A Classification of Neural Networks

Jérémie Cabessa

University Joseph Fourier – Grenoble 1

10 May 2009

Outline

Outline

Outline

Outline

Introduction

In 1967, Minsky proved the equivalence between McCulloch and Pitts's neural nets and finite state machines.

We extend this result for infinite word reading machines, and provide a refined classification of simple neural nets.

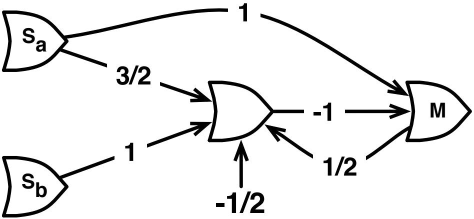
Introduction

In 1967, Minsky proved the equivalence between McCulloch and Pitts's neural nets and finite state machines.

We extend this result for infinite word reading machines, and provide a refined classification of simple neural nets.

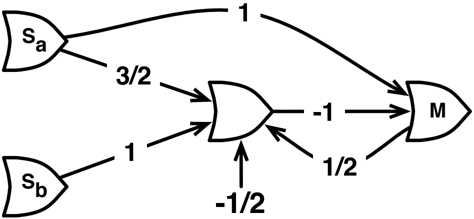
Neural nets and automata

McCulloch & Pitts' neural net



Neural nets and automata

McCulloch & Pitts' neural net

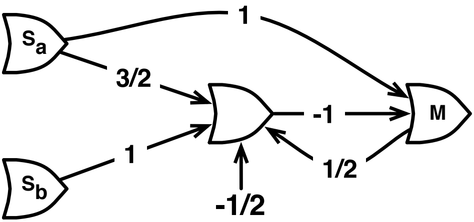


stimulus *abbabbabbabb* . . . . . . accepted

stimulus *bbbbbbbbbbbb* . . . . . . rejected

Neural nets and automata

McCulloch & Pitts' neural net

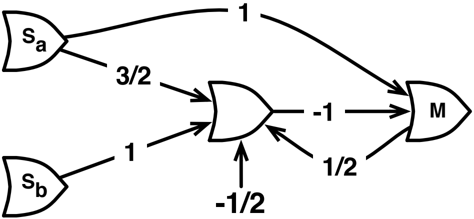


stimulus *abbabbabbabb . . . . . .* **accepted**

stimulus *bbbbbbbbbbbb . . . . . .* **rejected**

Neural nets and automata

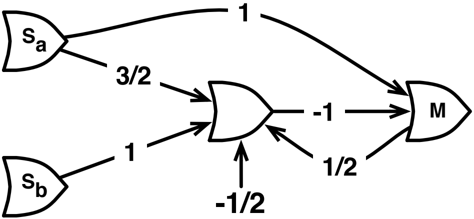
McCulloch & Pitts' neural net



stimulus	$abbabbabbabb\ldots\ldots$	**accepted**
stimulus	$bbbbbbbbbbbb\ldots\ldots$	**rejected**

Neural nets and automata

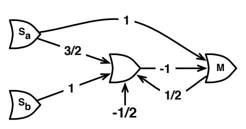
McCulloch & Pitts' neural net



The set of stimuli accepted by this net is the *language recognized by this net.*

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.

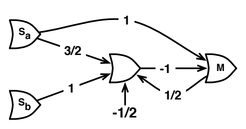
The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right), \text{ where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.

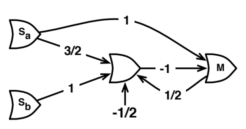
The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right) \text{, where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1 \text{ ,} \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.

The dynamic of cell i is given by

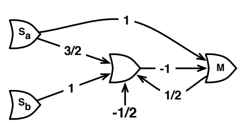
$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right), \text{ where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where

- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,



- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.

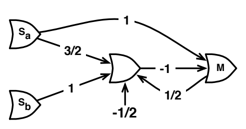
The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right), \text{ where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.

The dynamic of cell i is given by

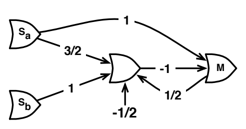
$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right), \text{ where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where

- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.



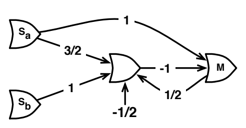
The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right), \text{ where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \to \mathbb{Q}$ gives the weights of the connections,
- $b : C \to \mathbb{Q}$ gives the background activity.

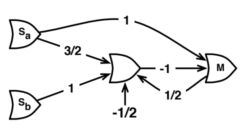
The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right), \text{ where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \longrightarrow \mathbb{Q}$ gives the weights of the connections,
- $b : C \longrightarrow \mathbb{Q}$ gives the background activity.

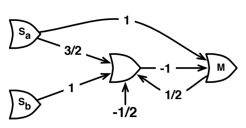
The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right) \text{ , where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1 \text{ ,} \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Definition

A *MP neural net* is a tuple $\mathcal{N} = (C, S, M, w, b)$, where



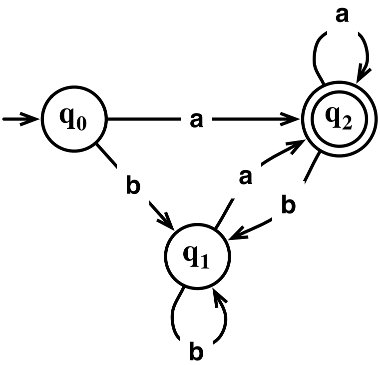
- C is a finite set of cells,
- $S \subseteq C$ is a set of sensory cells,
- $M \subseteq C$ is a set of motor cells,
- $w : C \times C \longrightarrow \mathbb{Q}$ gives the weights of the connections,
- $b : C \longrightarrow \mathbb{Q}$ gives the background activity.

The dynamic of cell i is given by

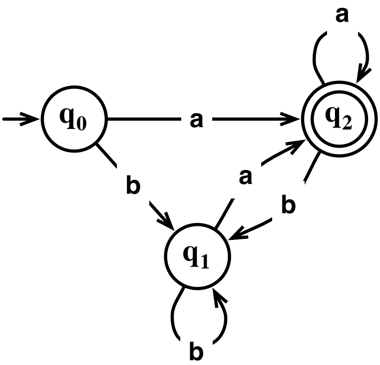
$$x_i(t+1) = \sigma\left(\sum_{j \in C} w(j,i) \cdot x_j(t) + b(i)\right) \text{ , where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1\ , \\ 1 & \text{if } x \geq 1. \end{cases}$$

- An infinite stimulus w is *accepted* by $\mathcal{N}$ iff it evokes infinitely many motor responses.
- The set of infinite stimuli accepted by $\mathcal{N}$ is called the *ω-language recognized by $\mathcal{N}$*.

Büchi automaton



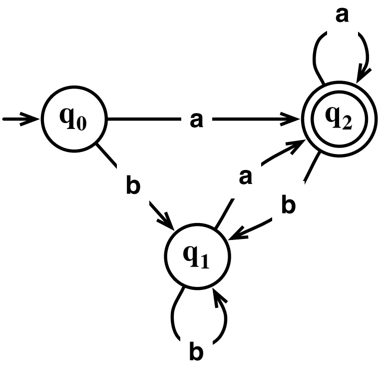
Büchi automaton



input *abababab* . . . . . . accepted

input *baabbbbb* . . . . . . rejected

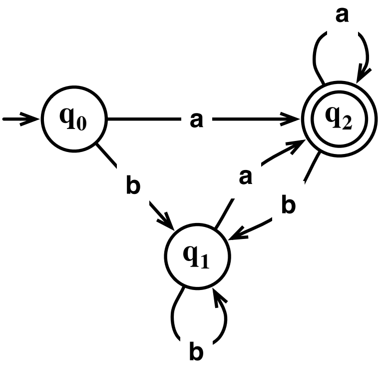
Büchi automaton



input ababab ab . . . . . . **accepted**

input baabbbbb . . . . . . **rejected**

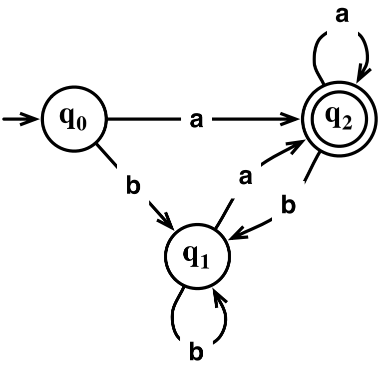
Büchi automaton



input $ababab ab\ldots\ldots$ **accepted**

input $baabbbbb\ldots\ldots$ **rejected**

Büchi automaton

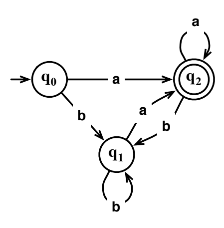


The set of words accepted by this automaton is the *language recognized by this automaton.*

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

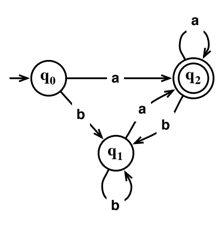


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $Inf(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

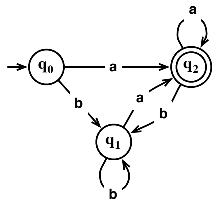


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $Inf(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

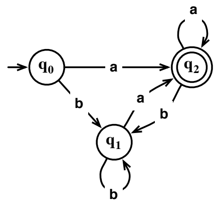


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $Inf(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

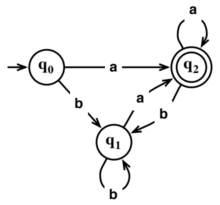


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $\mathit{Inf}(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

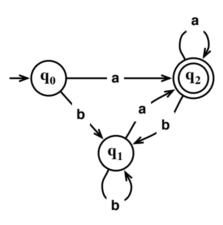


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $Inf(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

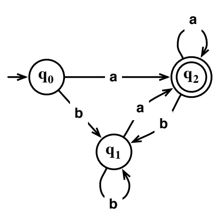


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $\mathit{Inf}(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.

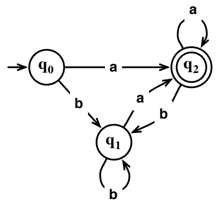


- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $Inf(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Definition

A *det. Büchi automaton* is a tuple $\mathcal{A} = (Q, A, \delta, i, F)$, where

- Q is a finite set of states,
- A is an alphabet,
- $\delta : Q \times A \longrightarrow Q$ is the transition function,
- $i \in Q$ is the initial state,
- $F \subseteq Q$ is the set of finite states.



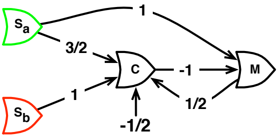
- An infinite input w over A is *accepted* by $\mathcal{A}$ iff $\mathit{Inf}(w) \cap F \neq \emptyset$.
- The set of infinite words accepted by $\mathcal{A}$ is called the *ω-language recognized by $\mathcal{A}$*.

Theorem

Let L be an ω-language. Then the following are equivalent:

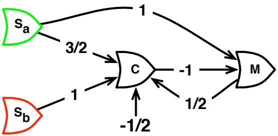
1. *L is recognizable by some MP neural net;*
2. *L is recognizable by some det. Büchi automata;*

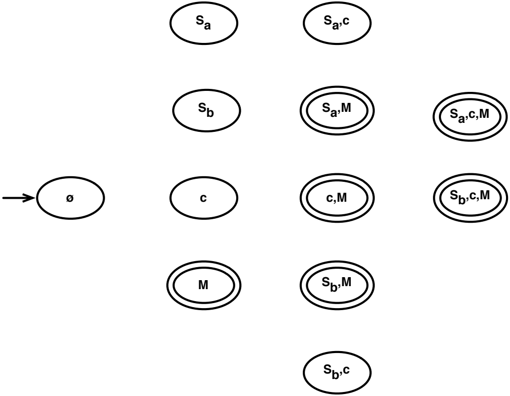
Proof: $(1) \Rightarrow (2)$: From MP neural nets to det. Büchi automata…



By construction, $L(\mathcal{N}) = L(\mathcal{A})$.

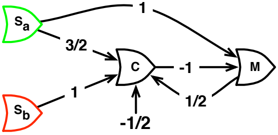
Proof: $(1) \Rightarrow (2)$: From MP neural nets to det. Büchi automata...

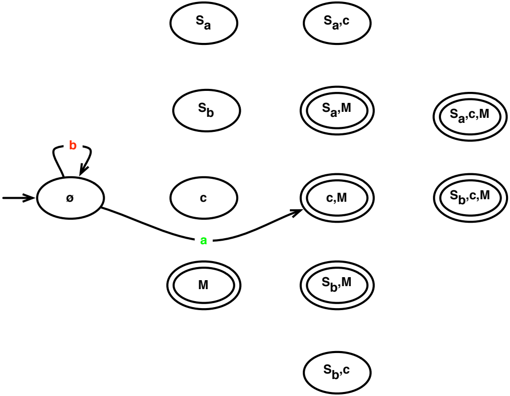




By construction, $L(\mathcal{N}) = L(\mathcal{A})$.

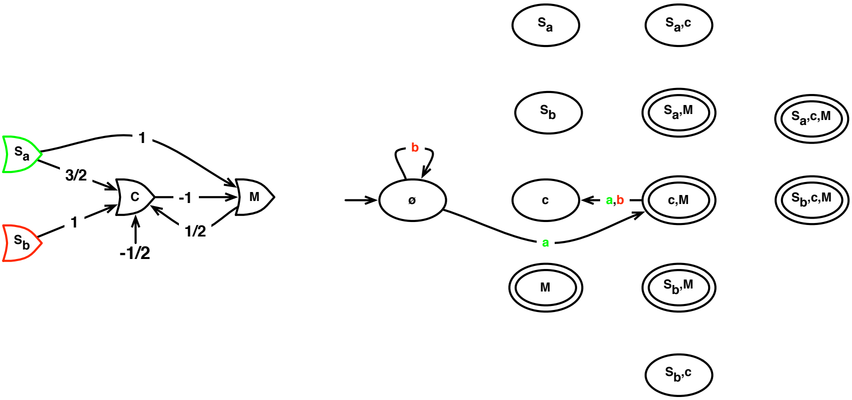
Proof: $(1) \Rightarrow (2)$: From MP neural nets to det. Büchi automata…





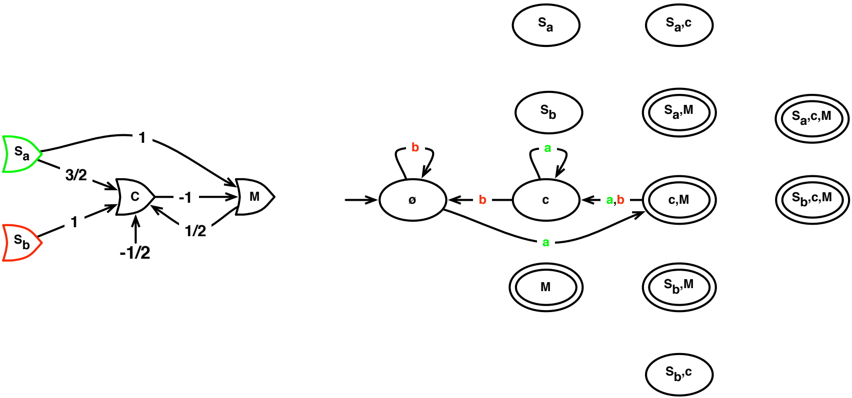
By construction, $L(\mathcal{N}) = L(\mathcal{A})$.

Proof: $(1) \Rightarrow (2)$: From MP neural nets to det. Büchi automata...



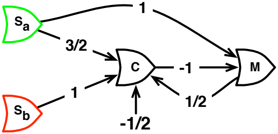
By construction, $L(\mathcal{N}) = L(\mathcal{A})$.

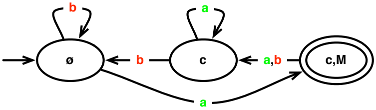
Proof: $(1) \Rightarrow (2)$: From MP neural nets to det. Büchi automata...



By construction, $L(\mathcal{N}) = L(\mathcal{A})$.

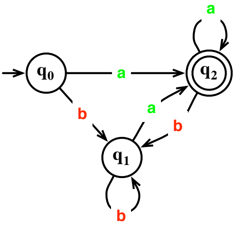
Proof: $(1) \Rightarrow (2)$: From MP neural nets to det. Büchi automata…





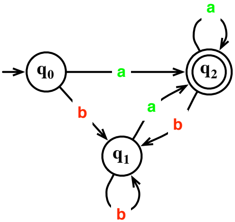
By construction, $L(\mathcal{N}) = L(\mathcal{A})$.

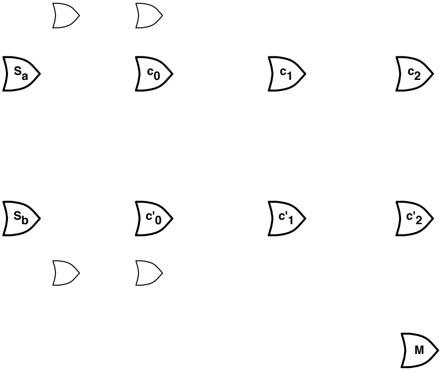
$(2) \Rightarrow (1)$: From det. Büchi automata to MP neural nets...



By construction, $L(\mathcal{A}) = L(\mathcal{N})$. $\square$

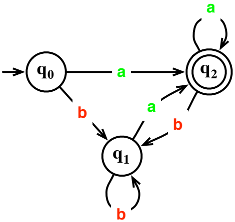
$(2) \Rightarrow (1)$: From det. Büchi automata to MP neural nets...

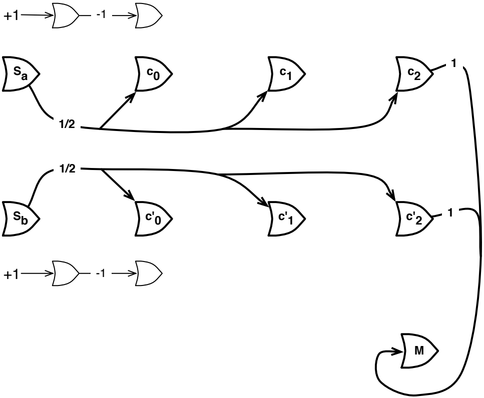




By construction, $L(\mathcal{A}) = L(\mathcal{N})$.

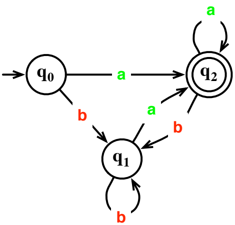
$(2) \Rightarrow (1)$: From det. Büchi automata to MP neural nets...

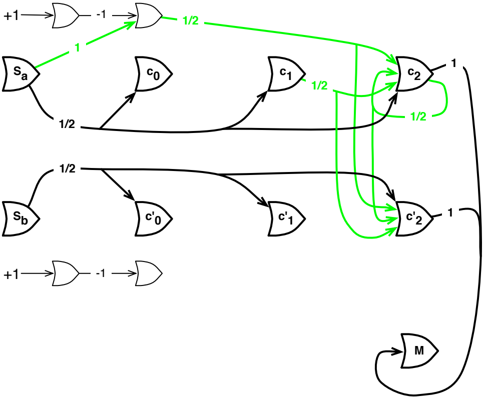




By construction, $L(\mathcal{A}) = L(\mathcal{N})$.

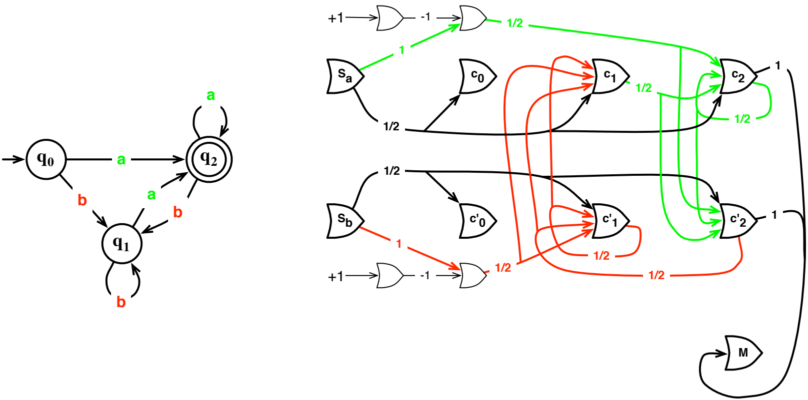
$(2) \Rightarrow (1)$: From det. Büchi automata to MP neural nets...





By construction, $L(\mathcal{A}) = L(\mathcal{N})$.

$(2) \Rightarrow (1)$: From det. Büchi automata to MP neural nets...



By construction, $L(\mathcal{A}) = L(\mathcal{N})$.

A classification of neural nets

Let A and B be two ω-languages:

$$A \leq_W B \quad \text{iff} \quad \text{there exists } f \text{ continuous s.t. } A = f^{-1}(B)$$
$$\text{iff} \quad \text{there exists } f \text{ continuous s.t. } x \in A \Leftrightarrow f(x) \in B$$

Then as usual

$$A <_W B \quad \text{iff} \quad A \leq_W B \text{ and } B \not\leq_W A$$
$$A \equiv_W B \quad \text{iff} \quad A \leq_W B \text{ and } B \leq_W A$$

A classification of neural nets

Let A and B be two ω-languages:

$$
\begin{aligned}
A \leq_W B \quad & \text{iff} \quad \text{there exists } f \text{ continuous s.t. } A = f^{-1}(B) \\
& \text{iff} \quad \text{there exists } f \text{ continuous s.t. } x \in A \Leftrightarrow f(x) \in B
\end{aligned}
$$

Then as usual

$$
\begin{aligned}
A <_W B \quad & \text{iff} \quad A \leq_W B \text{ and } B \not\leq_W A \\
A \equiv_W B \quad & \text{iff} \quad A \leq_W B \text{ and } B \leq_W A
\end{aligned}
$$

A classification of neural nets

Let A and B be two ω-languages:

$$\begin{aligned} A \leq_W B \quad & \text{iff} \quad \text{there exists } f \text{ continuous s.t. } A = f^{-1}(B) \\ & \text{iff} \quad \text{there exists } f \text{ continuous s.t. } x \in A \Leftrightarrow f(x) \in B \end{aligned}$$

Then as usual

$$\begin{aligned} A <_W B \quad & \text{iff} \quad A \leq_W B \text{ and } B \not\leq_W A \\ A \equiv_W B \quad & \text{iff} \quad A \leq_W B \text{ and } B \leq_W A \end{aligned}$$

$A \leq_W B$ means intuitively than A is “less complicated” than B.
Indeed we have …

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \ \text{ for some } f \text{ continuous}$$

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is "less complicated" than B. Indeed we have ...

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \text{ for some } f \text{ continuous}$$

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is "less complicated" than B. Indeed we have ...

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \text{ for some } f \text{ continuous}$$

Assume that for any element x, you know whether $x \in B$ or not

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is "less complicated" than B. Indeed we have . . .

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \text{ for some } f \text{ continuous}$$

Does a given $a \in A$?

Assume that for any element x, you know whether $x \in B$ or not

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is "less complicated" than B. Indeed we have ...

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \text{ for some } f \text{ continuous}$$

Does a given $a \in A$?

Assume that for any element x, you know whether $x \in B$ or not

$$a \longmapsto f \longrightarrow f(a)$$

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is "less complicated" than B. Indeed we have ...

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \text{ for some } f \text{ continuous}$$

Does a given $a \in A$?

Assume that for any element x, you know whether $x \in B$ or not

$$a \longmapsto f \longrightarrow f(a)$$

If $f(a) \in B$, then $a \in A$

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is "less complicated" than B. Indeed we have ...

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \ \text{ for some } f \text{ continuous}$$

Does a given $a \in A$?

Assume that for any element x, you know whether $x \in B$ or not

$$a \longmapsto f \longrightarrow f(a)$$

If $f(a) \in B$, then $a \in A$

If $f(a) \notin B$, then $a \notin A$

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

$A \leq_W B$ means intuitively than A is “less complicated” than B. Indeed we have ...

$$A \leq_W B \quad \text{iff}_{def} \quad x \in A \Leftrightarrow f(x) \in B \text{ for some } f \text{ continuous}$$

Does a given $a \in A$?

Assume that for any element x, you know whether $x \in B$ or not

$$a \longmapsto \xrightarrow{f} f(a)$$

If $f(a) \in B$, then $a \in A$

If $f(a) \notin B$, then $a \notin A$

In other terms, the belonging problem in A reduces via f to the belonging problem in B.

Definition

- The collection of ω-languages recognized by Det-Büchi automata ordered by $\leq_W$ is called the *Det. Büchi hierarchy*.
- The collection of ω-languages recognized by MP-neural nets ordered by $\leq_W$ is called the *MP-neural hierarchy*.

By the previous equivalence theorem, the Det-Büchi and the MP-neural hierarchies are equal.

Definition

- The collection of ω-languages recognized by Det-Büchi automata ordered by $\leq_W$ is called the *Det. Büchi hierarchy*.
- The collection of ω-languages recognized by MP-neural nets ordered by $\leq_W$ is called the *MP-neural hierarchy*.

By the previous equivalence theorem, the Det-Büchi and the MP-neural hierarchies are equal.

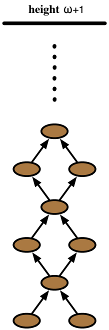
Definition

- The collection of ω-languages recognized by Det-Büchi automata ordered by $\leq_W$ is called the *Det. Büchi hierarchy*.
- The collection of ω-languages recognized by MP-neural nets ordered by $\leq_W$ is called the *MP-neural hierarchy*.

By the previous equivalence theorem, the Det-Büchi and the MP-neural hierarchies are equal.

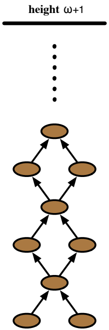
Theorem

- *The Det-Büchi hierarchy is well-founded, has width 2 and height $\omega + 1$.*
- *The MP-neural hierarchy is well-founded, has width 2 and height $\omega + 1$.*



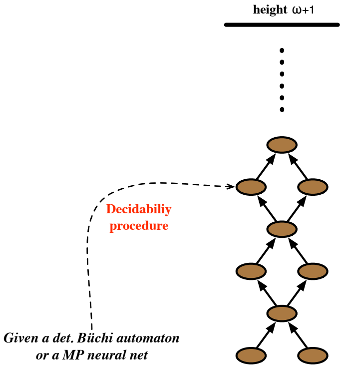
Theorem

- *The Det-Büchi hierarchy is well-founded, has width* 2 *and height* $\omega + 1$.
- *The MP-neural hierarchy is well-founded, has width* 2 *and height* $\omega + 1$.



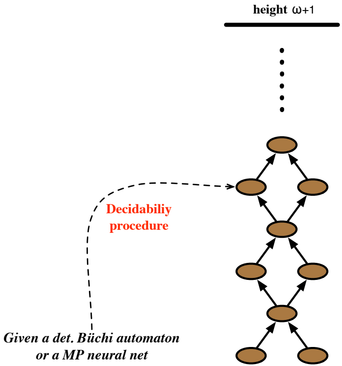
Theorem

- *The Det-Büchi hierarchy is decidable.*
- *The MP-Neural hierarchy is decidable.*



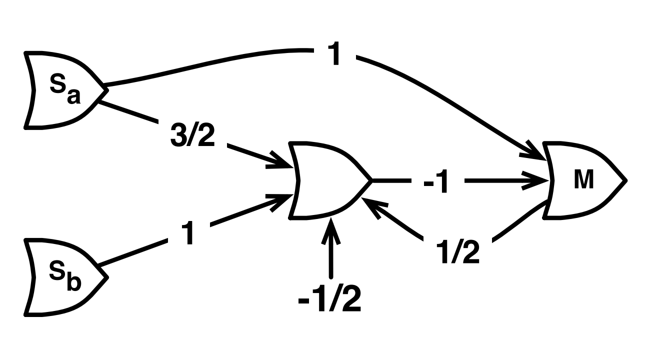
Theorem

- *The Det-Büchi hierarchy is decidable.*
- *The MP-Neural hierarchy is decidable.*



Example

Consider the following MP neural net $\mathcal{N}$.



Then the degree of $L(\mathcal{N})$ in the MP-neural hierarchy is ω.

Significant extension of this work:

we are also able to establish another equivalence between some kind of McCulloch and Pitts' neural nets and Muller automata; this induces another decidable hierarchy of networks of height ω^ω.

Significant extension of this work:

we are also able to establish another equivalence between some kind of McCulloch and Pitts' neural nets and Muller automata; this induces another decidable hierarchy of networks of height ω^ω.

Significant extension of this work:

we are also able to establish another equivalence between some kind of McCulloch and Pitts' neural nets and Muller automata; this induces another decidable hierarchy of networks of height ω^ω.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in this hierarchy is the new index of complexity that we propose.
- Future work: investigate the computational capabilities of more biologically plausible neural nets.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in this hierarchy is the new index of complexity that we propose.
- Future work: investigate the computational capabilities of more biologically plausible neural nets.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in this hierarchy is the new index of complexity that we propose.
- Future work: investigate the computational capabilities of more biologically plausible neural nets.