On the complexity of infinite computations

Damian Niwiński, Warsaw University

joint work with Igor Walukiewicz and Filip Murlak

Newton Institute, Cambridge, June 2006

Infinite computations

- Büchi (1960) and Rabin (1969) used the concept of infinite computations of finite automata to establish the decidability results in logic.
- D. Muller(1960) used similar concepts to analyse asynchronous digital circuits.
- Since 1980s, computer scientists study infinite computations in context of verification of computing systems (reactive, concurrent, open, . . .).

  Non-termination is an expected behaviour.
- Mathematicians have been playing infinite games since the 1930s (Banach–Mazur, later Gale–Stewart, . . .)

Complexity of finite computations

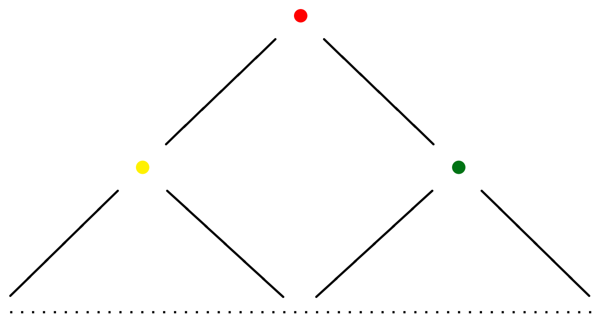
Finitary decision problem

$$A \subseteq \{0,1\}^* \approx \omega.$$

Classical complexity theory studies only decidable problems, in terms of the computation's time and space.

Complexity of infinite computations

An infinite computation can recognise an infinite string, or an infinite tree.



Such an object can be encoded as $f \in \omega^\omega \approx \mathcal{R}$.

Can we ask complexity questions about infinite computations ?

A problem is difficult if it cannot be defined

- by certain computation model,
- by certain logic ($\longrightarrow$ descriptive complexity).

- Which of the two problems is more difficult than the other ?
- Can we characterise/ recognise difficult problems ?

Example: M.O.Rabin discovered

Büchi tree automata $<$ Rabin tree automata

Question: Can we decide when a given Rabin automaton is equivalent to a Büchi automaton ?

BTW, the idea of the Rabin 1970 counter-example can be traced back to the discoveries of Suslin 1916...

$\longrightarrow$ classical definability theory.

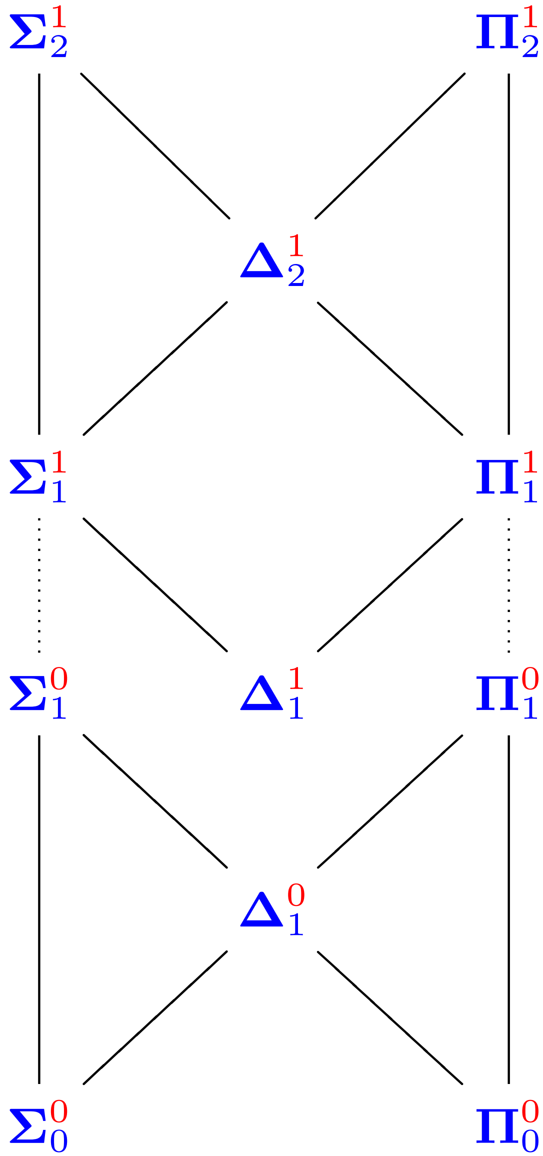
Classical definability theory

1900 Borel, Baire, Lebesgues

1917 Lusin, Suslin

1929 Tarski, Kuratowski

1940 Mostowski, Kleene



Classical hierarchies

of relations $r(\alpha;\beta) \subseteq \omega^k \times (\omega^\omega)^\ell$, defined by formulas $\varphi(\mathbf{x};\mathbf{y})$

Arithmetical hierarchy

$\Sigma^0_0 = \Pi^0_0 =$ bounded quantification

$\Sigma^0_{n+1} = \{\exists z\, \varphi(z,\mathbf{x};\mathbf{y}) : \varphi \in \mathbf{\Pi^0_n}\}$

$\Pi^0_{n+1} = \{\forall z\, \varphi(z,\mathbf{x};\mathbf{y}) : \varphi \in \mathbf{\Sigma^0_n}\}$

Analytical hierarchy

$\Sigma^1_0 = \Pi^1_0 =$ first order quantification

$\Sigma^1_{n+1} = \{\exists f\, \varphi(\mathbf{x};\mathbf{f},\mathbf{y}) : \varphi \in \mathbf{\Pi^1_n}\}$

$\Pi^1_{n+1} = \{\forall f\, \varphi(\mathbf{x};\mathbf{f},\mathbf{y}) : \varphi \in \mathbf{\Sigma^1_n}\}$

Boldface hierarchies (Borel/projective) obtained by introducing parameters from ω^ω.

Remarkable power of finite-state recognisability of infinite objects

Finitary problems beyond Σ^1_0 are considered as highly uncomputable.

- The first-order theory of the standard model of arithmetics is in Δ^1_1, but not in Σ^0_n, for any n.

In contrast,

- An ω-language

  $\{u \in \{a, b\}^\omega$: there are finitely many b's $\}$

  is in $\mathbf{\Sigma^0_2}$ but not in $\mathbf{\Pi^0_2}$.

- A tree language

  $\{t \in \{a, b\}^{\{l,r\}^*}$: on each path, there are finitely many b's $\}$

  is in $\mathbf{\Pi^1_1}$ but not in $\mathbf{\Sigma^1_1}$.

Still, finite-state automata can recognise these sets !

Remarkable (?) power of finite-state recognisability of infinite objects

It would be misleading to compare the properties of integers and the properties of reals with the same complexity measure !

But still. . .

- Regular sets of finite words/trees constitute the simplest level of Δ^0_1 ($\mathcal{O}(1)/\mathcal{O}(\log n)$ space).
- Finite-state automata on infinite trees can recognise $\mathbf{\Pi^1_1}$ and $\mathbf{\Sigma^1_1}$ complete sets.

Does the infinite case give an insight into the finite one ?

One of the strongest separation results in complexity theory is

Furst, Saxe, Sipser 1983

$$PARITY \not\in AC_0$$

The idea is based on a previous observation by

Sipser 1983 that ω-PARITY cannot be recognised by a countable circuit.

(Countable circuits recognise only Borel sets, while ω-PARITY is non-measurable.)

Infinite complexity ≈ descriptive (data) complexity of a logic

infinite complexity	expressive power	complexity of satisfiability
$\{t : t \models \varphi\}$	φ	$\overset{?}{\models} \varphi$

To measure the complexity of infinite computations, we have

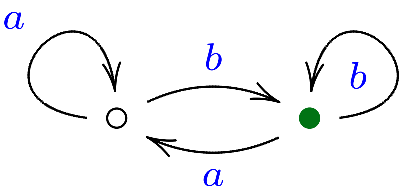
- classical definability hierarchies,
- automata index hierarchies,
- the μ-calculus alternation hierarchy.

In this talk we compare various measures/hierarchies with emphasis on the decidability questions.

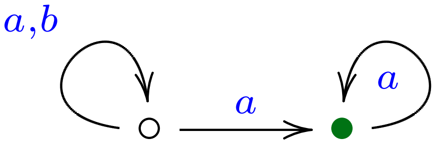
Büchi automata on infinite words

$$\mathcal{A} = \langle \Sigma, Q, q_I, Tr, F \rangle$$

where $Tr \subseteq Q \times \Sigma \times Q$, $F \subseteq Q$.



$$((a+b)^*b)^\omega$$



$$(a+b)^*a^\omega$$

The second one cannot be recognised by a deterministic automaton.

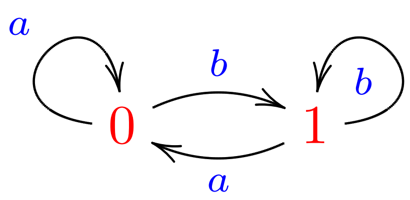
$$\xrightarrow{a}\xrightarrow{b}\xrightarrow{a}\xrightarrow{a}\xrightarrow{b}\xrightarrow{a}\xrightarrow{a}\xrightarrow{a}\xrightarrow{b}\xrightarrow{a}\xrightarrow{a}\xrightarrow{a}\xrightarrow{a}\xrightarrow{b}\xrightarrow{a} \ldots\ldots\ldots \xrightarrow{a}\xrightarrow{b}\xrightarrow{a} \ldots$$

Parity automata

$$\mathcal{A} = \langle \Sigma, Q, q_I, Tr, \textit{rank} \rangle$$

where *rank* : $Q \to \{0, 1, \ldots, k\}$.

$$\limsup_{i \to \infty} \textit{rank}(q_i) \text{ is even}$$



$$(a + b)^* a^\omega$$

The **index** of a parity automaton $\mathcal{A}$ is

$$(\textbf{min}\ \textit{rank}(Q), \textbf{max}\ \textit{rank}(Q))$$

We can assume **min** *rank*$(Q) \in \{0, 1\}$.

The McNaughton Theorem (1966)

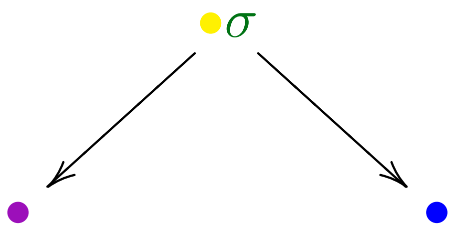
A nondeterministic Büchi automaton can be simulated by a deterministic parity automaton of some index (i, k).

The minimal index (i, k) may be arbitrarily high (Wagner 1979), but can be effectively computed
(in polynomial time, if the input automaton is deterministic N & Walukiewicz 1998, Carton & Maceiras 1999).

Parity tree automata

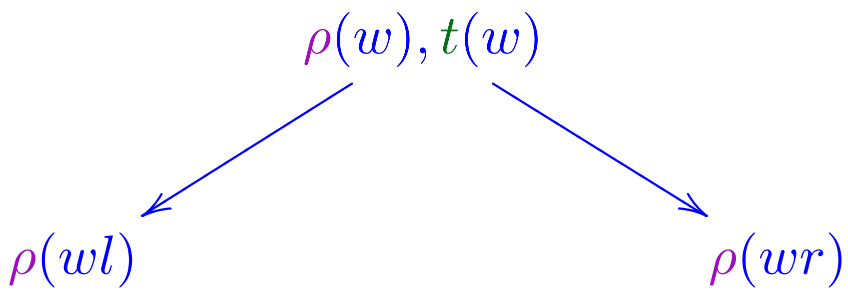
$$\mathcal{A} = \langle \Sigma, Q, q_I, Tr, \mathit{rank} \rangle$$

where $Tr \subseteq Q \times \Sigma \times Q \times Q$, $\mathit{rank} : Q \longrightarrow \{0, 1, \ldots, k\}$.



Parity tree automata ctd.

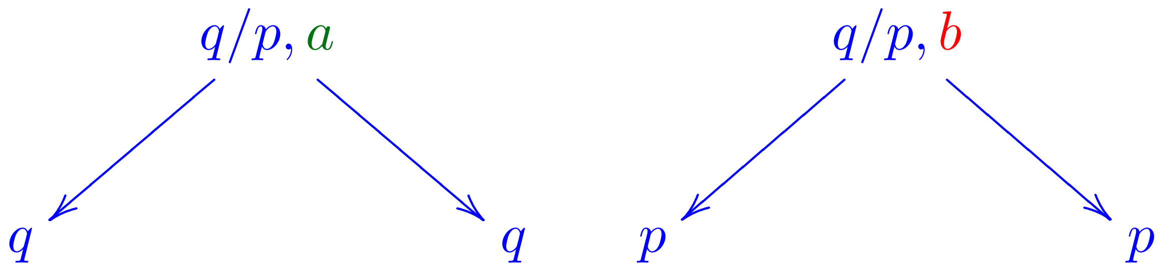
A run of $\mathcal{A}$ on a tree $t : \{l, r\}^* \to \Sigma$ is a tree $\rho : \{l, r\}^* \to Q$, such that, for each $w \in$ dom (ρ), $\langle \rho(w), t(w), \rho(wl), \rho(wr) \rangle \in Tr$



The run is accepting if, for each path $P = p_0 p_1 \ldots \in \{l, r\}^\omega$,

$$\limsup_{k \to \infty} \mathit{rank}(\rho(p_0 p_1 \ldots p_k)) \text{ is even.}$$

Example

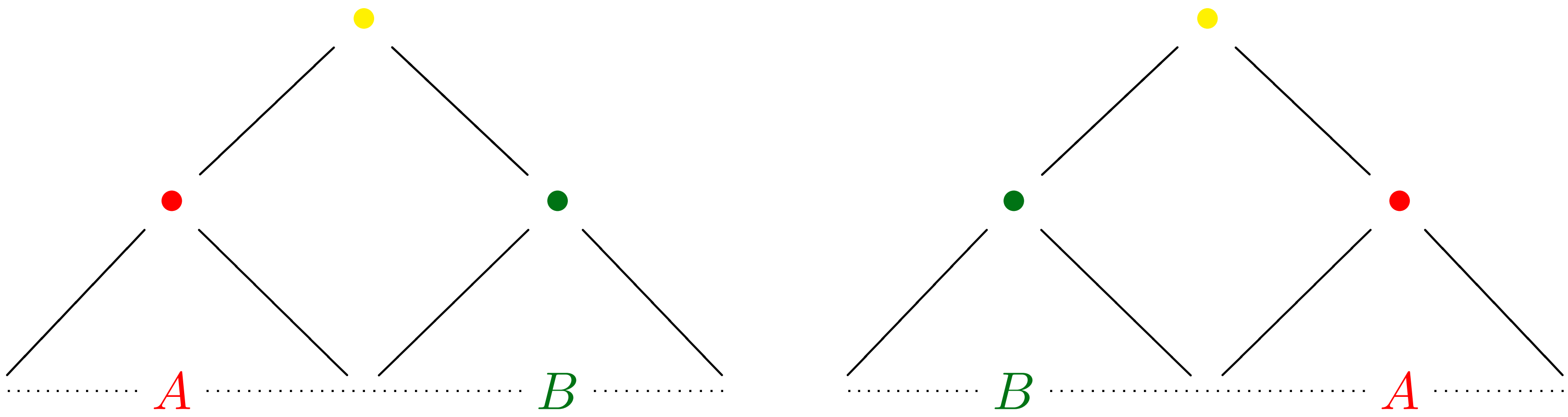


$rank(q) = 0$ $rank(p) = 1$

recognizes the set of trees where, on each branch, b appears only finitely often.

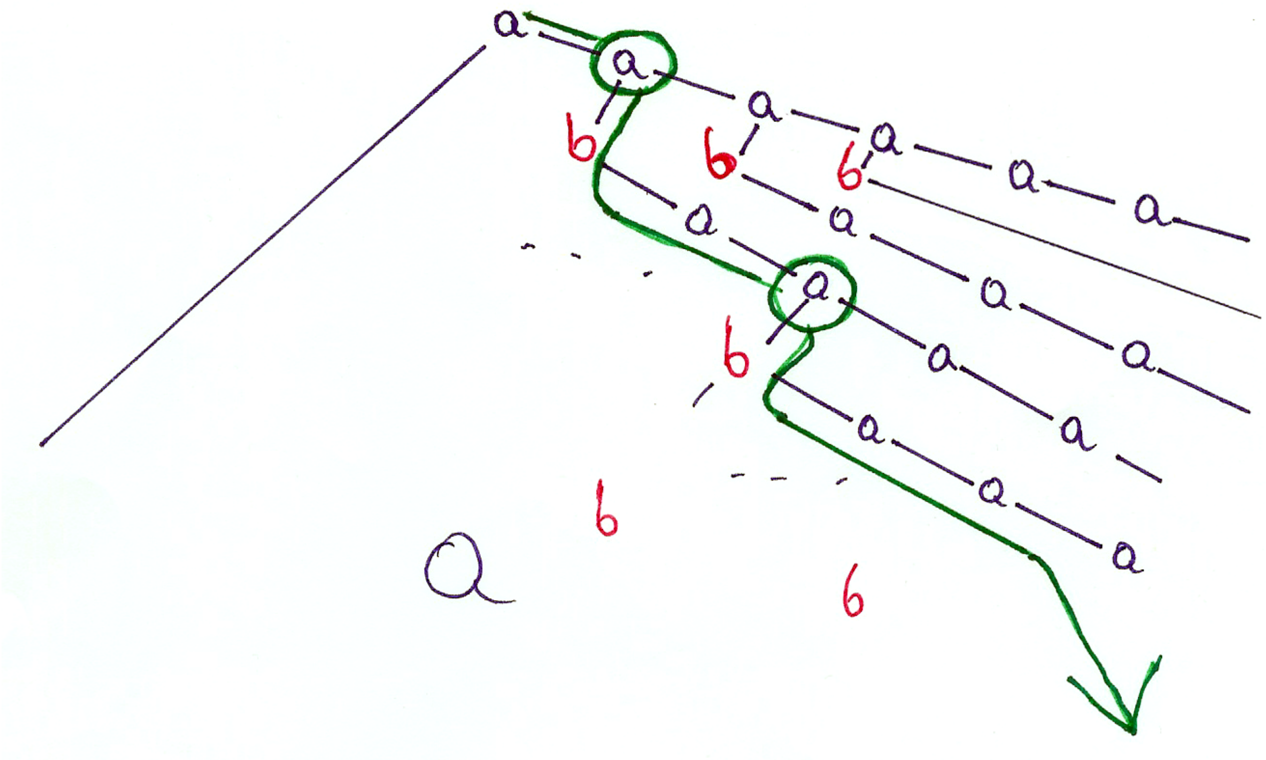
Nondeterminism

For trivial reasons, tree automata cannot be, in general, determinized.

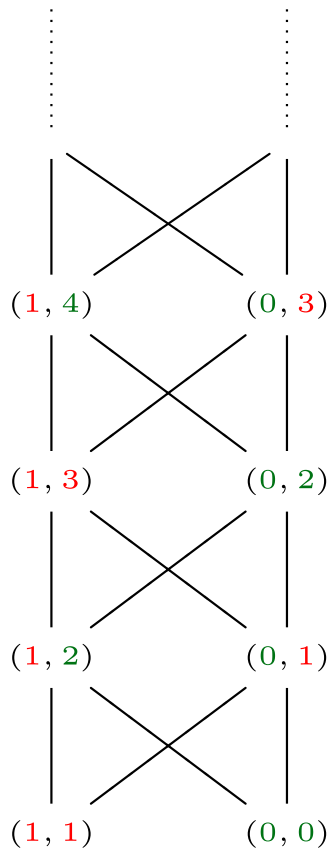


Rabin's counter–example

In contrast to the automata on words, the Büchi condition alone is not sufficient, even in the presence of nondeterminism (Rabin 1970).



The Mostowski index hierarchy



Strict for tree automata : deterministic (essentially Wagner 1979), non-deterministic (N 1986), alternating (Bradfield, Arnold 1999).

The Mostowski index hierarchy ctd.

Languages which witness the strictness of the hierarchy.

For deterministic automata on words :

$$M_{\iota,\kappa} = \{u \in \{\iota, \ldots, \kappa\}^\omega : \limsup_{\ell\to\infty} u_\ell \text{ is even}\}$$

For deterministic/non-deterministic automata on trees :

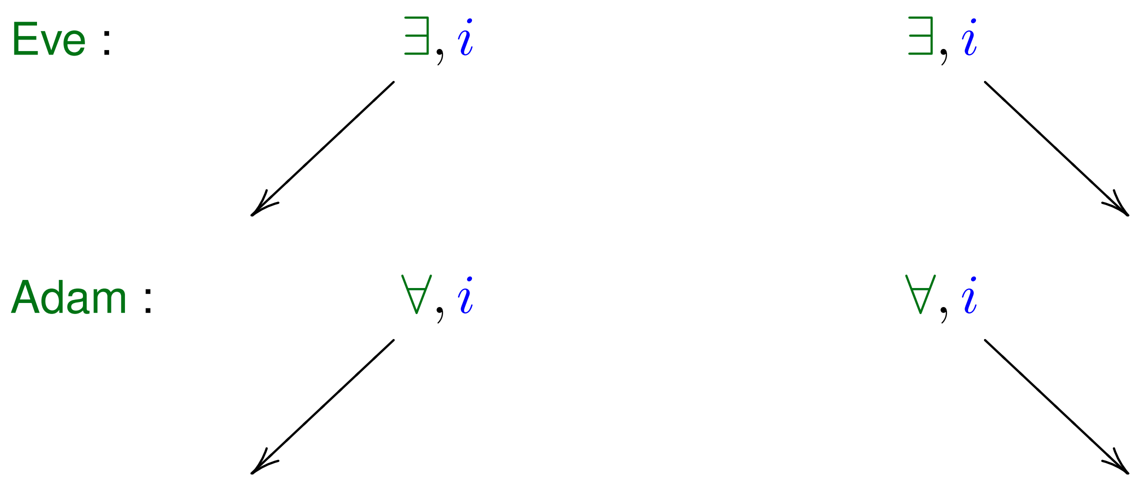
$$T_{\iota,\kappa} = \{t \in \{\iota, \ldots, \kappa\}^{\{l,r\}^*} : \text{ each branch is in } M_{\iota,\kappa}\}$$

For alternating tree automata :

$W_{\iota,\kappa}$ = the “game version” of the above.

Game tree languages

Alphabet : $\{\exists, \forall\} \times \{\iota, \ldots, \kappa\}$.

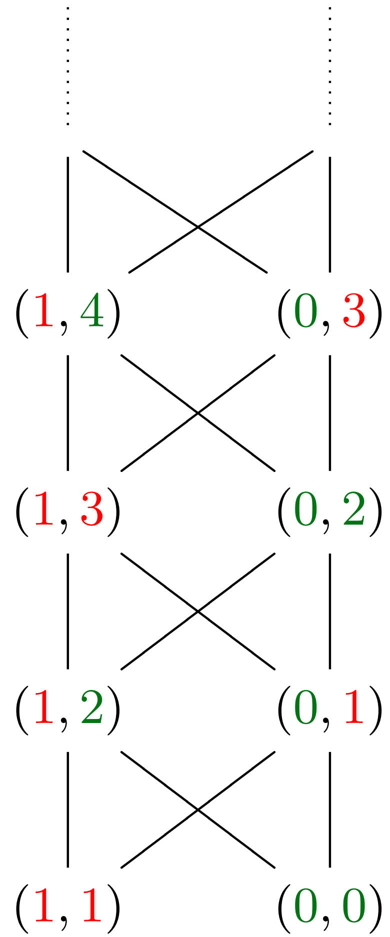


Eve wins an infinite play $(x_0, i_0),\ (x_1, i_1),\ (x_2, i_2), \ldots \quad (x_\ell \in \{\exists, \forall\})$

iff $\limsup_{\ell \to \infty} i_\ell$ is even.

The set $W_{\iota,\kappa}$ consists of all trees such that Eve has a winning strategy.

Can we decide the level of a recognizable tree language in the Mostowski hierarchy ?



We know the answer only if an input automaton is deterministic.

The problem

Given : a deterministic parity tree automaton

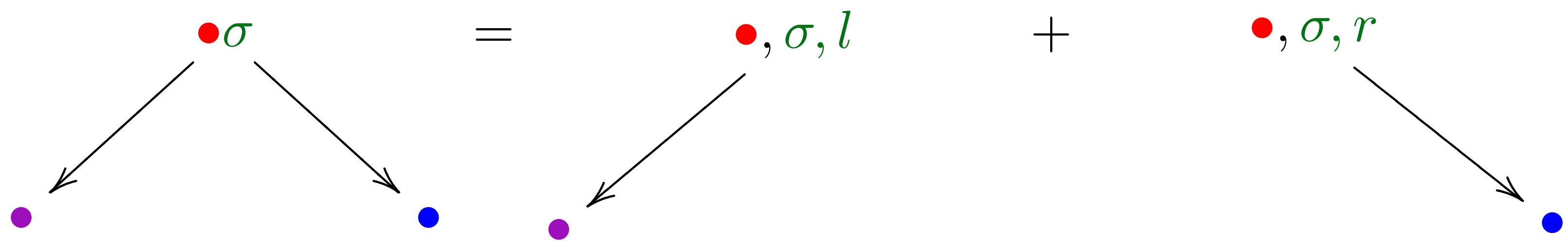
Compute : the minimal Mostowski index of a non-deterministic automaton recognising the same language.

T.Urbański 2000 solved the question $\equiv$ *(non-deterministic) Büchi ?*

N & Walukiewicz 2004 settled the whole non-deterministic hierarchy.

From trees to words : path automata

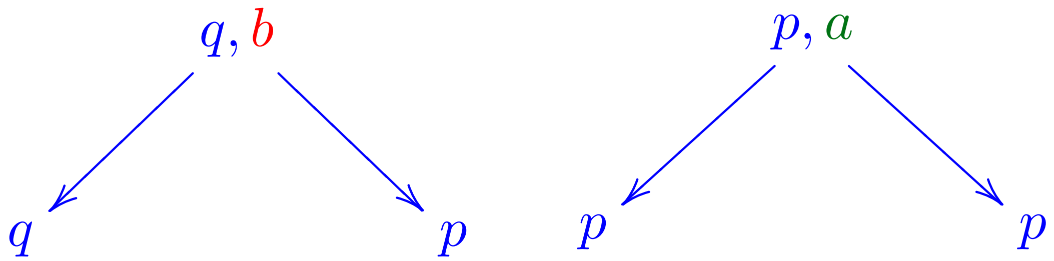
A deterministic tree automaton $\mathcal{A}$ over alphabet Σ can be identified with a deterministic word automaton $\mathcal{A}'$ over alphabet $\Sigma \times \{l, r\}$,



$\mathcal{A}$ recognizes a tree t iff $\mathcal{A}'$ recognizes all paths of t.

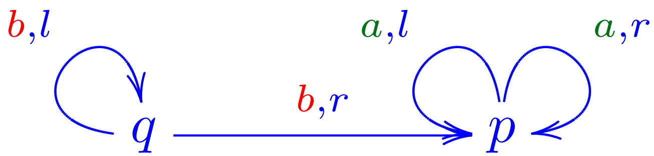
Example

Deterministic tree automaton :



rank$(q) = 0$ *rank*$(p) = 1$

Corresponding path automaton :

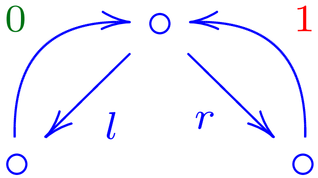
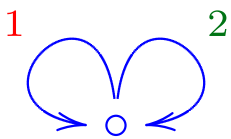
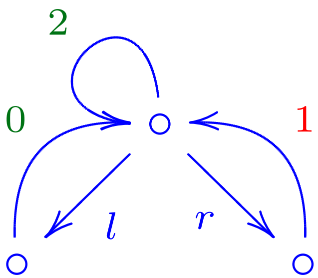
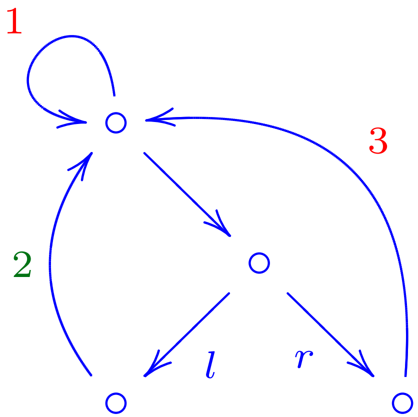


Determinization, whenever possible, is effective

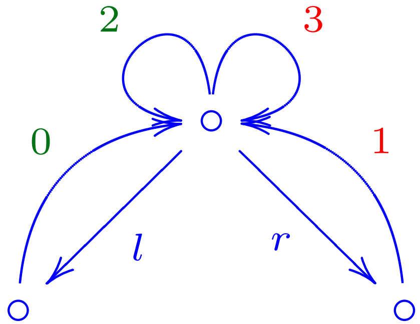
The concept of path automaton allows us to decide (in EXPTIME), if a given non-deterministic tree automaton is equivalent to a deterministic one.

It suffices to verify if

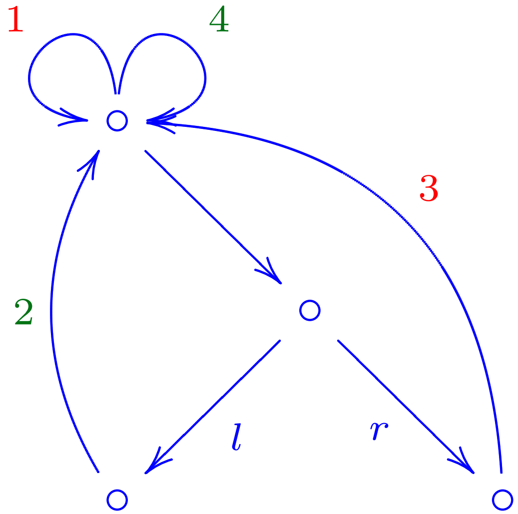
$$L(\mathcal{A}) = Trees(Paths(L(\mathcal{A})))$$

Index class	Forbidden pattern
(1,2)	
(0,1)	
(0,2)	
(1,3)	

The $(0, n)$ case, $n \geq 3$



The $(1, n)$ case, $n \geq 4$



Theorem. Let $\mathcal{A}$ be a deterministic tree automaton.

Then $L(\mathcal{A})$ can be recognised by a non-deterministic tree automaton of index (ι, n) if and only if the corresponding path automaton does not contain any productive $\overline{(\iota, n)}$ pattern.

An idea of the proof.

($\Leftarrow$) Unravel a forbidden pattern into a tree and refine Rabin's argument.

($\Rightarrow$) Decompose $\mathcal{A}$ into strongly connected components, and apply inductive arguments to the sub-automata induced this way.

Corollary. Consequently, the index of a deterministic tree language can be computed within the complexity of computing productive states (i.e., NP $\cap$ co-NP).

Rabin's counter–example revisited

Descriptive complexity argument :

The Büchi recognisable sets of trees are always in $\boldsymbol{\Sigma}^1_1$,
while the Rabin counter–example is $\boldsymbol{\Pi}^1_1$-complete.

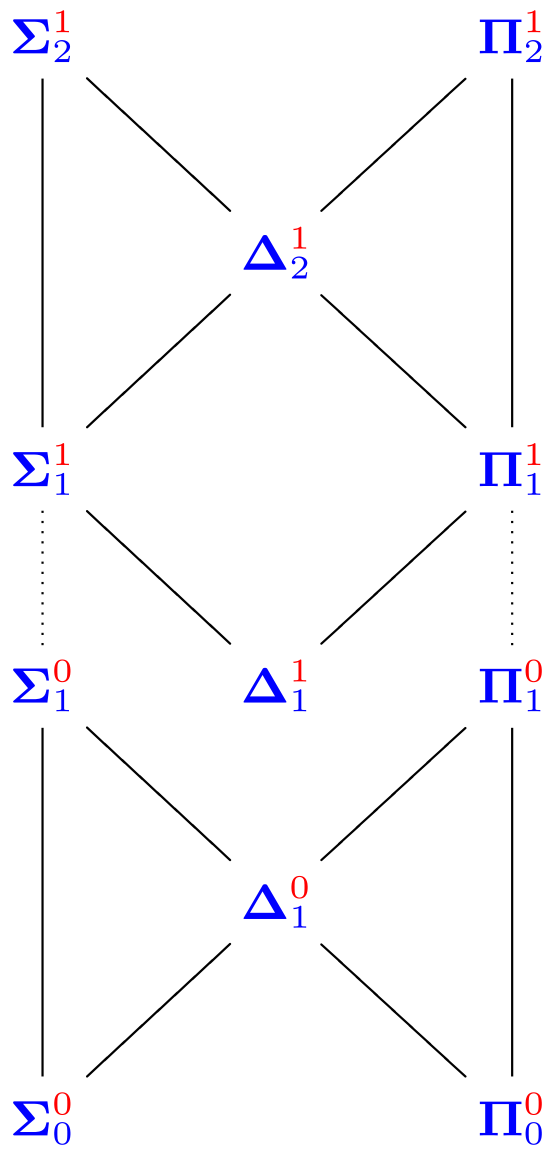
The idea can be traced back to the Suslin 1916 discovery that Borel's sets are not closed under projections.

The set

$$\{\langle T, u\rangle : u \text{ is a branch of } T \text{ with infinitely many } b\text{'s}\}$$

is Borel ($\boldsymbol{\Pi}^0_2$), but its projection is $\boldsymbol{\Sigma}^1_1$-complete .

Can we decide the level of a recognisable tree language in the Borel/projective hierarchies ?



For the case of infinite words, the question was settled already by Wagner 1979.

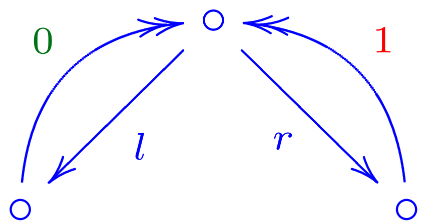
For trees, we can determine the exact level of $\mathcal{T}(\mathcal{A})$, provided that $\mathcal{A}$ is a deterministic automaton

(N & Walukiewicz 2003, Murlak 2005).

Non-deterministic case is completely open.

Criterion : forbidden patterns

If a path automaton $\mathcal{A}'$ contains a (productive) pattern



then $\mathcal{T}(\mathcal{A})$ is $\mathbf{\Pi}^1_1$-complete, hence non-Borel.

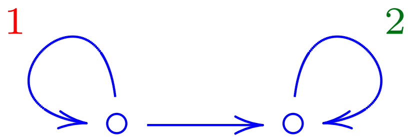
Otherwise it is in $\mathbf{\Pi}^0_3$ (N & Walukiewicz 2003).

Dichotomy!

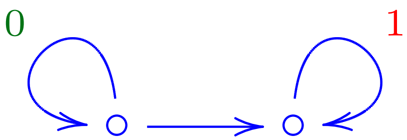
The algorithm of detecting patterns runs in time of solving the non-emptiness problem of parity tree automata (NP $\cap$ co-NP).

F. Murlak 2005 settles the remaining cases :

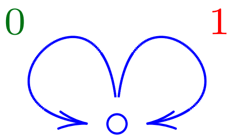
Class | Forbidden pattern

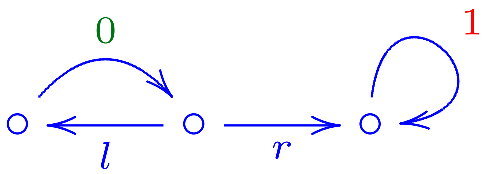
$\mathbf{\Pi}_1^0$ 

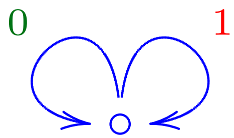
"folklore"

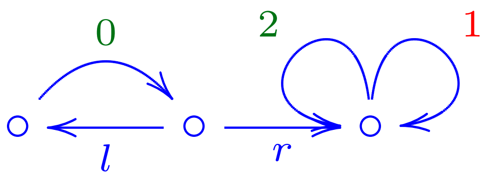
$\mathbf{\Sigma}_1^0$ 

....

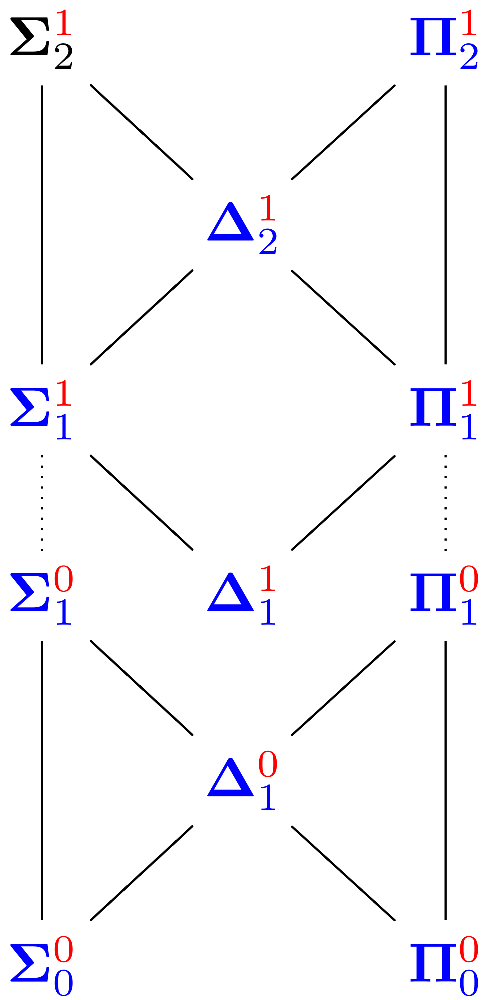
$\mathbf{\Pi}_2^0$ 

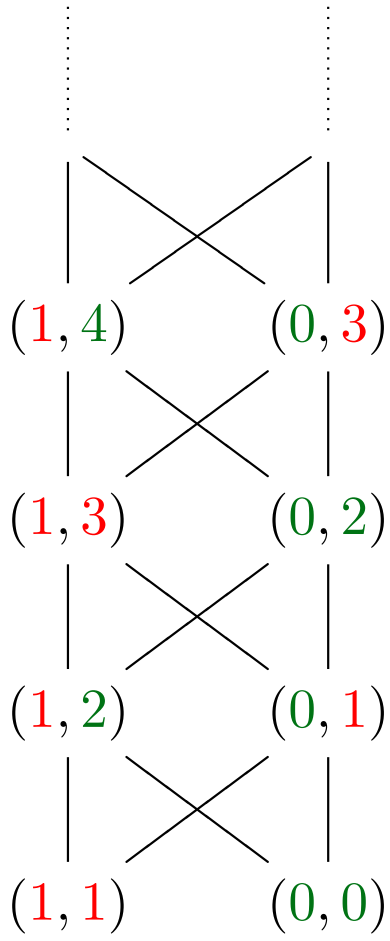
$\mathbf{\Sigma}_2^0$ 



$\mathbf{\Delta}_3^0$ 

Relating the hierarchies





A deterministic tree language is non-Borel iff it is non-Büchi ($\succcurlyeq$ (0,1))

Relating the hierarchies cont'd.

Do the topological hardness and the automata-theoretic hardness always coincide ?

Skurczyński 1993 showed that there are recognisable tree languages on every finite level of the Borel hierarchy, and we now that there are also some $\mathbf{\Sigma^1_1}$ and $\mathbf{\Pi^1_1}$-complete ones.

For non-deterministic languages we only know that if a tree language is $\mathbf{\Pi}^1_1$ hard then it is above the (0,1) level.

The finest topological hierarchy is given by the Wadge reducibility

Let $\mathcal{T}_1, \mathcal{T}_2$ be topological spaces.

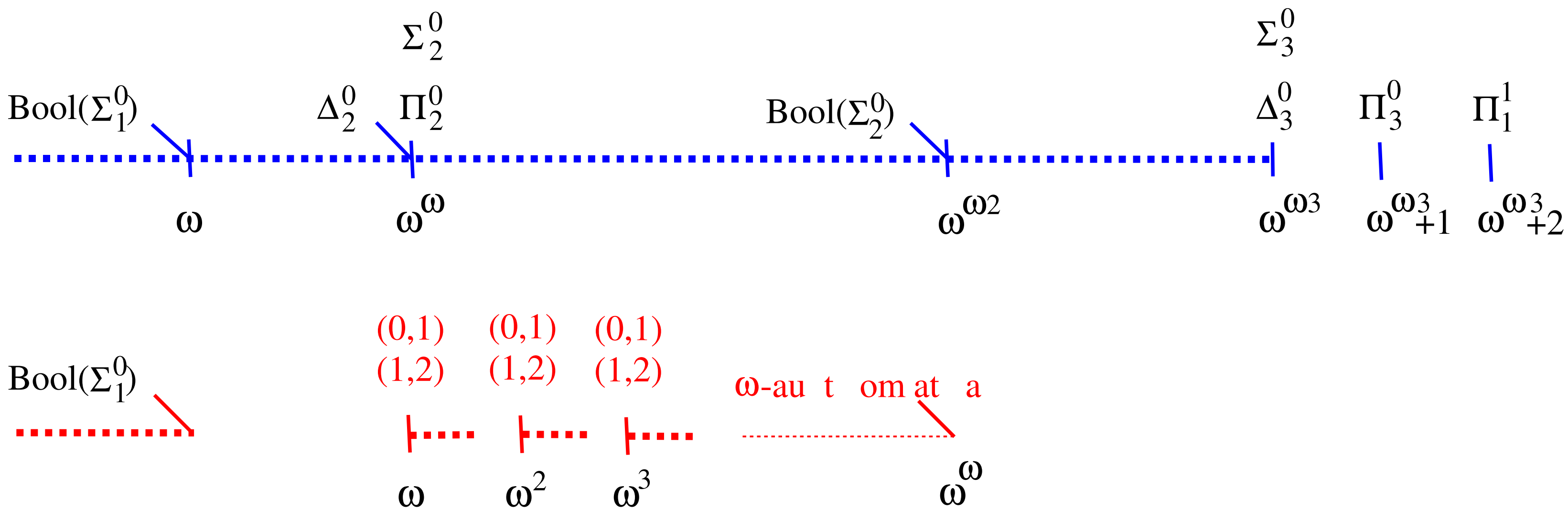
$A \subseteq \mathcal{T}_1$ is Wadge reducible to $B \subseteq \mathcal{T}_2$, in symbols $A \leq_w B$,
if there is a continuous reduction $\varphi : \mathcal{T}_1 \to \mathcal{T}_2$,

$$(\forall \tau \in \mathcal{T}_1)\ \tau \in A \iff \varphi(\tau) \in B.$$

K.Wagner 1979 completely described the Wadge hierarchy of ω-languages of words (height ω^ω).

F.Murlak 2006 completely described the Wadge hierarchy of deterministic languages of trees (height $\omega^{\omega \cdot 3} + 2$).

Wadge hierarchy for deterministic tree languages (Murlak 2006)



- The height is $\omega^{\omega\cdot 3}+2$ (*vs* ω^ω for word languages).
- Complete sets exist in $\mathbf{\Pi_1^1}$, $\mathbf{\Pi_3^0}$, and surprisingly, in $\mathbf{\Delta_3^0}$.

Wadge reducibility—decidability issues

Fact (Büchi & Landweber 1969). For Büchi automata on infinite words:

(1) If $\mathcal{L}(\mathcal{A}) \leq_w \mathcal{L}(\mathcal{B})$ then there exists a finite–state transducer reducing $\mathcal{L}(\mathcal{A})$ to $\mathcal{L}(\mathcal{B})$.

(2) It is decidable if $\mathcal{L}(\mathcal{A}) \leq_w \mathcal{L}(\mathcal{B})$.

For trees, (1) does not hold. Nevertheless, Murlak 2006 shows

Fact. It is decidable if $\mathcal{T}(\mathcal{A}) \leq_w \mathcal{T}(\mathcal{B})$, for deterministic tree automata $\mathcal{A}$, $\mathcal{B}$.

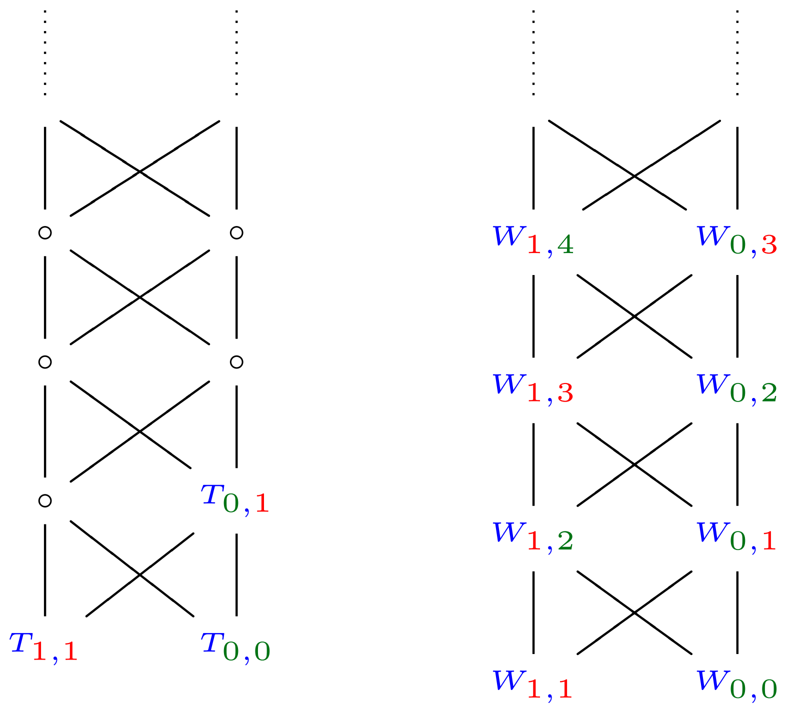
The non-deterministic case remains open.

Beyond the deterministic case—game languages revisited

Recall the hierarchy witness languages $T_{\iota,\kappa}, W_{\iota,\kappa}$. We have

(1) $T_{\iota,\kappa} \leq_w T_{0,1}$, for any ι, κ;

(2) $W_{\iota,\kappa} \leq_w W_{\iota',\kappa'} \Leftrightarrow (\iota,\kappa) \sqsubseteq ((\iota',\kappa')$.



A complete set for all deterministic tree languages

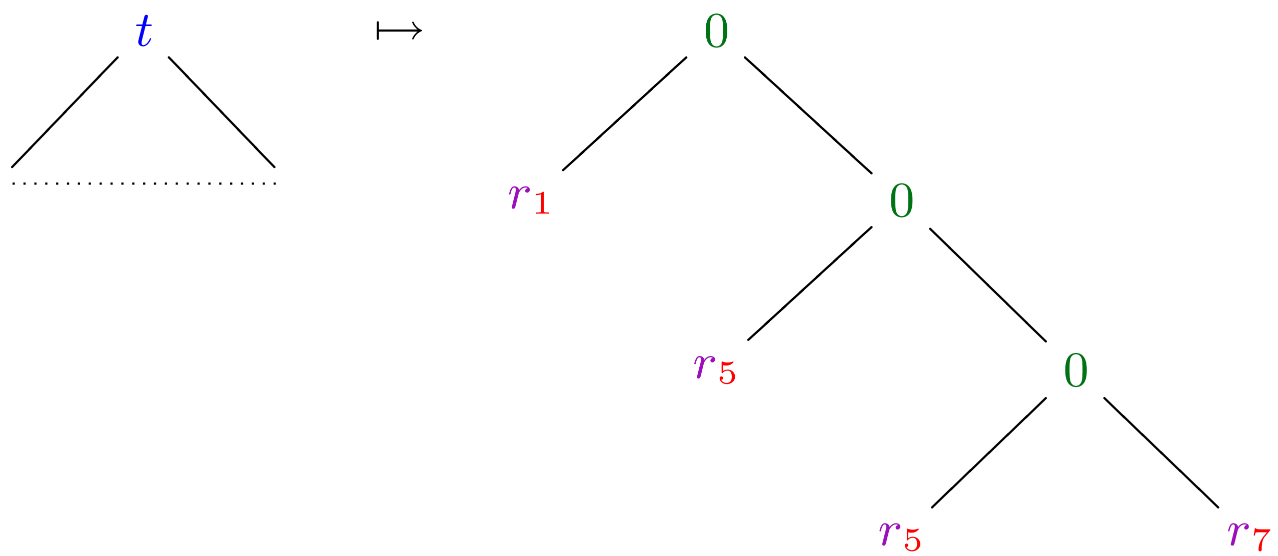
Any deterministically recognisable set of trees is reducible by a transducer to $T_{0,1}$.

We first show a generic reduction of $\mathcal{T}(A)$ to $T_{0,2}$.

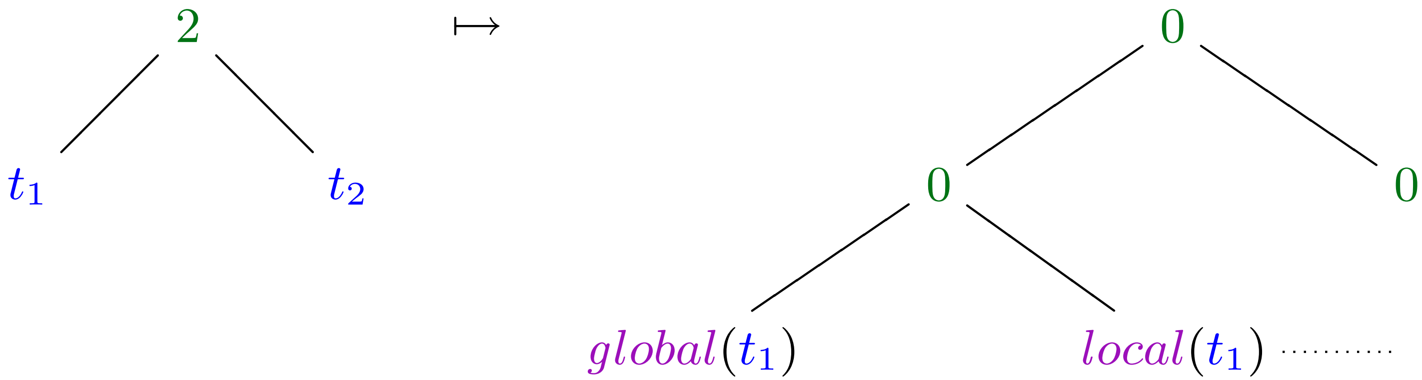
Let r be a unique run of an automaton A on a tree t. For each *odd* $i \leq n$, let

$$r_i(w) = \begin{cases} 0 & \text{if } \textit{rank}\, r(w) < i \\ 1 & \text{if } \textit{rank}\, r(w) = i \\ 2 & \text{if } \textit{rank}\, r(w) > i \end{cases}$$

$$t \mapsto (r_1, 0(r_3, 0(r_5, \ldots 0(r_{2 \cdot \lceil \frac{n}{2} \rceil - 3}, r_{2 \cdot \lceil \frac{n}{2} \rceil - 1}) \ldots)))$$


t
↦
0
r1
0
r5
0
r5
r7

Reduction of $T_{0,2}$ to $T_{0,1}$.



where $local(t_i)$ is t_i reproduced till first 2,

$global(0(t'_1, t'_2)) = global(1(t'_1, t'_2)) = 0(global(t'_1), global(t'_2))$,

$global(2(t'_1, t'_2))$ as above.

$W_{\iota,\kappa}$ form a strict hierarchy—sketch of proof

We identify $T_\Sigma \approx \Sigma^\omega$, and view it as a metric space.

$f : \Sigma^\omega \longrightarrow \Sigma^\omega$ is contracting if

$d(f(t_1), f(t_2)) \leq c \cdot d(t_1, t_2)$, for some constant $0 < c < 1$.

Note that by the Banach Fixed-Point Theorem, no $\overline{L} = \Sigma^\omega - L$ is reducible to L *via* a contracting reduction.

In particular $\overline{W_{\iota,\kappa}} \approx W_{\overline{\iota,\kappa}}$ does not reduce to $W_{\iota,\kappa}$ *via* a contracting reduction.

Neither does $W_{\iota',\kappa'}$ with $(\iota',\kappa') \sqsupseteq \overline{(\iota,\kappa)}$.

Main Lemma If f reduces $W_{\iota,\kappa}$ to some L then there is a mapping $h : \Sigma^\omega \longrightarrow \Sigma^\omega$ (padding), such that

- h reduces $W_{\iota,\kappa}$ to itself,
- $f \circ h$ is contracting.

Recall For any continuous $f : \Sigma^\omega \longrightarrow \Sigma^\omega$, there is $f_* : \Sigma^* \longrightarrow \Sigma^*$, such that, $f(u) = \lim_{n\to\infty} f_*(u \upharpoonright n)$.

Waiting time For any continuous $f : \Sigma^\omega \longrightarrow \Sigma^\omega$,
$wait(f, n) = \min\{k : (\forall v)\ |v| \geq k \implies |f_*(v)| \geq n\}$.

Sub-lemma Let $f, g : \Sigma^\omega \longrightarrow \Sigma^\omega$ be continuous functions satisfying
$|g_*(v)| \geq wait(f, |v| + 1)$,
for all $v \in \Sigma^*$. Then $f \circ g$ is contracting with the constant $c = \frac{1}{2}$.

Yet another link between automata and topology

A.Arnold 1998 showed that

$$\mathcal{T}(A) \quad \leq_w \quad W_{\iota,\kappa},$$

for any alternating automaton of index (ι, κ).

Corollary If $\overline{W}_{\iota,\kappa} \leq_w \mathcal{T}(A)$ then $\mathcal{T}(A)$ cannot be recognised by an alternating automaton of index (ι, κ).

Question: iff ?

Related questions and results

General goal: find a simplest description of an object.

Given : a formula of some logic $\mathcal{L}$.

Question : is it equivalent to a formula of some sub-logic $\mathcal{L}' \subseteq \mathcal{L}$?

....

Given : a formula of the μ-calculus.

Question : Determine its level in the $\mu\nu$-hierarchy.

M. Otto 1999 showed how to decide if μ and ν can be completely eliminated in a formula.

Walukiewicz 2002 settled the μ and ν levels.

What about the next levels ?

Conclusion. In contrast to the finitary case, finite state automata running over infinite words or trees can recognise highly complex properties of infinite computations (e.g., $\mathbf{\Pi}^1_1$-complete).

Automata also provide fine hierarchies, complementary to the classical Borel/projective hierarchies.

For deterministic automata, we can decide its exact level in the complexity hierarchies.

The *non-deterministic* case needs new ideas.

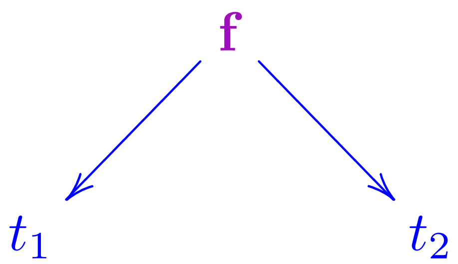
Appendix

The Mostowski hierarchy — relation to the μ-calculus

The set of trees over alphabet $\{a, b\}$ where, on each branch, b appears only finitely often can be presented by

$$\mu z.\nu y.\, a(y, y) \cup b(z, z)$$

where

- $\mu x.t$ is the least fixed point of $x = t(x)$,
- $\nu x.t$ is the greatest fixed point of $x = t(x)$,
- $\mathbf{f}(L_1, L_2) = \{$ 

$: t_1 \in L_1,\ t_2 \in L_2\}$.

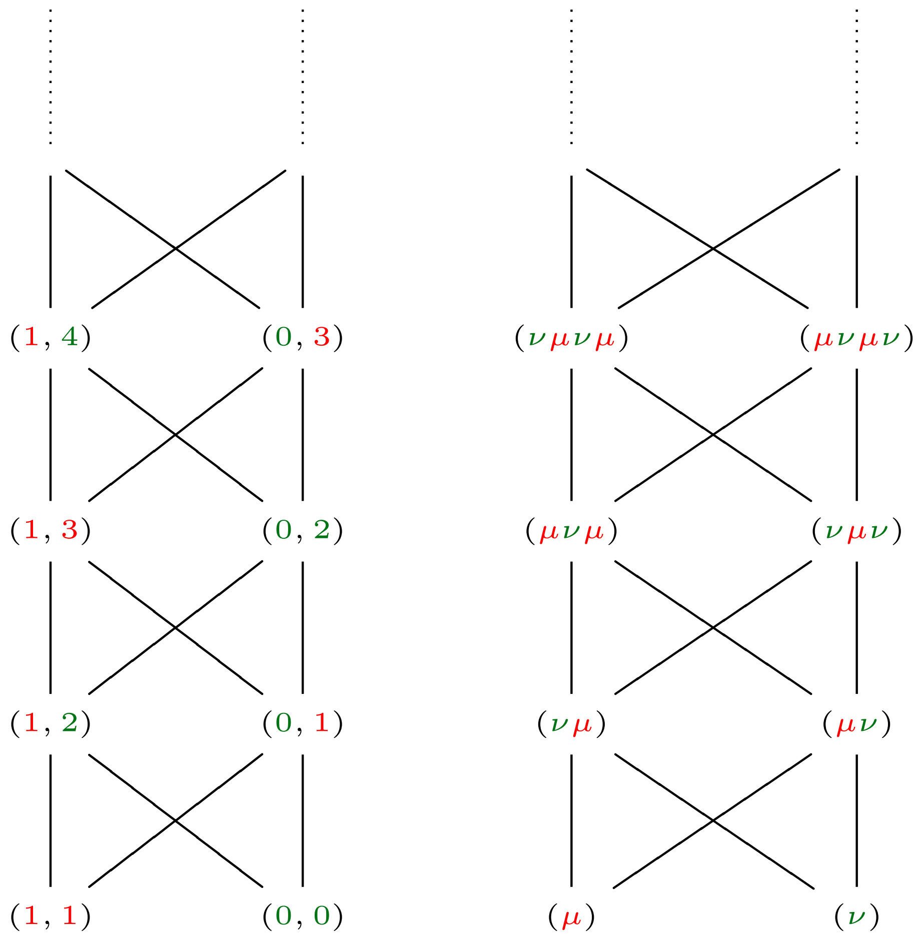
The Mostowski hierarchy — relation to the μ-calculus ctd.

$$T_n \quad = \quad \vartheta x_n \ldots \mu x_2 . \ldots \nu x_1 . \mu x_0 . \bigcup_i \underline{\mathbf{i}}(x_i, x_i)$$

$$W_n \quad = \quad \vartheta x_n \ldots \mu x_2 . \ldots \nu x_1 . \mu x_0 . \bigcup_i (d_i(x_i, tt) \cup d_i(tt, x_i) \cup c_i(x_i, x_i))$$

The index hierarchy of automata coincides with the μ-calculus hierarchy of nesting alternately the least (μ) and the greatest (ν) fixed points.

The two hierarchies



The two hierarchies in two versions

Non-deterministic hierarchy :

$$x \mid f(t_1, \ldots, t_k) \mid t_1 \vee t_2 \mid \mu x.t \mid \nu x.t \quad \equiv \text{ non-deterministic automata}$$

Alternating hierarchy :

$$x \mid f(t_1, \ldots, t_k) \mid t_1 \vee t_2 \mid t_1 \wedge t_2 \mid \mu x.t \mid \nu x.t \quad \equiv \text{ alternating automata}$$

We have

$$\bigcup \text{Non-deterministic hierarchy} = \bigcup \text{Alternating hierarchy}$$

but neither of the hierarchies refines the other :

- All T_n's are in the level $\mu\nu \equiv (0, 1)$ of the alternating hierarchy.
- T_n and W_n are on the same level in non-deterministic hierarchy, but not in the alternating hierarchy.

The Mostowski hierarchy — relation to complexity

The non-emptiness problem for non-deterministic parity tree automata is in NP $\cap$ co-NP (even UP $\cap$ co-UP).

It is polynomial–time equivalent to the *model–checking* problem for the μ-calculus.

Restricted to the automata $\mathcal{A}$ of index n, the problem can be solved in time $|\mathcal{A}|^{\mathcal{O}(n)}$.