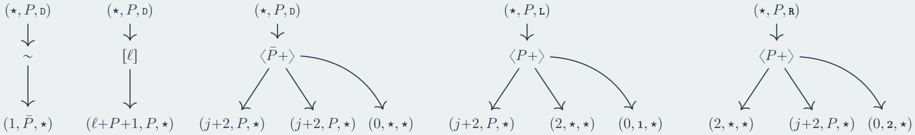
[continuous]

$$\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{0,2}$$

Summary

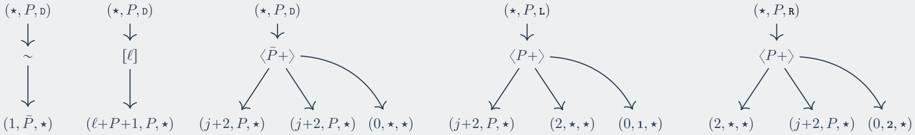
Summary

1. **Simple** automaton $\mathcal{A}_{i,j}$



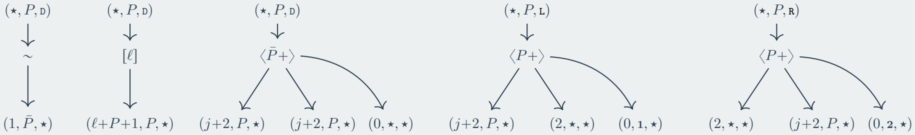
Summary

1. **Simple** automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



Summary

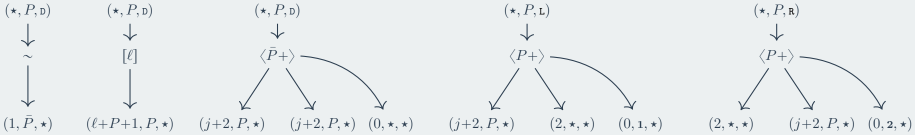
1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



2. Unambiguous (almost) by the definition

Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j+2)$)

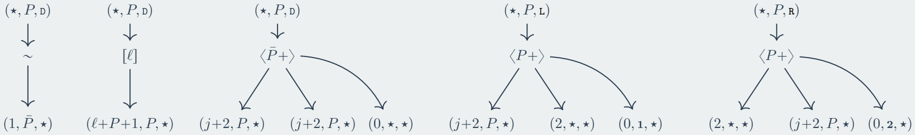


2. Unambiguous (almost) by the definition

3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \succeq W_{i,j}$

Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



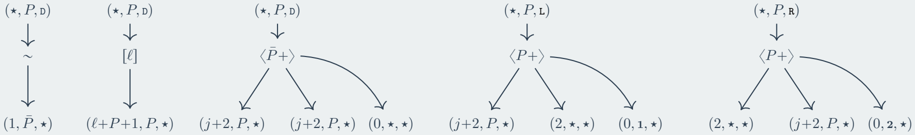
2. Unambiguous (almost) by the definition

3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \succeq W_{i,j}$

3.1 use **signatures**

Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



2. Unambiguous (almost) by the definition

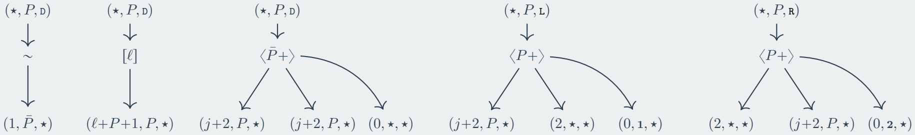
3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \geq W_{i,j}$

3.1 use **signatures**

3.2 $\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{i,j}$

Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



2. Unambiguous (almost) by the definition

3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \geq W_{i,j}$

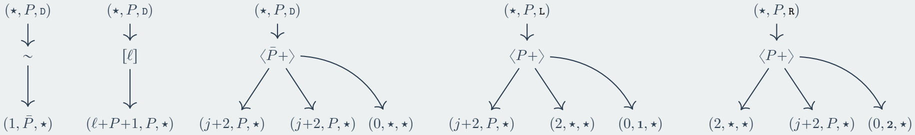
3.1 use **signatures**

3.2 $\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{i,j}$

■

Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



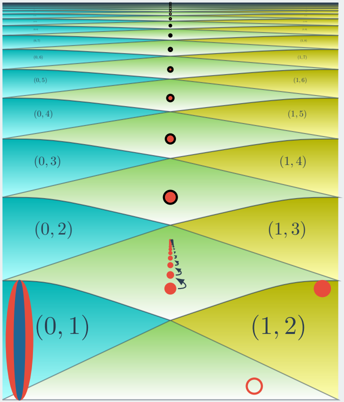
2. Unambiguous (almost) by the definition

3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \geq W_{i,j}$

3.1 use **signatures**

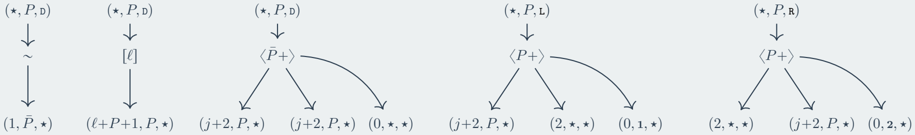
3.2 $\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{i,j}$

■



Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



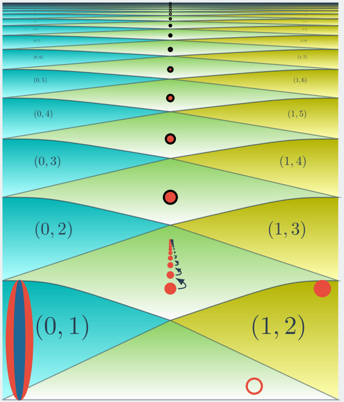
2. Unambiguous (almost) by the definition

3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \geq W_{i,j}$

3.1 use **signatures**

3.2 $\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{i,j}$

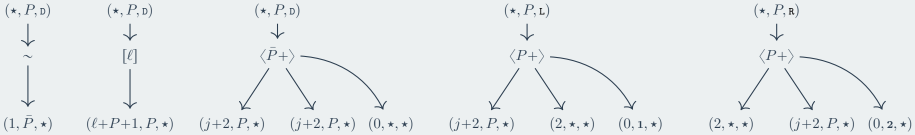
■



⇝ some **unambiguous** languages are **complex**

Summary

1. **Simple** automaton $\mathcal{A}_{i,j}$ (of index $(i, j+2)$)

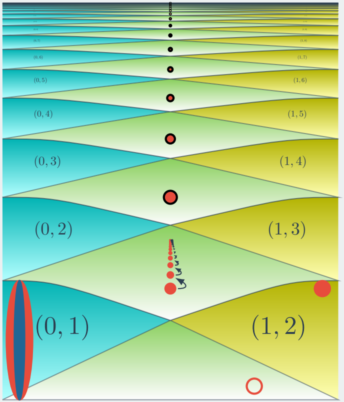


2. **Unambiguous** (almost) by the definition

3. **Claim**: $\mathrm{L}(\mathcal{A}_{i,j}) \geq W_{i,j}$

   3.1 use **signatures**

   3.2 $\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{i,j}$ ■

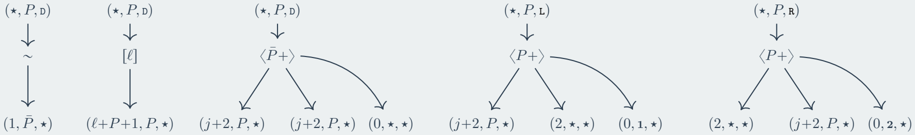


⇝ some **unambiguous** languages are **complex**

⇝ possible insights into **parity games**

Summary

1. Simple automaton $\mathcal{A}_{i,j}$ (of index $(i, j{+}2)$)



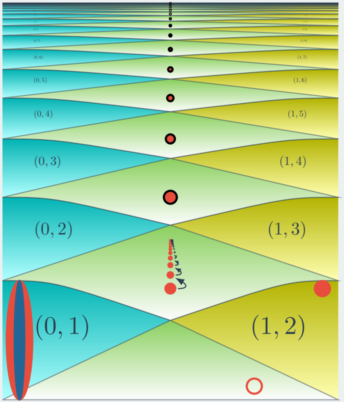
2. Unambiguous (almost) by the definition

3. Claim: $\mathrm{L}(\mathcal{A}_{i,j}) \geq W_{i,j}$

3.1 use **signatures**

3.2 $\sigma(t_1) \leqslant \sigma(t_2) \iff c(t_1, t_2) \in W_{i,j}$

■



⇝ some **unambiguous** languages are **complex**

⇝ possible insights into **parity games**

⇝ relations with **stage comparison theorems**