Analog Recurrent Neural Networks on Infinite Input Streams

Jérémie Cabessa

UMass CS

8 November 2010

Outline

Introduction

Topology on the Cantor Space

ω-automata

ω-ARNNs

Conclusion

Outline

Introduction

Topology on the Cantor Space

ω-automata

ω-ARNNs

Conclusion

Outline

Introduction

Topology on the Cantor Space

ω-automata

ω-ARNNs

Conclusion

Outline

Introduction

Topology on the Cantor Space

ω-automata

ω-ARNNs

Conclusion

Outline

Introduction

Topology on the Cantor Space

ω-automata

ω-ARNNs

Conclusion

Introduction

Theorem (Siegelmann & Sontag)

- *Analog recurrent neural networks (ARNNs) can recognize in exponential time all possible languages.*
- *Analog recurrent neural networks recognize in polynomial time the class of languages $P/poly$.*

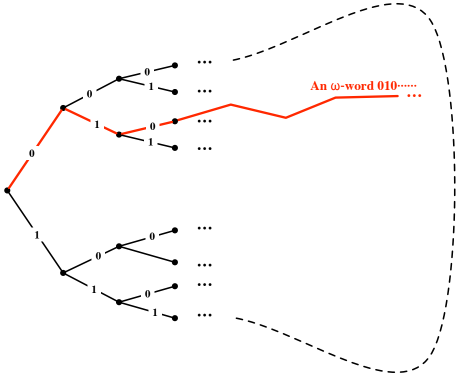
We prove the counterpart of this result in the context of infinite computation.

Definition

The *Cantor space* $\mathcal{C} = \{0,1\}^\omega$ is the set of all infinite sequences of bits.

For instance, $010110001010001010 0 \cdots \in \mathcal{C}$

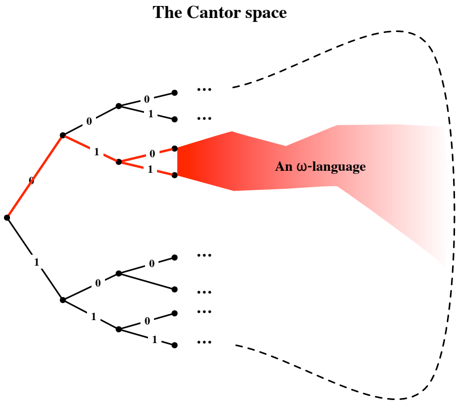
The Cantor space



Definition

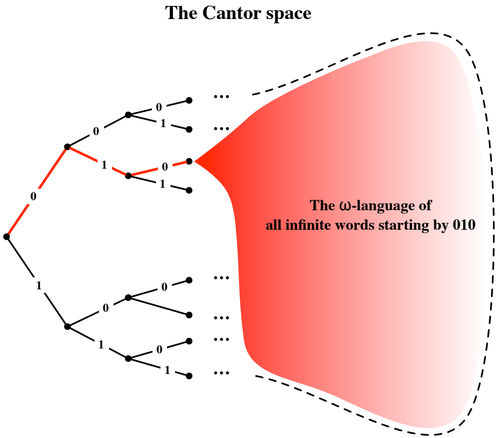
The *Cantor space* $\mathcal{C} = \{0,1\}^\omega$ is the set of all infinite sequences of bits.

For instance, $010110001010001010 0 \cdots \in \mathcal{C}$



We equip the Cantor space with the product topology of the discrete topology on $\{0, 1\}$.

The basic open ω-languages are of the form $L = \mathit{prefix} \cdot \{0, 1\}^\omega$



Definition

The levels of the *Borel hierarchy* are defined by induction on $\alpha < \omega_1$ as follows:

- $\mathbf{\Sigma}^0_1 = \{A \subseteq \mathcal{C} : A \text{ is open}\}$
- $\mathbf{\Sigma}^0_\alpha = \{\bigcup_{n \in \mathbb{N}} A_n : A_n \in \mathbf{\Pi}^0_\beta \text{ for } \beta < \alpha\}$
- $\mathbf{\Pi}^0_\alpha = \{A : A^\complement \in \mathbf{\Sigma}^0_\alpha\}$
- $\mathbf{\Delta}^0_\alpha = \{A : A \in \mathbf{\Sigma}^0_\alpha \cap \mathbf{\Pi}^0_\alpha\}$

Definition

The levels of the *Borel hierarchy* are defined by induction on $\alpha < \omega_1$ as follows:

- $\mathbf{\Sigma^0_1} = \{A \subseteq \mathcal{C} : A \text{ is open}\}$
- $\mathbf{\Sigma^0_\alpha} = \{\bigcup_{n\in\mathbb{N}} A_n : A_n \in \mathbf{\Pi^0_\beta} \text{ for } \beta < \alpha\}$
- $\mathbf{\Pi^0_\alpha} = \{A : A^{\complement} \in \mathbf{\Sigma^0_\alpha}\}$
- $\mathbf{\Delta^0_\alpha} = \{A : A \in \mathbf{\Sigma^0_\alpha} \cap \mathbf{\Pi^0_\alpha}\}$

Definition

The levels of the *Borel hierarchy* are defined by induction on $\alpha < \omega_1$ as follows:

- $\boldsymbol{\Sigma}^0_1 = \{A \subseteq \mathcal{C} : A$ is open$\}$
- $\boldsymbol{\Sigma}^0_{\boldsymbol{\alpha}} = \{\bigcup_{n \in \mathbb{N}} A_n : A_n \in \boldsymbol{\Pi}^0_{\boldsymbol{\beta}}$ for $\beta < \alpha\}$
- $\boldsymbol{\Pi}^0_{\boldsymbol{\alpha}} = \{A : A^{\complement} \in \boldsymbol{\Sigma}^0_{\boldsymbol{\alpha}}\}$
- $\boldsymbol{\Delta}^0_{\boldsymbol{\alpha}} = \{A : A \in \boldsymbol{\Sigma}^0_{\boldsymbol{\alpha}} \cap \boldsymbol{\Pi}^0_{\boldsymbol{\alpha}}\}$

Definition

The levels of the *Borel hierarchy* are defined by induction on $\alpha < \omega_1$ as follows:

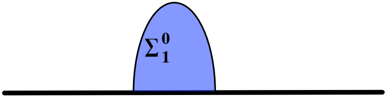
- $\mathbf{\Sigma}^0_1 = \{A \subseteq \mathcal{C} : A \text{ is open}\}$
- $\mathbf{\Sigma}^0_{\boldsymbol{\alpha}} = \{\bigcup_{n \in \mathbb{N}} A_n : A_n \in \mathbf{\Pi}^0_{\boldsymbol{\beta}} \text{ for } \boldsymbol{\beta} < \boldsymbol{\alpha}\}$
- $\mathbf{\Pi}^0_{\boldsymbol{\alpha}} = \{A : A^{\complement} \in \mathbf{\Sigma}^0_{\boldsymbol{\alpha}}\}$
- $\mathbf{\Delta}^0_{\boldsymbol{\alpha}} = \{A : A \in \mathbf{\Sigma}^0_{\boldsymbol{\alpha}} \cap \mathbf{\Pi}^0_{\boldsymbol{\alpha}}\}$

Definition

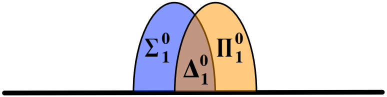
The levels of the *Borel hierarchy* are defined by induction on $\alpha < \omega_1$ as follows:

- $\boldsymbol{\Sigma}^0_1 = \{A \subseteq \mathcal{C} : A \text{ is open}\}$
- $\boldsymbol{\Sigma}^0_{\boldsymbol{\alpha}} = \{\bigcup_{n \in \mathbb{N}} A_n : A_n \in \boldsymbol{\Pi}^0_{\boldsymbol{\beta}} \text{ for } \boldsymbol{\beta} < \boldsymbol{\alpha}\}$
- $\boldsymbol{\Pi}^0_{\boldsymbol{\alpha}} = \{A : A^{\complement} \in \boldsymbol{\Sigma}^0_{\boldsymbol{\alpha}}\}$
- $\boldsymbol{\Delta}^0_{\boldsymbol{\alpha}} = \{A : A \in \boldsymbol{\Sigma}^0_{\boldsymbol{\alpha}} \cap \boldsymbol{\Pi}^0_{\boldsymbol{\alpha}}\}$

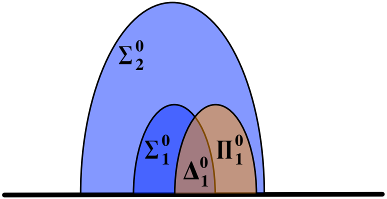
The Borel Hierarchy



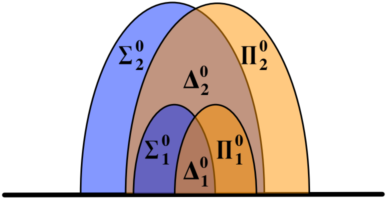
The Borel Hierarchy



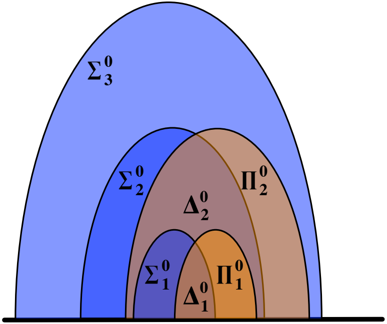
The Borel Hierarchy



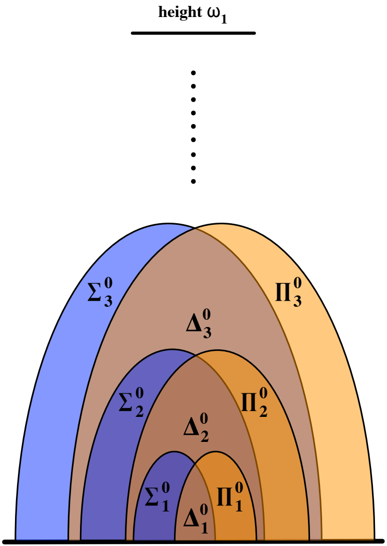
The Borel Hierarchy



The Borel Hierarchy



The Borel Hierarchy

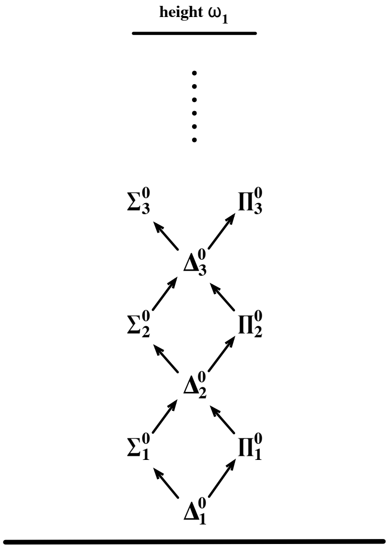


The Borel Hierarchy

height ω_1

$$\begin{array}{ccc}
 & \vdots & \\
\boldsymbol{\Sigma}^0_3 & & \boldsymbol{\Pi}^0_3 \\
 & \subsetneq \quad \subsetneq & \\
 & \boldsymbol{\Delta}^0_3 & \\
 & \subsetneq \quad \subsetneq & \\
\boldsymbol{\Sigma}^0_2 & & \boldsymbol{\Pi}^0_2 \\
 & \subsetneq \quad \subsetneq & \\
 & \boldsymbol{\Delta}^0_2 & \\
 & \subsetneq \quad \subsetneq & \\
\boldsymbol{\Sigma}^0_1 & & \boldsymbol{\Pi}^0_1 \\
 & \subsetneq \quad \subsetneq & \\
 & \boldsymbol{\Delta}^0_1 &
\end{array}$$

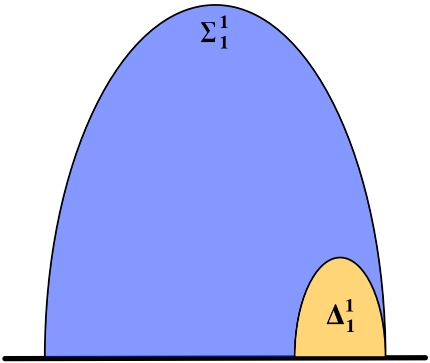
The Borel Hierarchy



Definition

An ω-language is called *analytic* ($\mathbf{\Sigma}^1_1$) if it is the 1st projection of some $\mathbf{\Pi}^0_2$-set of $\mathcal{C} \times \mathcal{C}$.

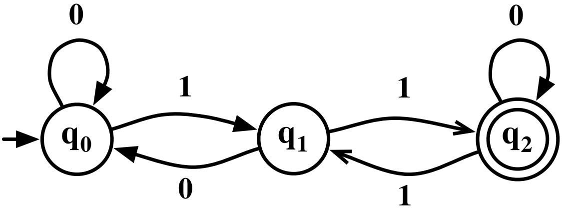
The class of analytic sets strictly contains the class of Borel sets $\mathbf{\Sigma}^1_1 \supsetneq \mathbf{\Delta}^1_1$



Computation over infinite objects

- Purely theoretical interest
- Connections with formal language theory, game theory, algebra, and logics
- Practical implications: modeling infinite behavior, verification of non-terminating reactive systems, ...

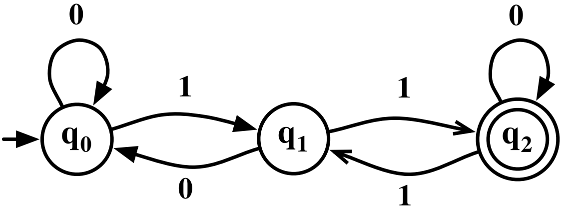
Classical finite automaton



The set of words accepted by the automaton is the called the language recognized by this automaton.

Many other kinds of abstract machines: counter automata, pushdown automata, Turing machines, etc.

Classical finite automaton



The set of words accepted by the automaton is the called the language recognized by this automaton.

Many other kinds of abstract machines: counter automata, pushdown automata, Turing machines, etc.

ω-automaton (Muller)

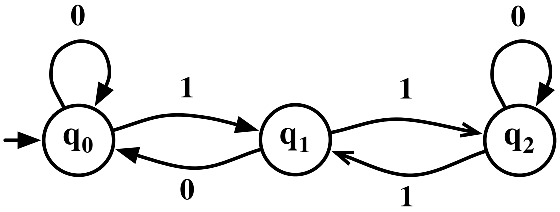


table $\mathcal{T} = \{\{q_1, q_2\}, \{q_2\}\}$

ω-automaton (Muller)

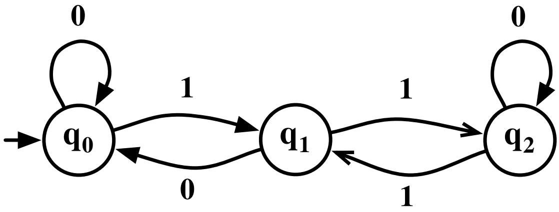


table $\mathcal{T} = \{\{q_1, q_2\}, \{q_2\}\}$

input 001111111111 . . . . . . accepted

input 010101010101 . . . . . . rejected

input 110000000000 . . . . . . accepted

ω-automaton (Muller)

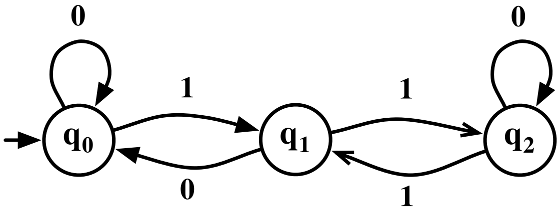


table $\mathcal{T} = \{\{q_1, q_2\}, \{q_2\}\}$

input 001111111111 . . . . . . **accepted**

input 010101010101 . . . . . . **rejected**

input 110000000000 . . . . . . **accepted**

ω-automaton (Muller)

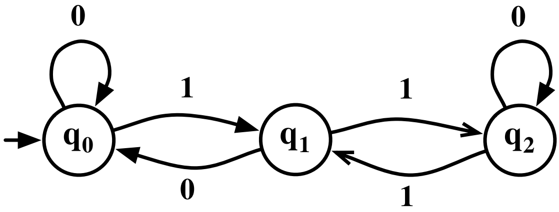


table $\mathcal{T} = \{\{q_1, q_2\}, \{q_2\}\}$

input 001111111111 . . . . . . **accepted**

input 010101010101 . . . . . . **rejected**

input 110000000000 . . . . . . **accepted**

ω-automaton (Muller)

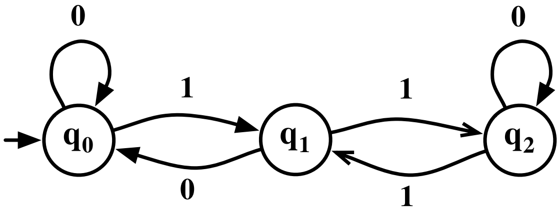


table $\mathcal{T} = \{\{q_1, q_2\}, \{q_2\}\}$

The set of ω-words accepted by the automaton is the called the ω-language recognized by this automaton.

ω-automaton (Muller)

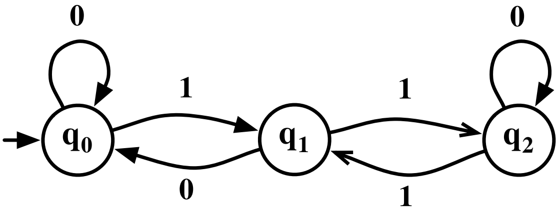
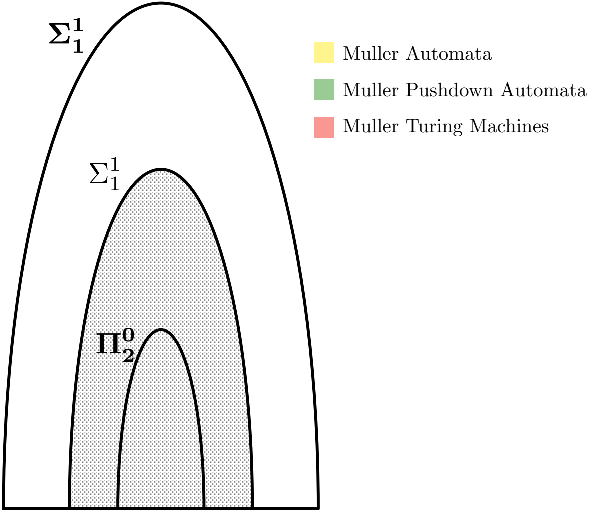


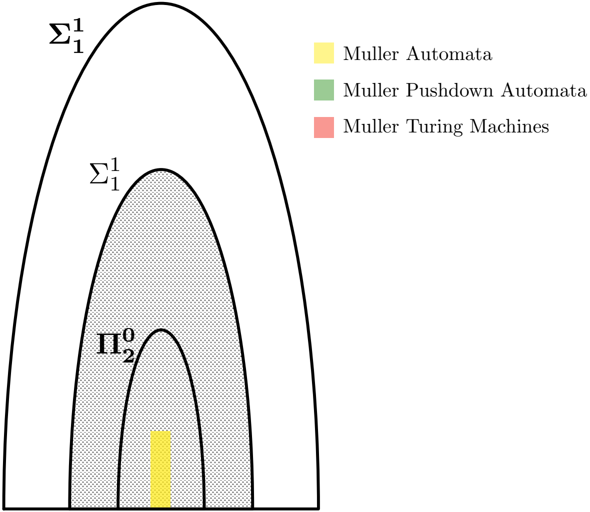
table $\mathcal{T} = \{\{q_1, q_2\}, \{q_2\}\}$

Many other kinds of abstract ω-machines: Muller counter automata, Muller pushdown automata, Muller Turing machines, etc.

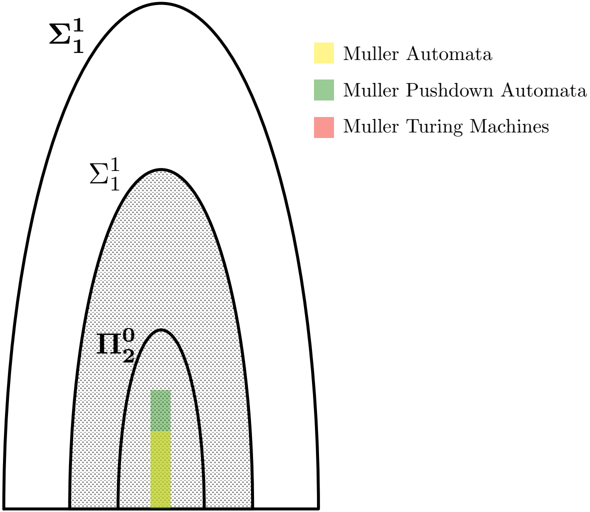
Deterministic case



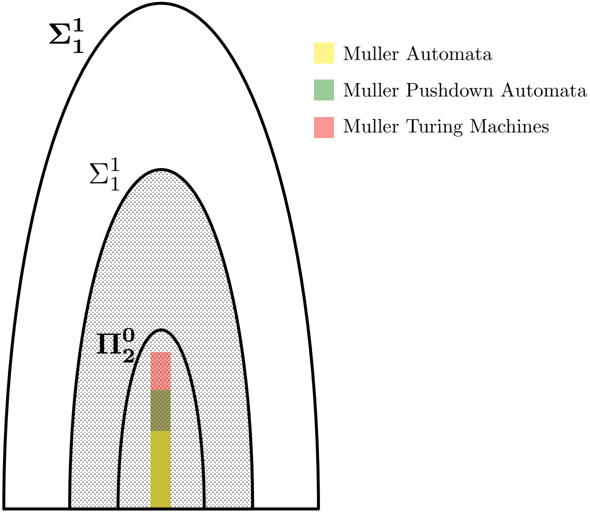
Deterministic case



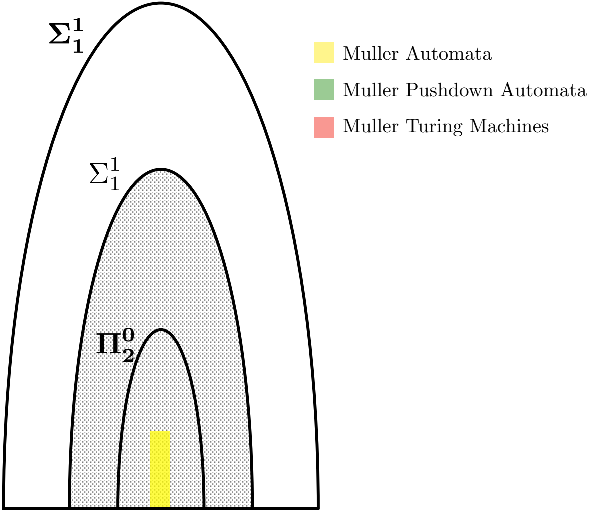
Deterministic case



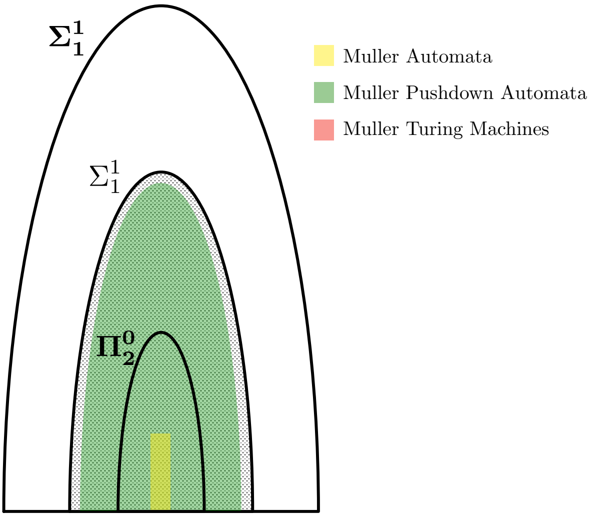
Deterministic case



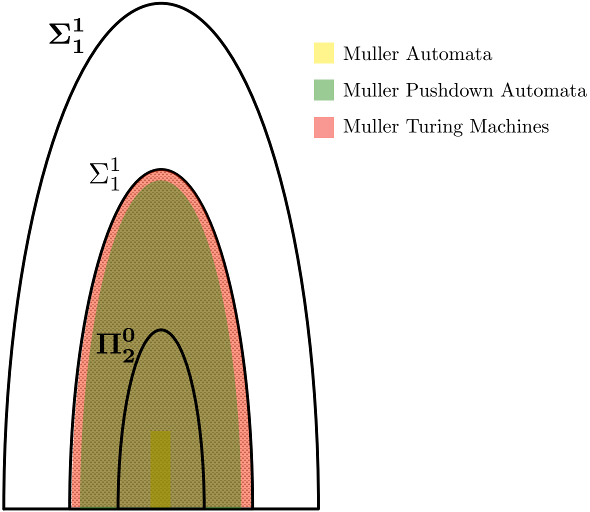
Non-deterministic case



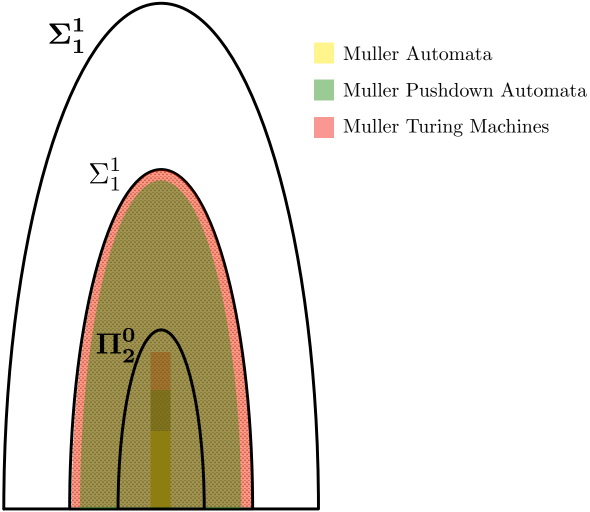
Non-deterministic case



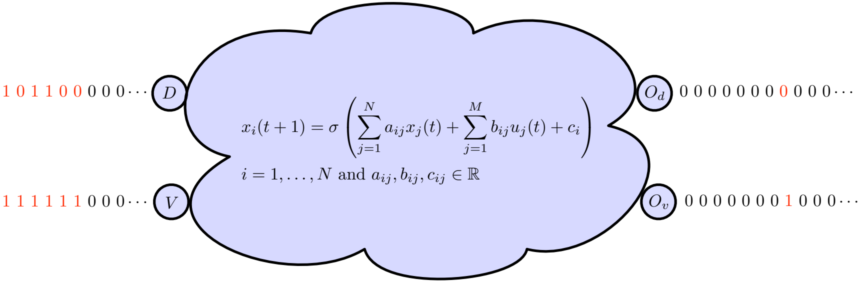
Non-deterministic case



Deterministic and non-deterministic cases

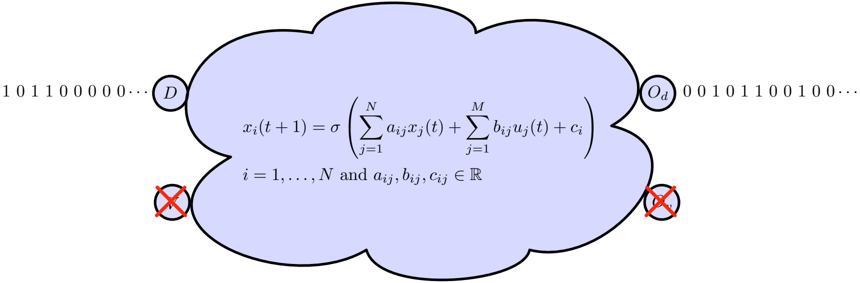


Classical ARNNs



The set of finite words accepted by $\mathcal{N}$ is called the language recognized by $\mathcal{N}$.

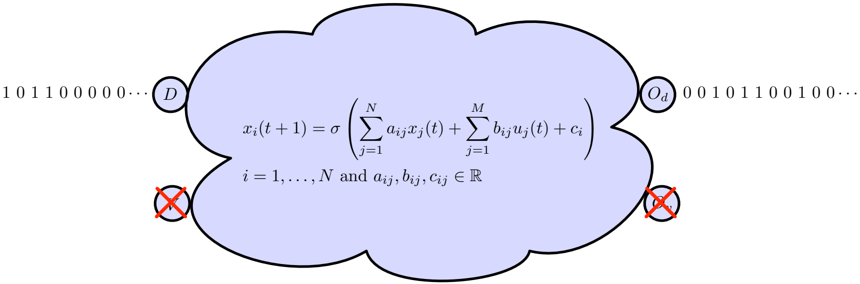
ω-ARNNs



An ω-word is said to be accepted by $\mathcal{N}$ iff it induces infinitely many output answers (spikes) along the infinite computation.

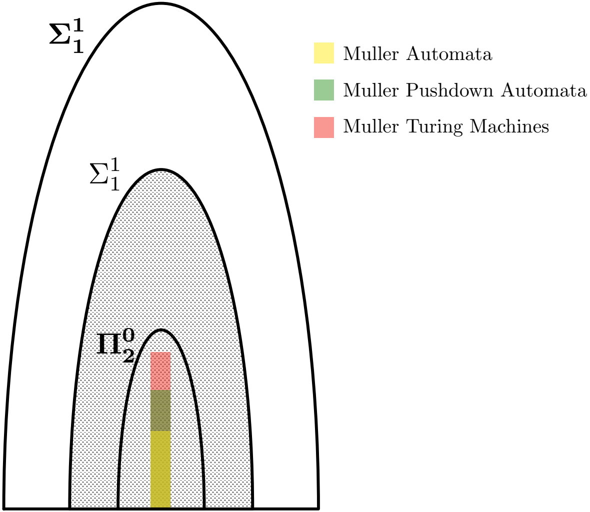
The set of finite ω-words accepted by $\mathcal{N}$ is called the language recognized by $\mathcal{N}$

ω-ARNNs

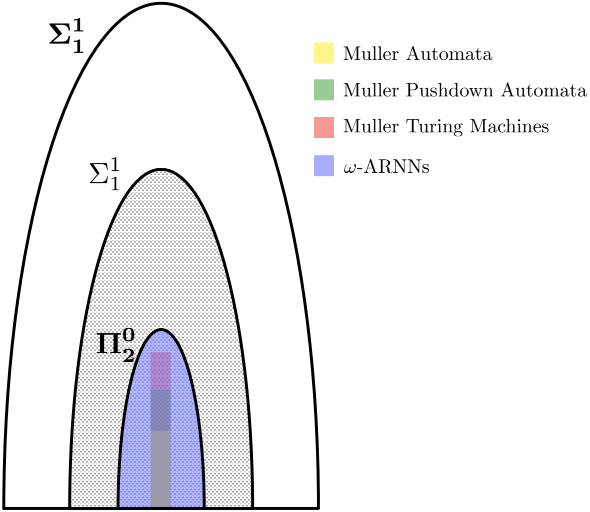


An ω-word is said to be accepted by $\mathcal{N}$ iff it induces infinitely many output answers (spikes) along the infinite computation. The set of finite ω-words accepted by $\mathcal{N}$ is called the language recognized by $\mathcal{N}$

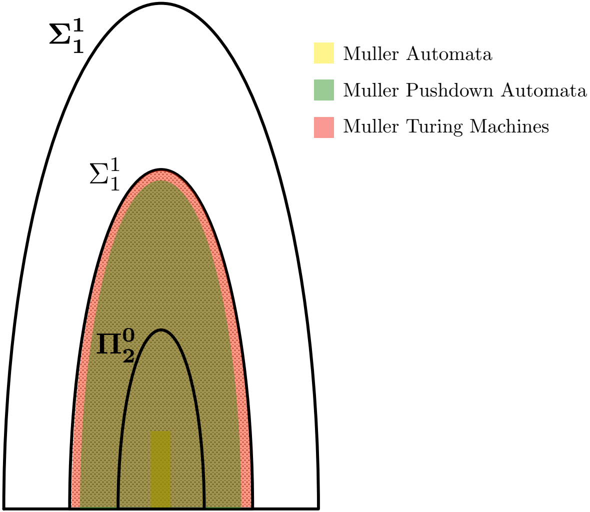
Deterministic case



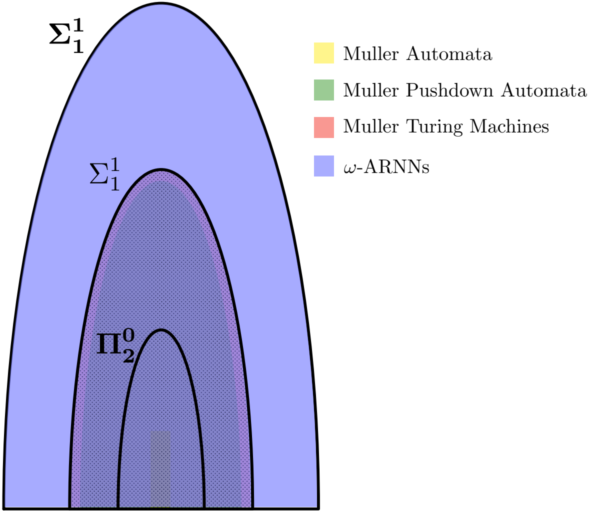
Deterministic case



Non-deterministic case



Non-deterministic case



Theorem

Let $L \subseteq \{0,1\}^\omega$ be an ω-language. The following conditions are equivalent:

1. *L is recognizable by some deterministic ω-ARNN*
2. $L \in \mathbf{\Pi}^0_2$

Theorem

Let $L \subseteq \{0,1\}^\omega$ be an ω-language. The following conditions are equivalent:

1. *L is recognizable by some non-deterministic ω-ARNN*
2. $L \in \mathbf{\Sigma}^1_1$

Theorem

Let $L \subseteq \{0,1\}^\omega$ be an ω-language. The following conditions are equivalent:

1. *L is recognizable by some deterministic ω-ARNN*
2. $L \in \mathbf{\Pi}^0_2$

Theorem

Let $L \subseteq \{0,1\}^\omega$ be an ω-language. The following conditions are equivalent:

1. *L is recognizable by some non-deterministic ω-ARNN*
2. $L \in \mathbf{\Sigma}^1_1$

Conclusion

- Deterministic ω-ARNNs are strictly more expressive than any other usual deterministic abstract machines, in particular det Muller Turing machines.
- Non-deterministic ω-ARNNs are strictly more expressive than any other usual non-deterministic abstract machines, in particular non-det Muller Turing machines.