A tentative approach for the Wadge-Wagner hierarchy of regular tree languages of index $[0, 2]$

Jacques Duparc, Kevin Fournier

UNIL | Université de Lausanne

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Descriptional Complexity of Formal Systems 2015

DCFS 2015,

Waterloo, 25-27 June 2015

Joint work with

1. Kevin Fournier (Lausanne) kevin.fournier@unil.ch

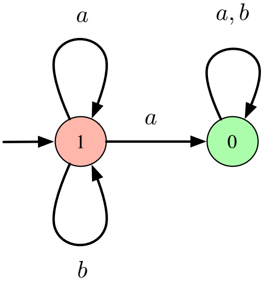
References

- Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the AMS*, 141:1–23, 1969.
- William W. Wadge. Reducibility and determinateness on the Baire space. *PhD thesis, University of California, Berkeley*, 1984.
- André Arnold, Jacques Duparc, Filip Murlak, and Damian Niwiński. On the topological complexity of tree languages. *Logic and automata: History and Perspectives*, 2:9–29, 2007.
- Arnaud Carayol and Christof Löding. MSO on the infinite binary tree: Choice and order. *In CSL*, pages 161–176, 2007.
- Filip Murlak. The wadge hierarchy of deterministic tree languages. *Logical Methods in Computer Science*, 4(4), 2008.
- Arnaud Carayol, Christof Löding, Damian Niwiński, and Igor Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *Central European Journal of Mathematics*, 8:662–682, 2010.

Outline

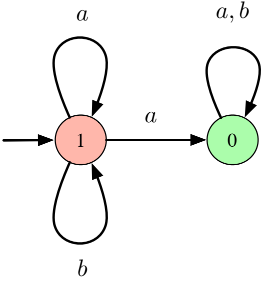
Automata on Infinite Words

- Automata that read infinite words
  (*parity acceptance conditions*: the largest priority visited infinitely often is even)



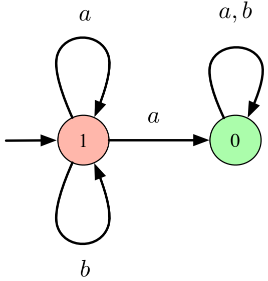
Automata on Infinite Trees

- What is the language?



Automata on Infinite Trees

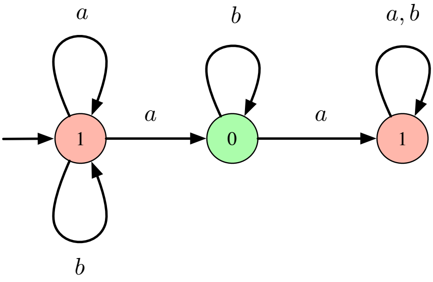
- (Σ^0_1-complete)



$$\left\{ w \mid w \textit{ contains the letter } a \right\}$$

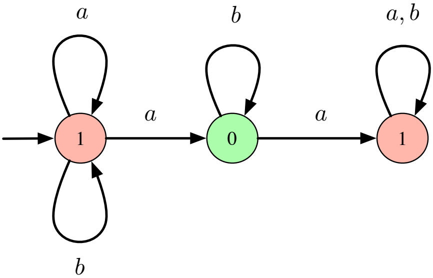
Automata on Infinite Trees

- What is the language?



Automata on Infinite Trees

- ($\mathbf{\Sigma}^0_2$-complete)

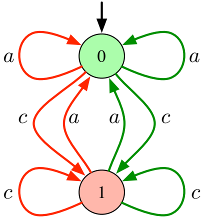


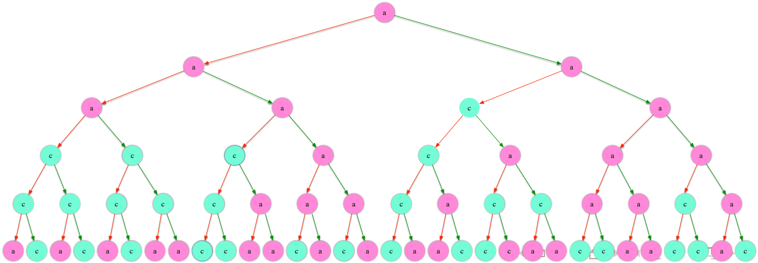
$$\left\{ w \mid w \textit{ contains at least one but finitely many letters } a \right\}$$

Outline

Automata on Infinite Trees

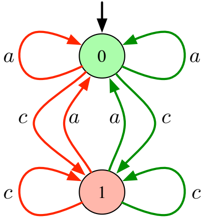
- Automata that read infinite (binary) trees
  (*parity acceptance conditions*: the largest priority visited infinitely often is even)

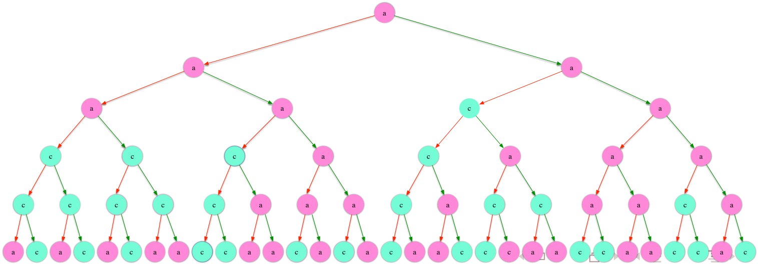




Automata on Infinite Trees

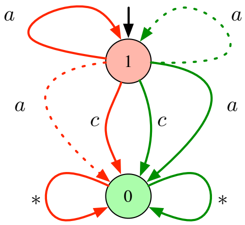
- Deterministic ⇒ tree is *accepted* iff *all* branches are accepted
  ⇒ tree is *rejected* iff *some* branch is rejected

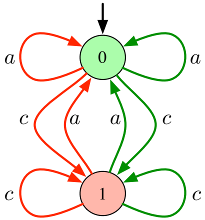




Automata on Infinite Trees

- Nondeterministic ⇒ Acceptance or Rejection of an input relies on *solving a game*: input tree is accepted iff *Automaton* has a w.s.
- two players:
  - *Automaton* takes charge of *nondeterminism*
  - *Pathfinder* takes charge of *branching* (left or right)
  - *Automaton* wins iff largest priority visited infinitely often is *even*
- yields $\mathbf{\Delta}^1_2$ languages





Automata on Infinite Trees

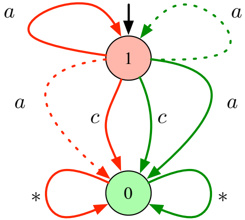
- *Nondeterministic parity tree automaton*

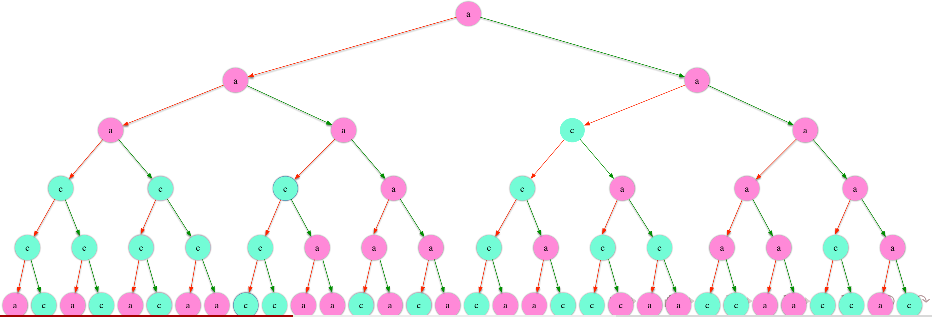
$$\mathcal{A} = \langle A, Q, I, \delta, r \rangle$$

  - a finite input alphabet A,
  - a finite set Q of states,
  - a set of initial states $I \subseteq Q$,
  - a transition relation $\delta \subseteq Q \times A \times Q \times Q$ and
  - a priority function $r : Q \to \omega$.

Automata on Infinite Trees

- What is the language?

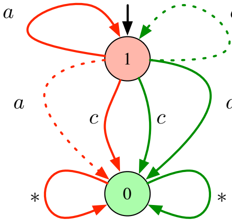


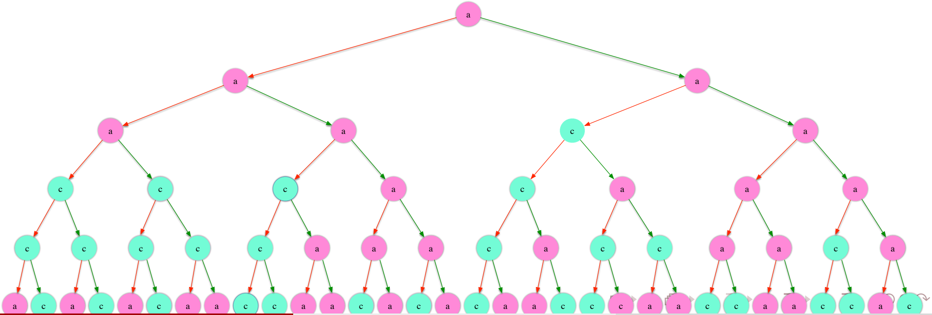


Automata on Infinite Trees

- ($\mathbf{\Sigma}^0_1$-complete)

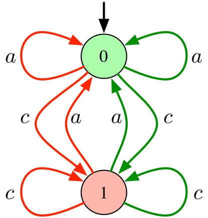
$\left\{ t \mid t \textit{ contains a } \text{c} \right\}$

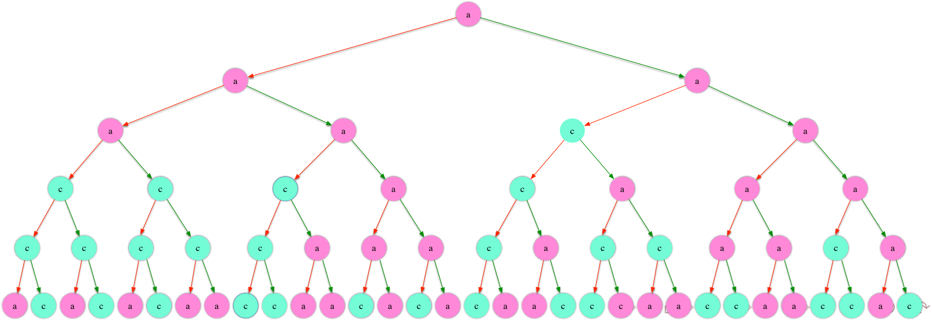




Automata on Infinite Trees

- What is the language?

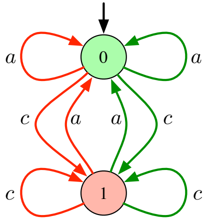


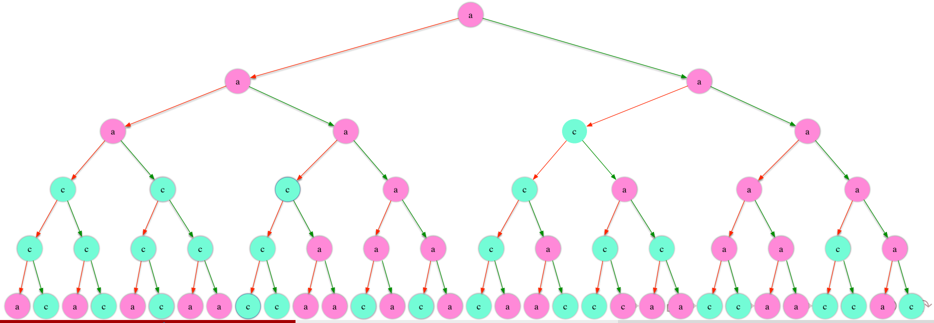


Automata on Infinite Trees

- ($\mathbf{\Pi}^1_1$-complete)

$$\left\{ t \mid \text{ no branch contains } \infty \text{ many } c \right\}$$





Automata on Infinite Trees

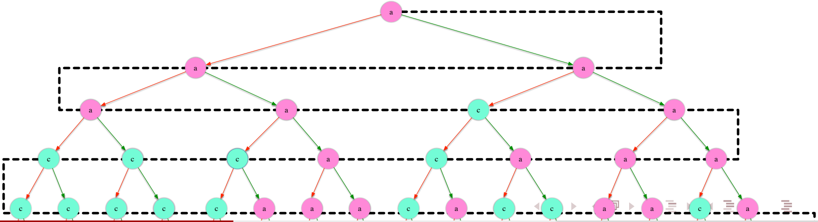
- The space T_A (of all full binary trees on the alphabet A) is homeomorphic to the Cantor space:

  1. either consider the prefix topology: any basic open set is of the form

  $$N_s = \left\{ t \in T_A \mid s \subseteq t \right\}$$

  for some finite tree s over A;

  2. or view each tree $t \in T_A$ as an infinite sequence $x_t \in A^\omega$.
  A is finite $\Rightarrow$ A^ω (equipped with the product topology of the discrete topology on A) is homeomorphic to the Cantor Space.



Outline

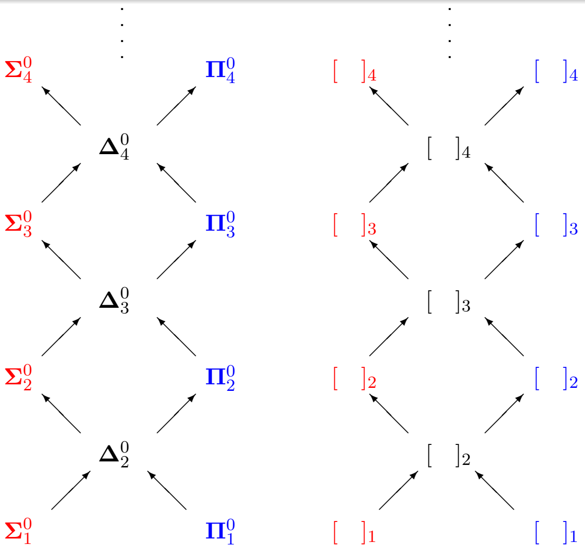
Automata on Infinite Trees

1. Given $L, M \subseteq T_A$, in the Wadge game $\boxed{\mathbb{W}(L, M)}$:
   1. player I builds $t_I \in T_A$ and player $I\!I$ builds $t_{I\!I} \in T_A$;
   2. at every round, player I plays first, and both players add a finite number of children to the terminal nodes of their ongoing trees;
   3. player $I\!I$ is allowed to skip her turn, but has to produce a tree in T_A throughout a game;
   4. Player $I\!I$ wins the game if and only if $t_I \in L \Leftrightarrow t_{I\!I} \in M$.
2. $L \leq_W M \Leftrightarrow$ $I\!I$ has a w.s. in $\mathbb{W}(L, M)$

   $\Leftrightarrow$ there exists f continuous s.t. $L = f^{-1}M$
3. $L \equiv_W M \;\Leftrightarrow\; L \leq_W M$ *and* $M \leq_W L$.

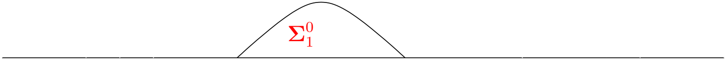
Outline

The Wadge Hierarchy

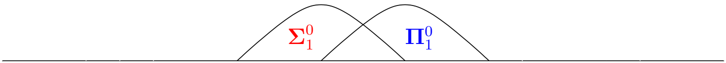


Borel hierarchy *Wadge hierarchy*

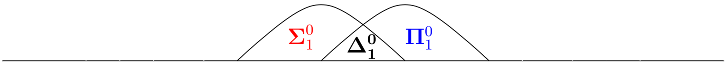
The Wadge Hierarchy



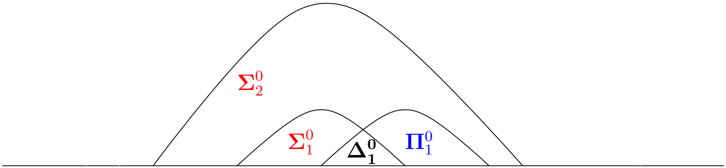
The Wadge Hierarchy



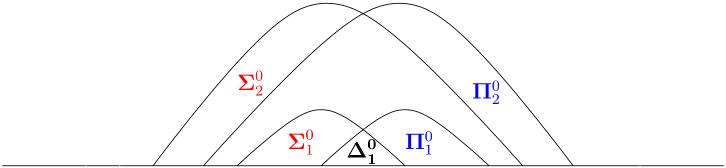
The Wadge Hierarchy



The Wadge Hierarchy

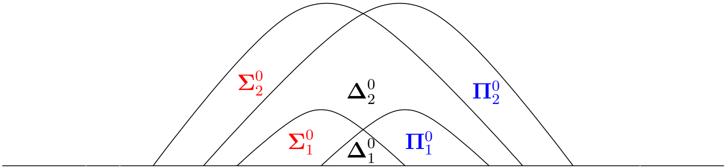


The Wadge Hierarchy

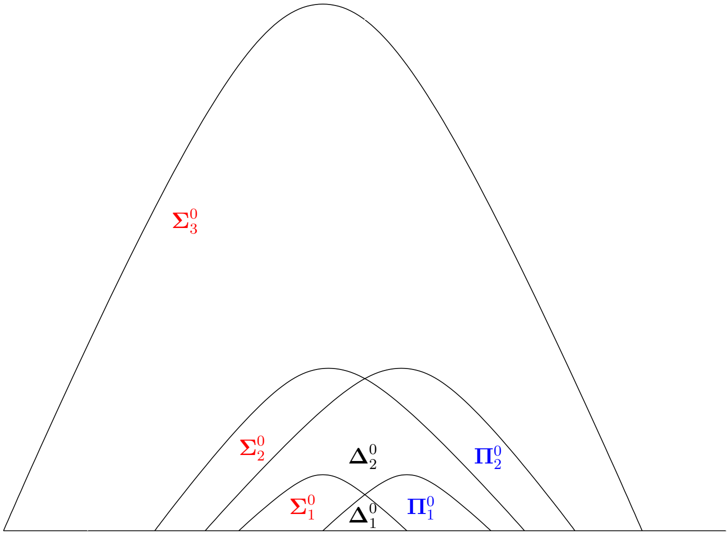


The Wadge Hierarchy

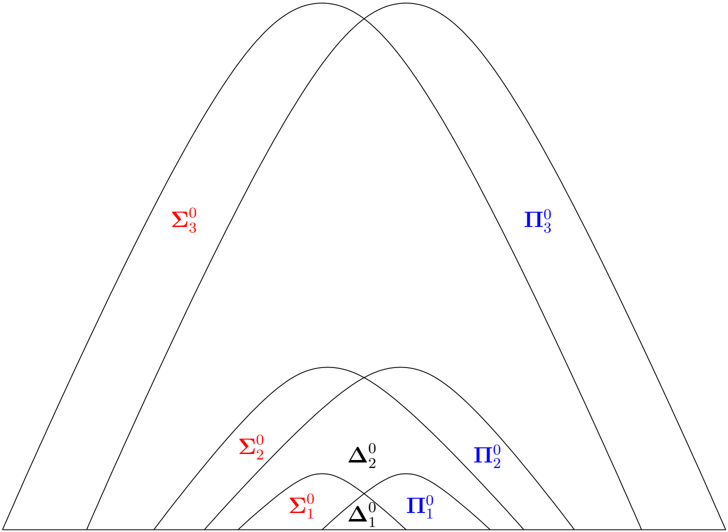
ω_1



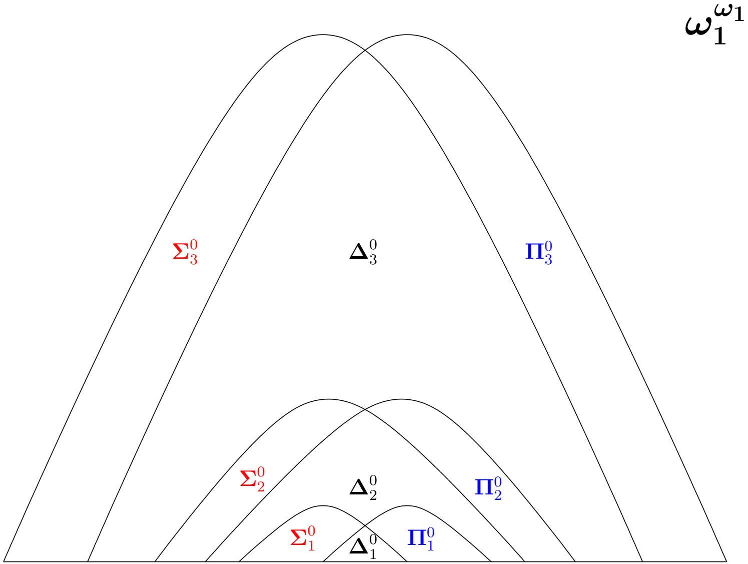
The Wadge Hierarchy



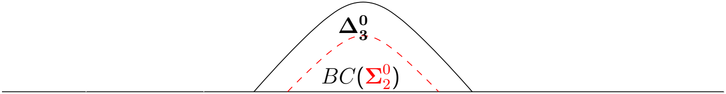
The Wadge Hierarchy



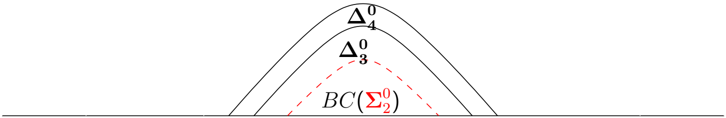
The Wadge Hierarchy



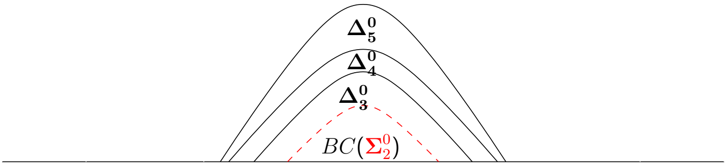
The Wadge Hierarchy



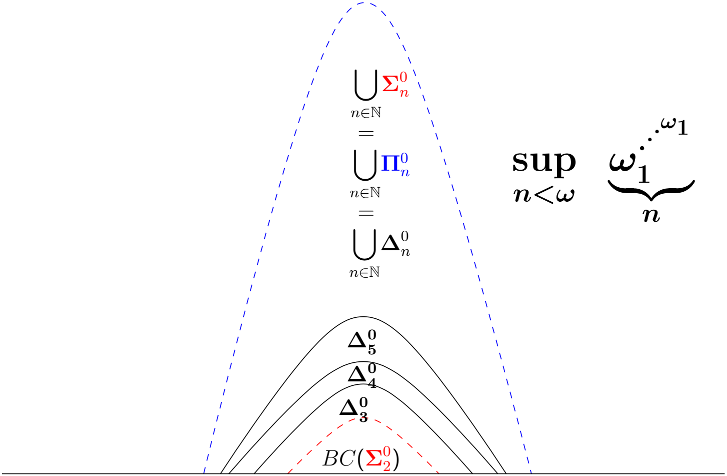
The Wadge Hierarchy



The Wadge Hierarchy

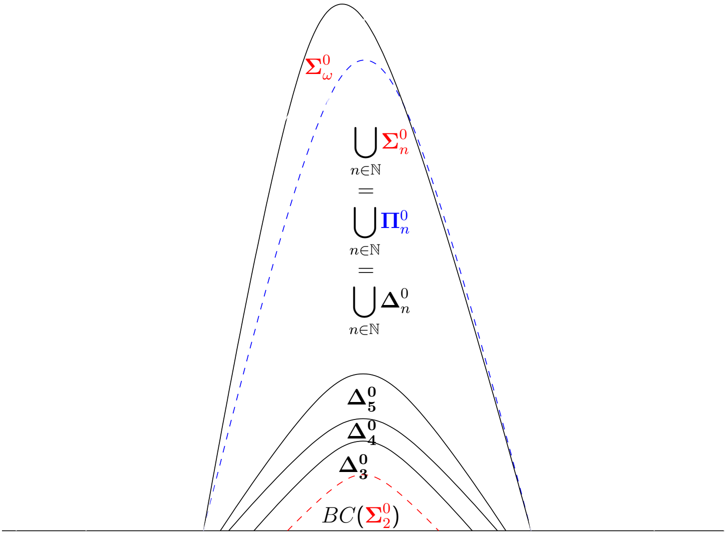


The Wadge Hierarchy

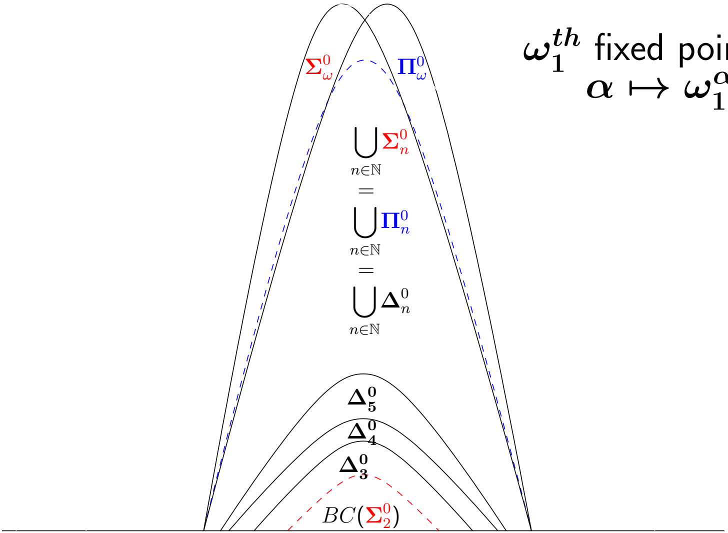


$$\sup_{n<\omega} \underbrace{\omega_1^{\cdot^{\cdot^{\cdot^{\omega_1}}}}}_{n}$$

The Wadge Hierarchy



The Wadge Hierarchy



ω_1^{th} fixed point of $\alpha \mapsto \omega_1^\alpha$

The Wadge Hierarchy

- Which classes are inhabited by regular languages?
- Height of Wadge Hierarchy restricted to regular languages?
  - On words: ω^ω [Klaus Wagner 1979]
  - On trees:
    - deterministic: $(\omega^\omega)^3 + 3$ [Filip Murlak 2006]
    - non-deterministic: *??????*

this paper: for priorities ≤ 2

$$\geq \varphi_2(0) + 1 = \sup_{n<\omega} \underbrace{\varepsilon_{\varepsilon_{\cdot_{\cdot_{\cdot_{\varepsilon_0}}}}}}_{n} + 1$$

The Wadge Hierarchy

1. $\varepsilon_0 = \sup_{n<\omega} \underbrace{\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^0}}}}}}_{n}$;
2. $\varepsilon_{\alpha+1} = \sup_{n<\omega} \underbrace{\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^{(\varepsilon_\alpha+1)}}}}}}}_{n}$;
3. $\varepsilon_\lambda = \sup_{\alpha<\lambda} \varepsilon_\alpha$, for λ some limit ordinal.

$$\varphi_2(0) = \sup_{n<\omega} \underbrace{\varepsilon_{\cdot_{\cdot_{\cdot_{\varepsilon_0}}}}}_{n}$$

The Wadge Hierarchy

Another characterisation of $\varphi_2(0)$:

- $\varphi_2(0)$ is the *first non-null ordinal closed under*

1. $x, y \longmapsto x + y.$
2. $x \longmapsto \omega^x$ *or equiv.* $x \longmapsto (\omega^\omega)^x$
3. $x \longmapsto \varepsilon_x$

$$\varphi_2(0) = \sup_{n<\omega} \underbrace{\varepsilon_{\cdot_{\cdot_{\cdot_{\varepsilon_0}}}}}_{n}$$

Conciliatory Tree Automata

Conciliatory Tree Automata

- *conciliatory* tree: finite or infinite (binary) tree
- For *conciliatory* languages L, M we define the *conciliatory* version of the Wadge game: $\mathbb{C}(L, M)$
  1. rules are similar, except that both players are allowed to skip and to produce trees with finite branches - or even finite trees.
  2. $L \leq_c M$ if and only if $I\!I$ has a winning strategy in the game $\mathbb{C}(L, M)$.
  3. $L \equiv_c M$ if and only if $L \leq_c M$ and $M \leq_c L$
  4. The conciliatory hierarchy is thus the partial order induced by $\leq_c$ on the equivalence classes given by $\equiv_c$.

Conciliatory Tree Automata

- Any *conciliatory* language L over alphabet A yields L^b of *full trees* over alphabet $A \cup \{b\}$:

$$L^b = \left\{t \in T_{A\cup\{b\}} : t_{[\ /b]} \in L\right\}$$

where

1. b is an extra symbol that stands for "blank", and
2. $t_{[\ /b]}$, the *undressing* of t, is informally the conciliatory tree over A obtained once all the occurences of b have been removed in a top-down manner

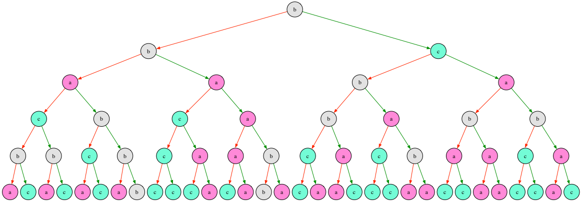
Lemma

Given L, M *any conciliatory languages,*

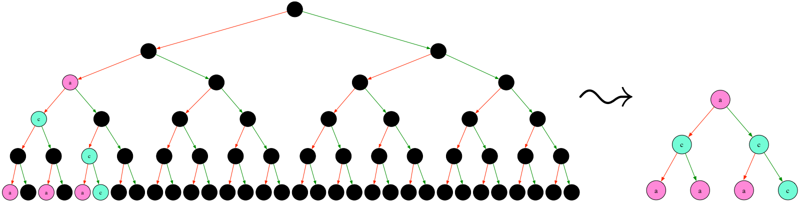
$$L \leq_c M \iff L^b \leq_W M^b$$

Conciliatory Tree Automata

- An infinite tree t with blanks



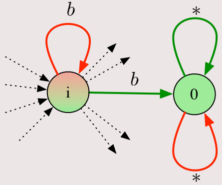
- The blanks are deleted in a top-down manner



Conciliatory Tree Automata

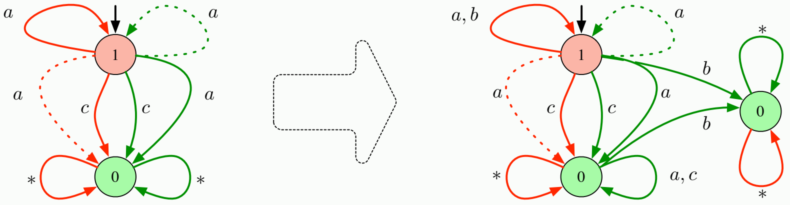
Definition (tree^b automaton)

A tree^b automaton $\mathcal{A}^b$ on the alphabet $A \cup \{b\}$ is any tree automaton s.t. for each state the only possible b-transition is of the form:



Conciliatory Tree Automata

Example



Conciliatory Tree Automata

Definition (conciliatory tree automaton)

A conciliatory tree automaton $\mathcal{A}$ on the alphabet A is a regular tree automaton, except that it also accepts or rejects conciliatory trees per

$$\mathcal{A} \text{ accepts } t \iff \mathcal{A}^b \text{ accepts } t^b.$$

where t^b is the full tree on $A \cup \{b\}$ obtained by completing the missing branches with b's.

Proposition

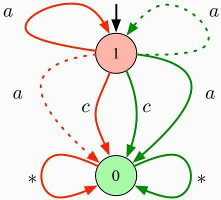
Given any conciliatory tree automaton $\mathcal{A}$, and any conciliatory language L over A,

$$\mathcal{A} \text{ recognizes } L \iff \mathcal{A}^b \text{ recognizes } L^b.$$

Conciliatory Tree Automata

Example

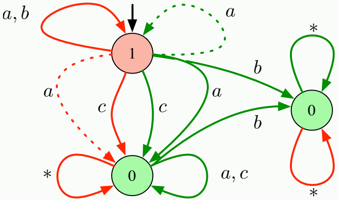
conciliatory language



$\mathcal{A}$ conciliatory

$\left\{ t \mid t \text{ contains } \text{c} \right\}$

full tree language



$\mathcal{A}^b$ (full description)

$\left\{ t \mid t_{[\ /b]} \text{ contains } \text{c} \right\}$

Outline

Operations on Conciliatory Tree Automata

Operations on Conciliatory Tree Automata

1. sum:

$$\mathcal{A}, \mathcal{B} \longrightarrow \mathcal{A} + \mathcal{B}$$

2. multiplication:

$$\mathcal{A}, \alpha \longrightarrow \mathcal{A} \bullet \alpha$$

3. pseudo-exponentiation:

$$\mathcal{A} \longrightarrow (\omega^{\omega})^{\mathcal{A}}$$

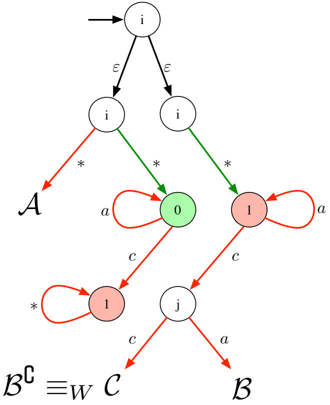
4. The pseudo-epsilon:

$$\mathcal{A} \longrightarrow \varepsilon_{\mathcal{A}}$$

Operations on Conciliatory Tree Automata

- The sum $(+)$

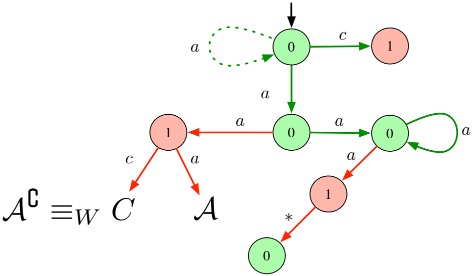
$\mathcal{B}+\mathcal{A} \quad = $



Operations on Conciliatory Tree Automata

- The multiplication ($\bullet$) by ω

$$\mathcal{A}\bullet\omega \quad = $$



Operations on Conciliatory Tree Automata

- Multiplication by any integers $0 < n < \omega$

$$\mathcal{A} \bullet n = \underbrace{\mathcal{A} + \mathcal{A} + \cdots + \mathcal{A}}_{n}$$

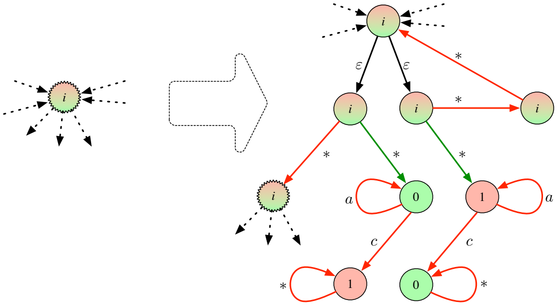
- Yields the multiplication ($\bullet$) by any ordinals $0 < \alpha < \omega^\omega$

Example

$$\mathcal{A} \bullet (\omega^2 + \omega \cdot 3 + 1) = (\mathcal{A} \bullet \omega) \bullet \omega + (\mathcal{A} \bullet \omega) \bullet 3 + \mathcal{A}$$

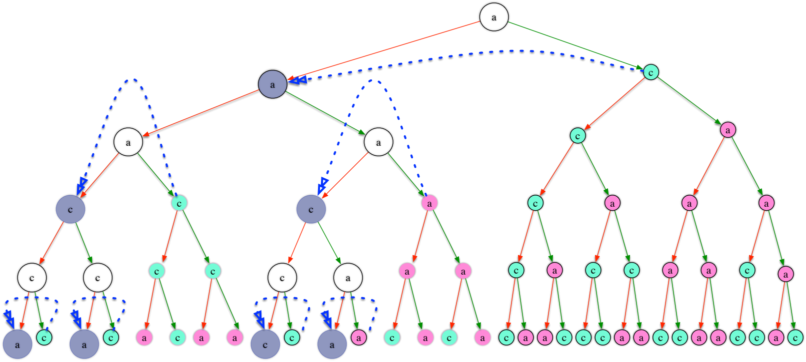
Operations on Conciliatory Tree Automata

- The pseudo-exponentiation $\left((\omega^{\omega})^{\mathcal{A}}\right)$



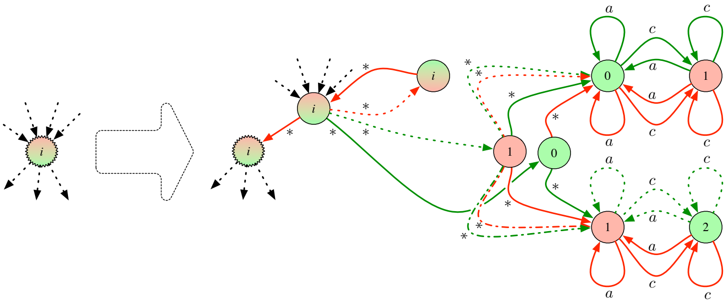
Operations on Conciliatory Tree Automata

- The pseudo-exponentiation $\left((\omega^{\omega})^{\mathcal{A}}\right)$



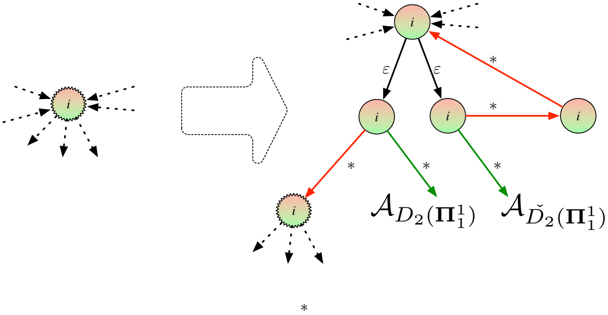
Operations on Conciliatory Tree Automata

- The pseudo-epsilon ($\varepsilon_{\mathcal{A}}$)



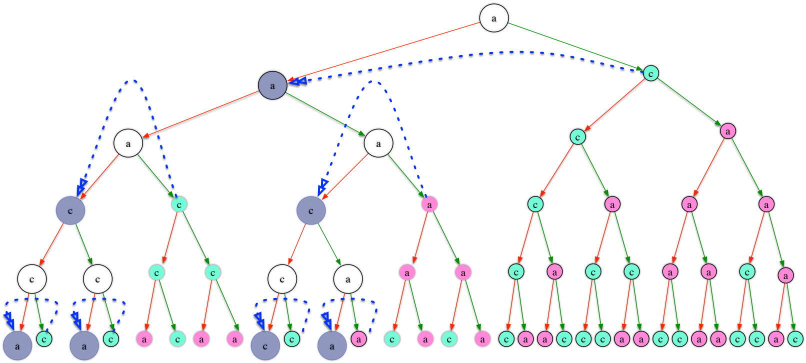
Operations on Conciliatory Tree Automata

- The pseudo-epsilon ($\varepsilon_{\mathcal{A}}$)



Operations on Conciliatory Tree Automata

- The pseudo-epsilon ($\varepsilon_{\mathcal{A}}$)



Operations on Conciliatory Tree Automata

- In the following we endow $<_c$ with the stronger meaning:

$$L <_c M \;\Leftrightarrow\; L \leq_c M \text{ and } \boxed{I \text{ has a w.s. in } \mathbb{C}(M, L)} \;\; (\text{hence } M \not\leq_c L)$$

Operations on Conciliatory Tree Automata

$+$ preserves
1. $^{\complement}$
2. $\leq_c$
3. $<_c$

Proposition

1. $$(\mathcal{A}+\mathcal{B})^{\complement} \equiv_c \mathcal{A}^{\complement}+\mathcal{B}^{\complement}$$

- $\mathcal{A} \leq_c \mathcal{A}'$ *and* $\mathcal{B} \leq_c \mathcal{B}'$ $\Rightarrow$

2. $$\mathcal{A}+\mathcal{B} \leq_c \mathcal{A}'+\mathcal{B}'$$

- $\mathcal{A} \leq_c \mathcal{A}'$ *and* $\mathcal{B} <_c \mathcal{B}'$ $\Rightarrow$

3. $$\boxed{\mathcal{A}+\mathcal{B} <_c \mathcal{A}'+\mathcal{B}'}$$

Operations on Conciliatory Tree Automata

$\bullet$ preserves $<_c$

Proposition

- $0 < \alpha < \beta < \omega^\omega \Rightarrow$
  1. $\mathcal{A} \bullet \alpha <_c \mathcal{A} \bullet \beta$
- $\mathcal{A} <_c \mathcal{B} \Rightarrow$
  2. $\mathcal{A} \bullet \alpha <_c \mathcal{B} \bullet \alpha$

Operations on Conciliatory Tree Automata

$$\mathcal{A} \longrightarrow (\omega^\omega)^{\mathcal{A}} \quad \text{preserves} \quad \begin{cases} 1 & \complement \\ 2 & \leq_c \\ 3 & <_c \end{cases}$$

Proposition

1. $$((\omega^\omega)^{\mathcal{A}})^{\complement} \equiv_c (\omega^\omega)^{(\mathcal{A}^{\complement})}$$

- $\mathcal{A} \leq_c \mathcal{B} \Rightarrow$

2. $$(\omega^\omega)^{\mathcal{A}} \leq_c (\omega^\omega)^{\mathcal{B}}$$

- $\mathcal{A} <_c \mathcal{B} \Rightarrow$

3. $$(\omega^\omega)^{\mathcal{A}} <_c (\omega^\omega)^{\mathcal{B}}$$

Operations on Conciliatory Tree Automata

$\mathcal{A} \longrightarrow \varepsilon_{\mathcal{A}}$ preserves

1. $\complement$
2. $\leq_c$
3. $<_c$

Proposition

1. $(\varepsilon_{\mathcal{A}})^{\complement} \equiv_c \varepsilon_{(\mathcal{A}^{\complement})}$

- $\mathcal{A} \leq_c \mathcal{B} \Rightarrow$

2. $\varepsilon_{\mathcal{A}} \leq_c \varepsilon_{\mathcal{B}}$

- $\mathcal{A} <_c \mathcal{B} \Rightarrow$

3. $\varepsilon_{\mathcal{A}} <_c \varepsilon_{\mathcal{B}}$

Operations on Conciliatory Tree Automata

any $(\omega^\omega)^c$ is closed under $\begin{cases} ❶ + \\ ❷ \bullet \end{cases}$

Proposition

$\mathcal{A}, \mathcal{B} <_c (\omega^\omega)^c \Rightarrow$

1. $\mathcal{A}+\mathcal{B} <_c (\omega^\omega)^c$

2. *for any ordinal* $0 < \alpha < \omega^\omega$

   $\mathcal{A}\bullet\alpha <_c (\omega^\omega)^c$

Operations on Conciliatory Tree Automata

any $\varepsilon_{\mathcal{C}}$ is closed under
1. $+$
2. $\bullet$
3. (ω^ω)

Proposition

$\mathcal{A}, \mathcal{B} <_c \varepsilon_{\mathcal{C}} \Rightarrow$

1. $$\mathcal{A} + \mathcal{B} <_c \varepsilon_{\mathcal{C}}$$
2. *for any ordinal* $0 < \alpha < \omega^\omega$
$$\mathcal{A} \bullet \alpha <_c \varepsilon_{\mathcal{C}}$$
3. $$(\omega^\omega)^{\mathcal{A}} <_c \varepsilon_{\mathcal{C}}$$

Operations on Conciliatory Tree Automata

pseudo-epsilon ($\varepsilon_{\mathcal{A}}$) is a fixed point of pseudo-exponentiation ($(\omega^{\omega})^{\mathcal{A}}$)

Theorem

$$(\omega^{\omega})^{(\varepsilon_{\mathcal{A}})} \equiv_c \varepsilon_{\mathcal{A}}$$

Outline

1. Automata on Infinite Words
2. Automata on Infinite Trees
3. The Wadge Game
4. The Wadge Hierarchy
5. Operations on Conciliatory Tree Automata
6. **The Wadge Hierarchy of $[0, 2]$-tree languages**

The Wadge Hierarchy of $[0,2]$-tree languages

Every ordinal $\alpha > 0$ admits a unique Cantor Normal Form of base ω^ω:

$$\alpha = (\omega^\omega)^{\alpha_k} \cdot \beta_k + \cdots + (\omega^\omega)^{\alpha_0} \cdot \beta_0$$

where

1. $k < \omega$
2. $\alpha_k > \cdots > \alpha_0$
3. $0 < \beta_i < \omega^\omega$.

The Wadge Hierarchy of $[0, 2]$-tree languages

For every $0 < \alpha < \varphi_2(0)$, inductively define $(\mathcal{A}_\alpha, \bar{\mathcal{A}}_\alpha)$:
If the Cantor Normal Form of base ω^ω of α is

$$\alpha = (\omega^\omega)^{\alpha_k} \cdot \beta_k + \cdots + (\omega^\omega)^{\alpha_0} \cdot \beta_0$$

set

$$\boxed{\mathcal{A}_\alpha = \mathcal{A}_{(\omega^\omega)^{\alpha_k}} \bullet \beta_k + \cdots + \mathcal{A}_{(\omega^\omega)^{\alpha_0}} \bullet \beta_0}$$

and

$$\boxed{\bar{\mathcal{A}}_\alpha = \mathcal{A}_{(\omega^\omega)^{\alpha_k}} \bullet \beta_k + \cdots + \bar{\mathcal{A}}_{(\omega^\omega)^{\alpha_0}} \bullet \beta_0}$$

where $\mathcal{A}_{(\omega^\omega)^{\alpha_i}}$ and $\bar{\mathcal{A}}_{(\omega^\omega)^{\alpha_i}}$ are respectively:

1. $\ominus$ and $\oplus$ if $\alpha_i = 0$
2. $(\boldsymbol{\omega}^{\boldsymbol{\omega}})^{\mathcal{A}_{\alpha_i}}$ and $(\boldsymbol{\omega}^{\boldsymbol{\omega}})^{\bar{\mathcal{A}}_{\alpha_i}}$ if $\alpha_i < (\omega^\omega)^{\alpha_i}$
3. $\boldsymbol{\varepsilon}_{\mathcal{A}_{2+\beta}}$ and $\boldsymbol{\varepsilon}_{\bar{\mathcal{A}}_{2+\beta}}$ if $\alpha_i = (\omega^\omega)^{\alpha_i}$ and $\alpha_i = \varepsilon_\beta$ for some $\beta < \alpha_i$.

The Wadge Hierarchy of $[0,2]$-tree languages

Lemma

Let $0 < \alpha < \beta < \varphi_2(0)$,

1. $\mathcal{A}_\alpha$ *and* $\bar{\mathcal{A}}_\alpha$ *are* **effective** *and* $[\mathbf{0},\mathbf{2}]$;
2. I *has a w.s. in both* $\mathbb{C}(\mathcal{A}_\alpha, \bar{\mathcal{A}}_\alpha)$ *and* $\mathbb{C}(\bar{\mathcal{A}}_\alpha, \mathcal{A}_\alpha)$

   $$\Rightarrow$$

   $\bar{\mathcal{A}}_\alpha$ *and* $\mathcal{A}_\alpha$ *are* $\leq_c$*-incomparable;*
3. - $\mathcal{A}_\alpha <_c \mathcal{A}_\beta$
   - $\bar{\mathcal{A}}_\alpha <_c \mathcal{A}_\beta$
   - $\mathcal{A}_\alpha <_c \bar{\mathcal{A}}_\beta$
   - $\bar{\mathcal{A}}_\alpha <_c \bar{\mathcal{A}}_\beta$;
4. *all the underlying winning strategies are* **effective**.

The Wadge Hierarchy of $[0, 2]$-tree languages

Theorem

There exist two families $\left(\mathcal{A}_\alpha{}^b\right)_{\alpha<\varphi_2(0)}$ *and* $\left(\bar{\mathcal{A}}_\alpha{}^b\right)_{\alpha<\varphi_2(0)}$ *of parity tree automata s.t.*

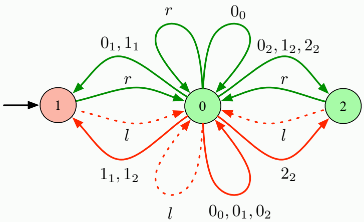
1. *they recognize languages of full trees over the alphabet* $\{a, b, c\}$*;*
2. *their priorities are restricted to* $[\mathbf{0}, \mathbf{2}]$*;*
3. $\alpha < \beta \Leftrightarrow \mathcal{A}_\alpha{}^b, \bar{\mathcal{A}}_\alpha{}^b <_W \mathcal{A}_\beta{}^b, \bar{\mathcal{A}}_\beta{}^b$*;*
4. $\mathcal{A}_\alpha{}^b$ *and* $\bar{\mathcal{A}}_\alpha{}^b$ *are* $\leq_W$*-incomparable;*
5. *all the underlying winning strategies are* **effective**.

The Wadge Hierarchy of $[0, 2]$-tree languages

- A maximal $[0, 2]$-automaton: $W_{[0,2]}$ is any automaton that can simulate *every* parity game with priorities among $[0, 2]$.

Example

The alphabet is $\{l, r, 0_0; 0_1; 0_2; 1_1; 1_2; 2_2\}$



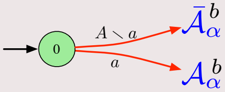
Proposition

$W_{[0,2]}$ is a fixed point of the pseudo-epsilon: $\varepsilon_{W_{[0,2]}} \equiv_c W_{[0,2]}$

The Wadge Hierarchy of $[0, 2]$-tree languages

Definition

The automaton $\mathcal{B}_\alpha$



The Wadge Hierarchy of $[0,2]$-tree languages

Conjecture

Let $\mathcal{A}$ be some nondeterministic $[0,2]$-tree automaton, exactly one of the following holds:

1. $\mathcal{A} \equiv_W W_{[0,2]}$
2. $\mathcal{A} \equiv_W {\mathcal{A}_\alpha}^b$ *(for some $0 < \alpha < \Omega$)*
3. $\mathcal{A} \equiv_W {\bar{\mathcal{A}}_\alpha}^b$ *(for some $0 < \alpha < \Omega$)*
4. $\mathcal{A} \equiv_W \mathcal{B}_\alpha$ *(for some $0 < \alpha < \Omega$)*

The Wadge Hierarchy of $[0, 2]$-tree languages

Theorem

The Wadge hierarchy of $[0, 2]$-tree languages has height at least

$$\varphi_2(0) + 1 = \sup_{n<\omega} \underbrace{\varepsilon_{\varepsilon_{\cdot_{\cdot_{\cdot_{\varepsilon_0}}}}}}_{n} + 1$$

The Wadge hierarchy of $[0, 2]$-word languages has height ω^ω

The Wadge hierarchy of **det.** $[0, 2]$-word languages has height $\omega^2 + 1$

Theorem (Filip Murlak)

The Wadge hierarchy of **deterministic** *tree languages has height*

$$(\omega^\omega)^3 + 3$$

The Wadge Hierarchy of $[0,2]$-tree languages

Conjecture

The Wadge hierarchy of $[0,2]$-tree languages has height exactly

$$\varphi_2(0) + 1 = \sup_{n<\omega} \underbrace{\varepsilon_{\cdot_{\cdot_{\cdot_{\varepsilon_0}}}}}_{n} + 1$$

Wrong!

It is at least

$$\varphi_3(0) + 1 = \sup_{n<\omega} \underbrace{\varphi_2(\varphi_2(\ldots(\varphi_2(0)\ldots)}_{n} + 1$$

The Wadge Hierarchy of $[0,2]$-tree languages

Conjecture

The Wadge hierarchy of $[0,2]$-tree languages has height exactly

$$\varphi_2(0)+1=\sup_{n<\omega}\underbrace{\varepsilon_{\varepsilon_{\cdot_{\cdot_{\cdot_{\varepsilon_0}}}}}}_{n}+1$$

Wrong!

It is at least

$$\varphi_3(0)+1=\sup_{n<\omega}\underbrace{\varphi_2(\varphi_2(\ldots(\varphi_2(0)\ldots)}_{n}+1$$

The Wadge Hierarchy of $[0,2]$-tree languages

Conjecture

The Wadge hierarchy of $[0,2]$-tree languages has height exactly

$$\varphi_3(0)+1 = \sup_{n<\omega} \underbrace{\varphi_2(\varphi_2(\ldots(\varphi_2(0)\ldots)}_{n} + 1$$