Parsing and Syntactic Analysis Seminar, Charles University, Prague, November 30, 2023

Chomsky-Like Neural Network Hierarchy

Jiří Šíma

www.cs.cas.cz/~sima

Institute of Computer Science
The Czech Academy of Sciences

joint work with

Petr Savický (ICS, Czech Academy of Sciences)

Petr Jančar (FSC, Palacký University Olomouc)

Martin Plátek (FMP, Charles University)

References

This lecture surveys selected computability results of the project

FoNeCo: Analytical Foundations of Neurocomputing

(Czech Science Foundation, GA22-02067S, 2019-2021),

published in the following papers (two won the Best ICS Paper Award):

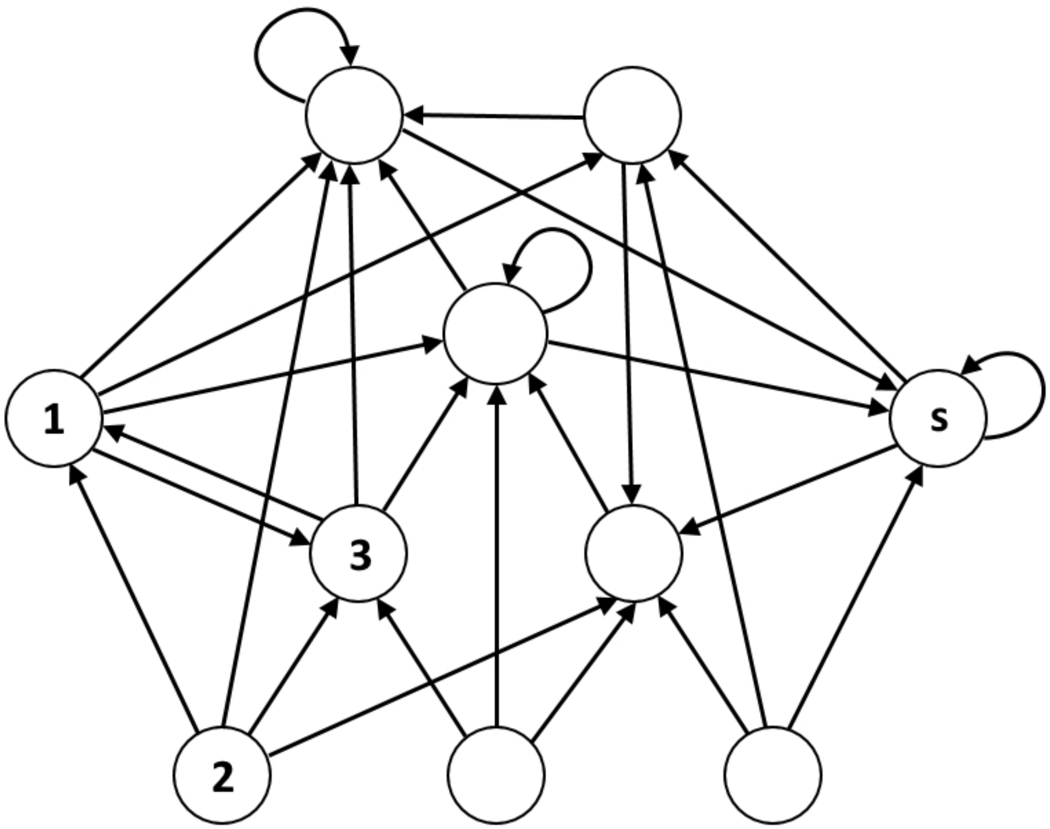
- J. Šíma: Subrecursive neural networks. *Neural Networks* 116:208-223, 2019.
- J. Šíma: Analog neuron hierarchy. *Neural Networks* 128:199-218, 2020.
- J. Šíma, P. Savický: Quasi-periodic β-expansions and cut languages. *Theoretical Computer Science* 720:1-23, 2018.
- P. Jančar, J. Šíma: The simplest non-regular deterministic context-free language. *Proceedings of the MFCS 2021*, LIPIcs 202, pp. 63:1-63:18, Dagstuhl, 2021.
- J. Šíma: Stronger separation of analog neuron hierarchy by deterministic context-free languages. *Neurocomputing* 493:605-612, 2022.

Outline of Talk

1. The Neural Network Model
2. The Computational Power of Neural Networks
3. A Chomsky-Like Neural Network Hierarchy
4. Periodic Numbers in Positional Systems with Non-Integer Base
5. $\mathcal{C}$-Simple Problems

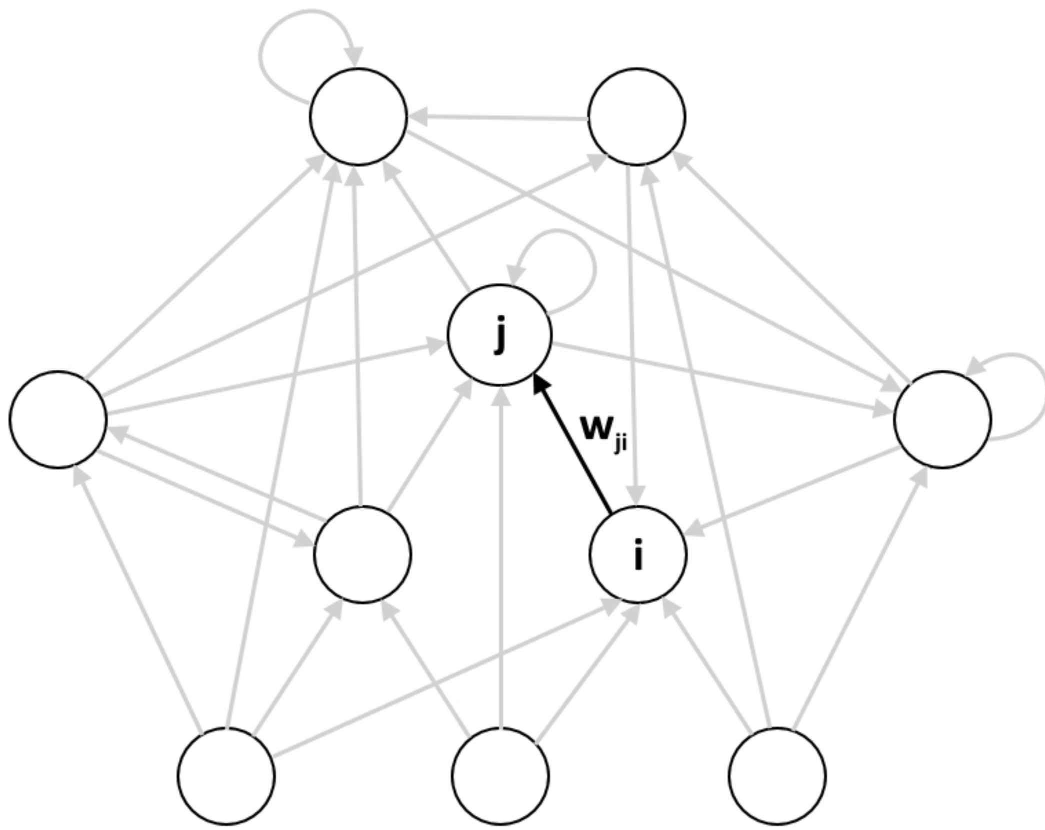
The Neural Network Model – Architecture

s computational units (neurons), indexed as $V = \{1, \ldots, s\}$, connected into a directed graph (V, E) where $E \subseteq V \times V$



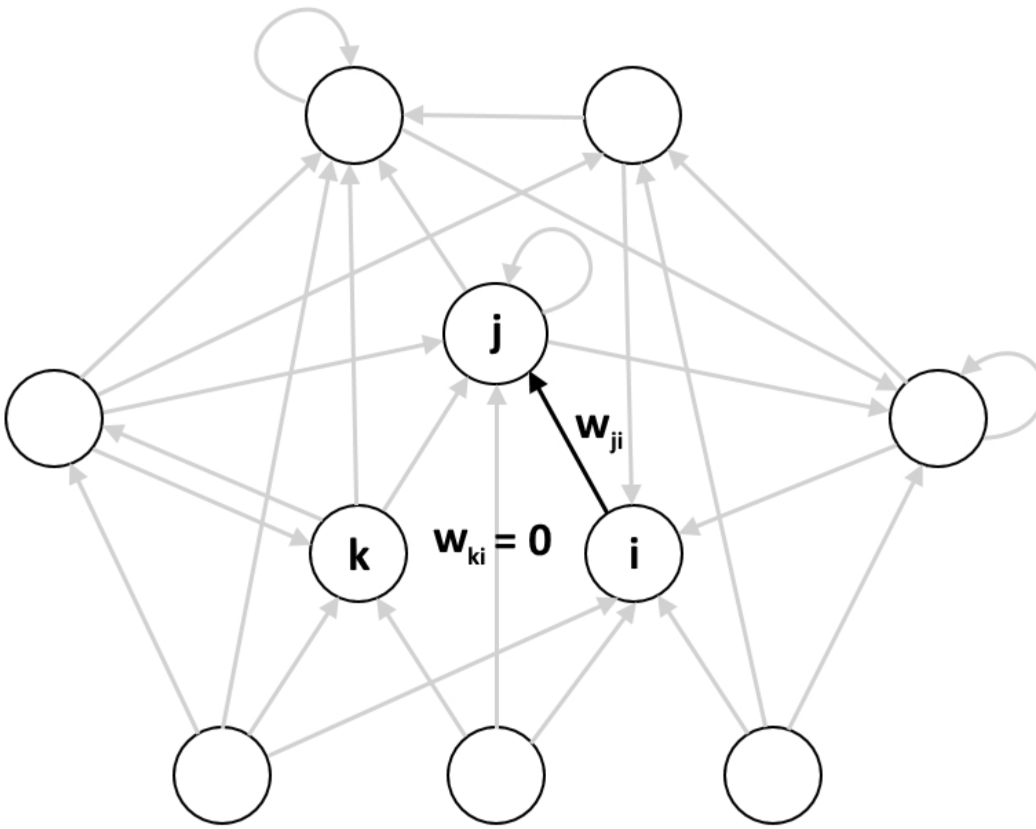
The Neural Network Model – Weights

each edge $(i, j) \in E$ from unit i to j is labeled with a real weight $w_{ji} \in \mathbb{R}$



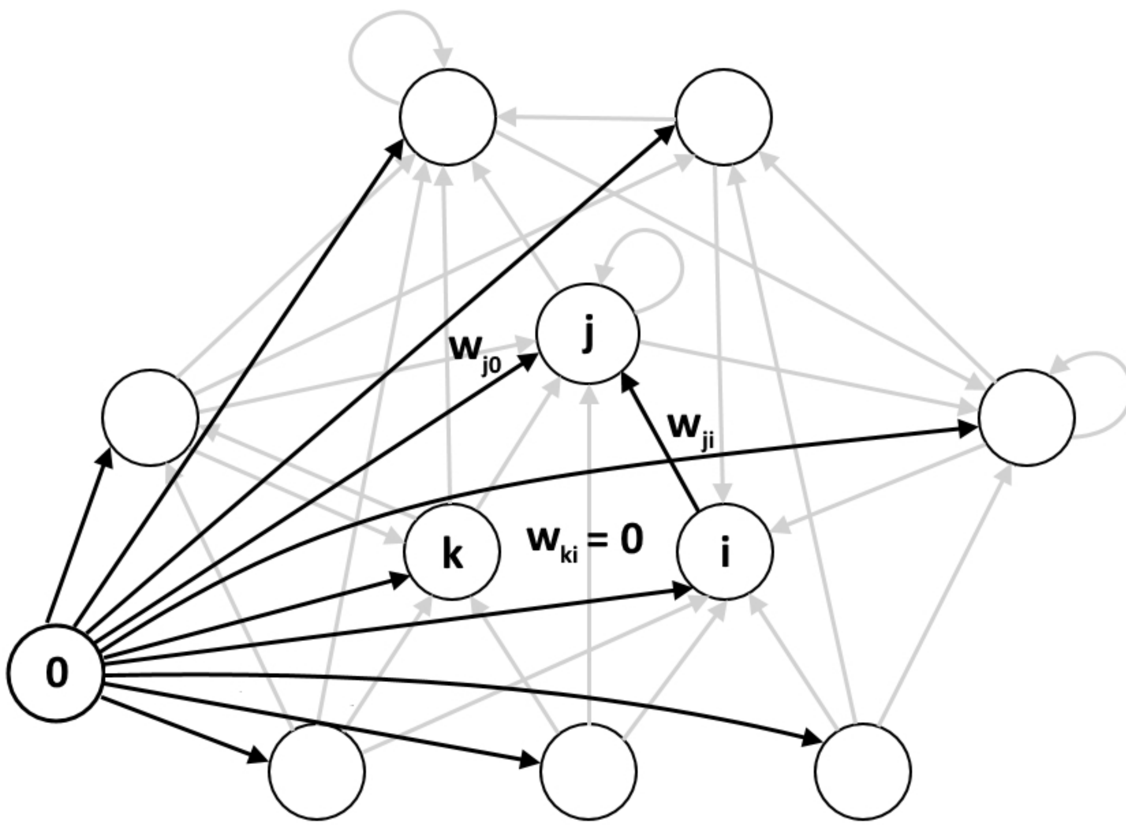
The Neural Network Model – Zero Weights

each edge $(i, j) \in E$ from unit i to j is labeled with a real weight $w_{ji} \in \mathbb{R}$ $(w_{ki} = 0$ iff $(i, k) \notin E)$



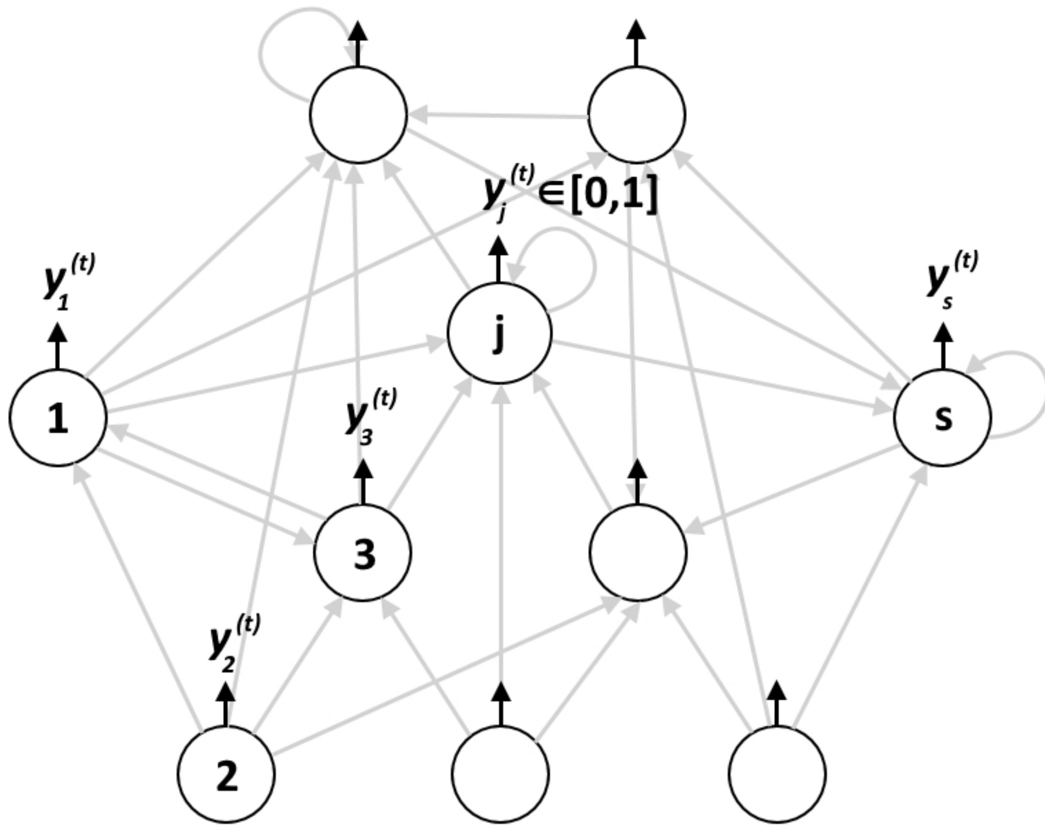
The Neural Network Model – Biases

each neuron $\boldsymbol{j} \in \boldsymbol{V}$ is associated with a real bias $\boldsymbol{w_{j0}} \in \mathbb{R}$
(i.e. a weight of $(\boldsymbol{0}, \boldsymbol{j}) \in \boldsymbol{E}$ from an additional formal neuron $\boldsymbol{0} \in \boldsymbol{V}$)



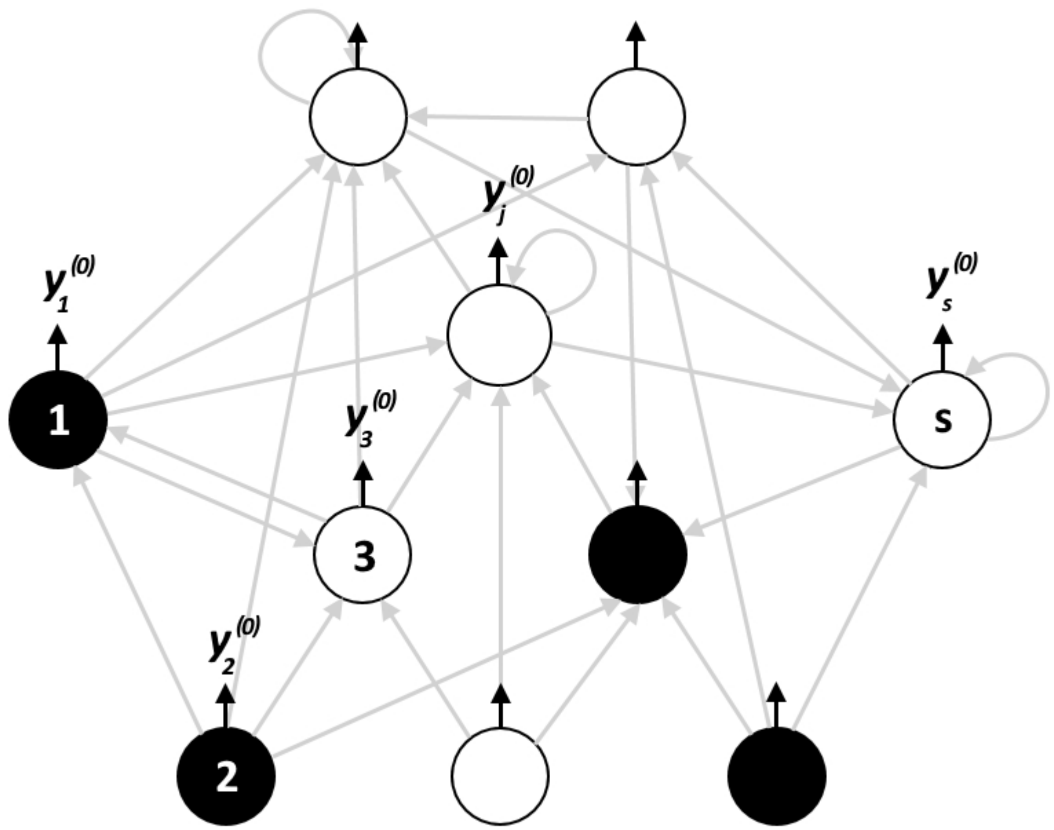
Discrete-Time Computational Dynamics – Network State

the evolution of global network state (output) $\mathbf{y}^{(t)} = (y_1^{(t)}, \ldots, y_s^{(t)}) \in [0,1]^s$
at discrete time instant $t = 0, 1, 2, \ldots$



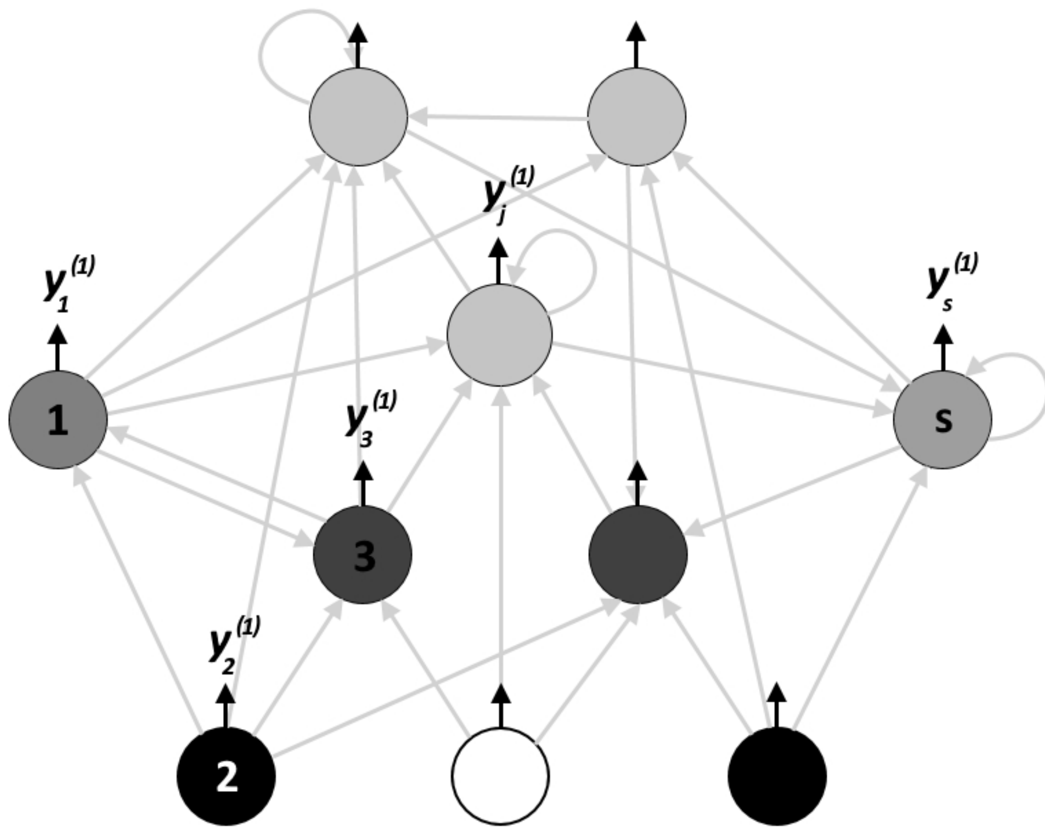
Discrete-Time Computational Dynamics – Initial State

$t = 0$: initial network state $\mathbf{y}^{(0)} \in \{0,1\}^s$



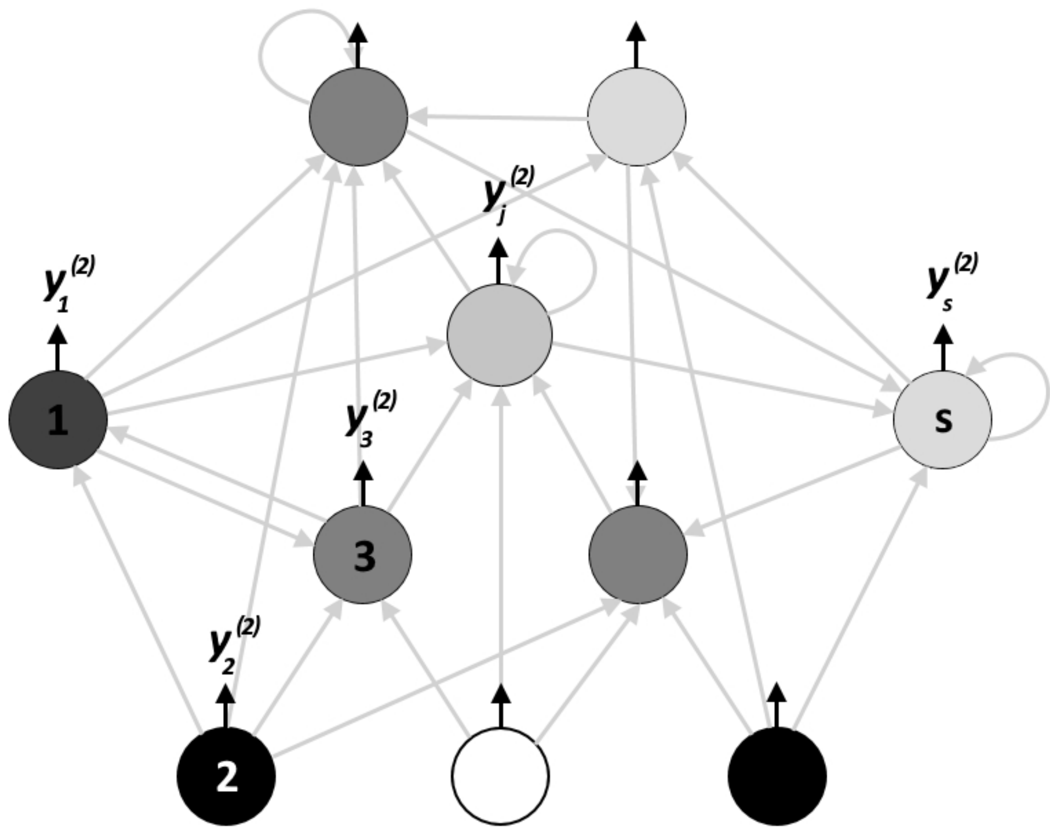
Discrete-Time Computational Dynamics: $t = 1$

$t = 1$: network state $\mathbf{y}^{(1)} \in [\mathbf{0}, \mathbf{1}]^s$



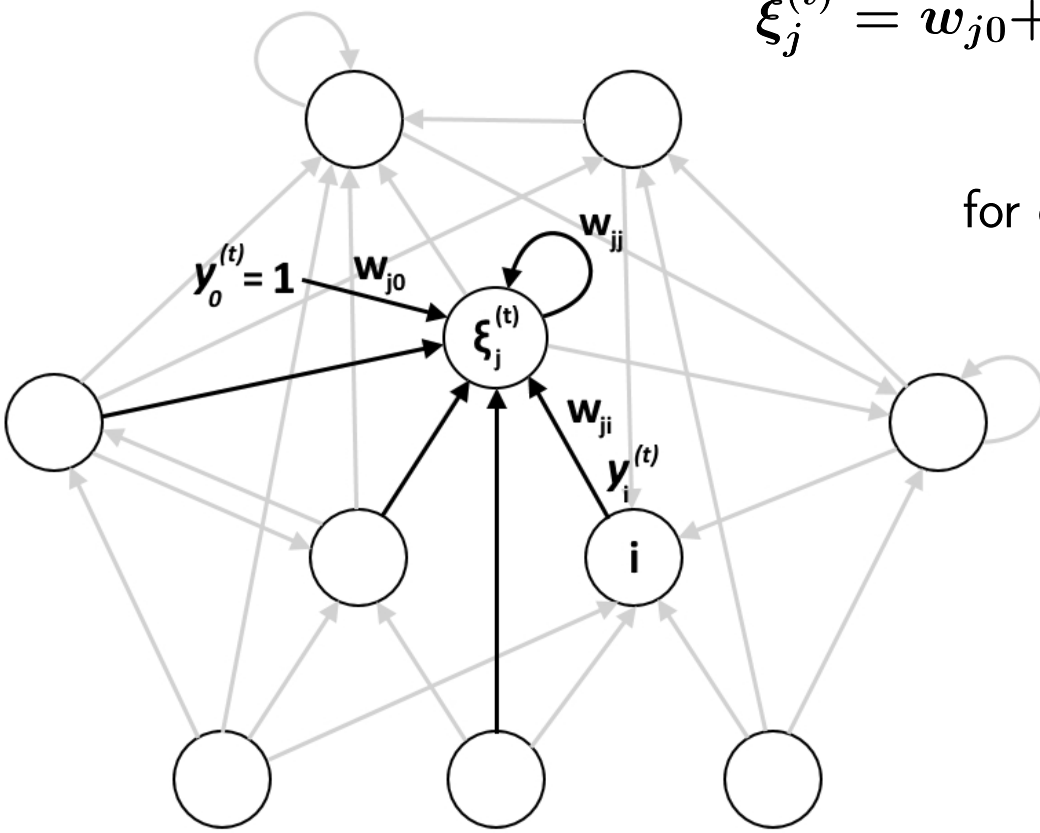
Discrete-Time Computational Dynamics: $t = 2$

$t = 2:$ network state $\mathbf{y}^{(2)} \in [0, 1]^s$



Discrete-Time Computational Dynamics – Excitations

at discrete time instant $t \geq 0$, an excitation is computed as



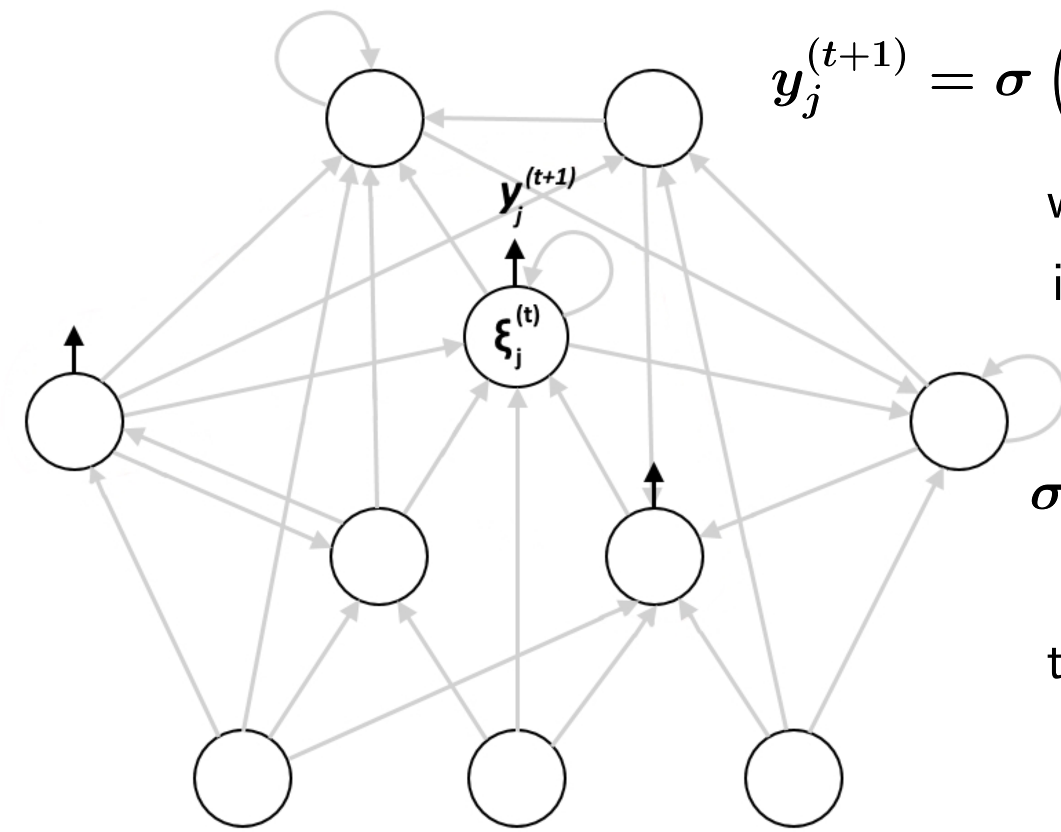
$$\xi_j^{(t)} = w_{j0} + \sum_{i=1}^{s} w_{ji} y_i^{(t)} = \sum_{i=0}^{s} w_{ji} y_i^{(t)}$$

for every $j \in \{1, \ldots, s\}$

where unit $0 \in V$ has constant output $y_0^{(t)} \equiv 1$ for every $t \geq 0$

Discrete-Time Computational Dynamics – Outputs

at the next time instant $t+1$, every neuron $j \in V$ updates its state in parallel (a so-called fully parallel mode):



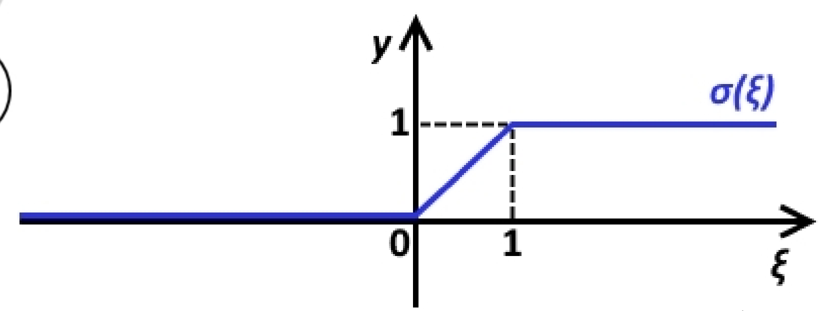
$$y_j^{(t+1)} = \sigma\left(\xi_j^{(t)}\right) \quad \text{for every } j = 1, \ldots, s$$

where $\sigma : \mathbb{R} \longrightarrow [0,1]$

is an activation function, e.g.

$$\sigma(\xi) = \begin{cases} 1 & \text{for } \xi \geq 1 \\ \xi & \text{for } 0 < \xi < 1 \\ 0 & \text{for } \xi \leq 0 \end{cases}$$

the saturated-linear function



The Computational Power of NNs – Motivations

- the potential and limits of general-purpose computation with NNs:

  What is ultimately or efficiently computable by particular NN models?

- idealized mathematical models of practical NNs which abstract away from implementation issues, e.g. analog numerical parameters are true real numbers

- methodology: the computational power and efficiency of NNs is investigated by comparing formal NNs to traditional computational models such as finite automata, Turing machines, Boolean circuits, etc.

- NNs may serve as reference models for analyzing alternative computational resources (other than time or memory space) such as analog state, continuous time, energy, temporal coding, etc.

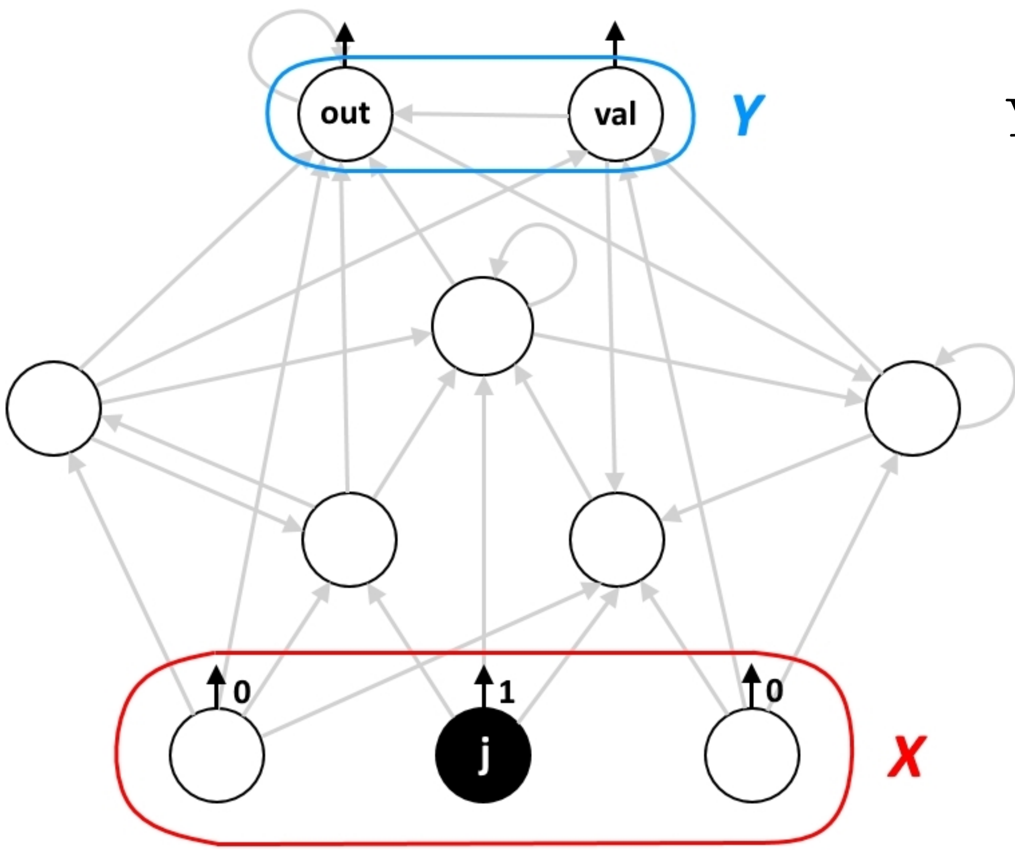
- NNs capture basic characteristics of biological nervous systems (plenty of densely interconnected simple unreliable computational units)

  $\longrightarrow$ computational principles of mental processes

Neural Networks As Formal Language Acceptors

a language $L \subseteq \Sigma^*$ over finite alphabet Σ represents a decision problem

$$y_{\text{out}}^{(T(n))} = \begin{cases} 1 & \text{if } \mathbf{x} \in L \\ 0 & \text{if } \mathbf{x} \notin L \end{cases} \qquad y_{\text{val}}^{(t)} = \begin{cases} 1 & \text{if } t = T(n) \\ 0 & \text{if } t \neq T(n) \end{cases}$$



$Y = \{\text{out}, \text{val}\}$ output neurons

$T(n)$ is the computation time in terms of input length $n \geq 0$

$d \geq 1$ is the time overhead for processing a single input symbol

$X = \text{enum}(\Sigma)$ input neurons
one-hot encoding

$$\uparrow \; y_j^{(d(i-1)+k)} = 1 \ \text{ iff } \ j = \text{enum}(x_i)$$

$$\mathbf{x} = x_1 \ldots x_{i-1} \longleftarrow x_i \longleftarrow x_{i+1} \ldots x_n \in \Sigma^*$$ input word

The Computational Power of NNs – Integer Weights

depends on the information content of weight parameters:

1. **integer** weights: finite automaton (FA) (Minsky, 1967)

$$\boldsymbol{w_{ji}} \in \mathbb{Z} \quad \longrightarrow \quad \text{excitations} \quad \boldsymbol{\xi_j} \in \mathbb{Z} \quad \longrightarrow \quad \text{states} \quad \boldsymbol{y_j} \in \{0, 1\}$$

$$\longrightarrow \quad \boldsymbol{2^s} \text{ global NN states } \mathbf{y} \in \{0,1\}^s \quad \sim \quad \text{FA states}$$

size-optimal implementations:

- $\boldsymbol{\Theta}\left(\sqrt{\boldsymbol{m}}\right)$ neurons for a deterministic FA with $\boldsymbol{m}$ states (Indyk, 1995; Horne, Hush, 1995)

- $\boldsymbol{\Theta}(\boldsymbol{m})$ neurons for a regular expression of length $\boldsymbol{m}$ (Šíma, Wiedermann 1998)

The Computational Power of NNs – Rational Weights

depends on the information content of weight parameters:

2. **rational** weights: Turing machine (Siegelmann, Sontag, 1995)

- $\boldsymbol{w_{ji}} \in \mathbb{Q}$ are fractions $\frac{\boldsymbol{p}}{\boldsymbol{q}}$ where $\boldsymbol{p} \in \mathbb{Z}$, $\boldsymbol{q} \in \mathbb{N}$
- NNs compute algorithmically solvable problems
- real-time simulation of TMs $\longrightarrow$ polynomial time $\equiv$ complexity class P
- a universal NN with 25 neurons (Indyk, 1995)

  $\longrightarrow$ the halting problem of whether a small NN terminates its computation, is algorithmically undecidable

The Computational Power of NNs – Real Weights

depends on the information content of weight parameters:

3. arbitrary **real** weights: "super-Turing" computation (Siegelmann, Sontag, 1994)

- $w_{ji} \in \mathbb{R}$, e.g. irrational weights $\sqrt{2}$, π
- infinite precision of **ONE** real weight (vs. an algorithm has a finite description)
  can encode any function f: $\mathbf{0\,.\,code(}C_1\mathbf{)\,code(}C_2\mathbf{)\,code(}C_3\mathbf{)\ldots}$
  (code(C_n) encodes the circuit C_n computing f for inputs of length n)

  $\longrightarrow$ exponential time $\equiv$ any I/O mapping
  (including algorithmically undecidable problems)

- polynomial time $\equiv$ nonuniform complexity class P/poly:

  problems solvable by a polynomial-time (P) algorithm that for input $x \in \Sigma^*$ of length $n = |x|$, receives an external advise: a string $s(n) \in \Sigma^*$ of polynomial length $|s(n)| = O(n^c)$ (poly), which depends only on n

The Computational Power of NNs – A Summary

depends on the information content of weight parameters:

1. **integer** weights: finite automaton
2. **rational** weights: Turing machine
   polynomial time $\equiv$ complexity class P
3. arbitrary **real** weights: "super-Turing" computation
   polynomial time $\equiv$ nonuniform complexity class P/poly
   exponential time $\equiv$ any I/O mapping

Neural Networks Between Rational and Real Weights

1. **integer** weights: finite automaton

2. **rational** weights: Turing machine

   polynomial time $\equiv$ **P**

   polynomial time & increasing **Kolmogorov complexity** of real weights:

   the length of the shortest program (in a fixed programming language) that produces a real weight,

   e.g. $K\left(\text{"}\sqrt{2}\text{"}\right) = O(1)$, $K(\text{"random strings"}) = n + O(1)$

   $\equiv$ a proper hierarchy of nonuniform complexity classes **between P and P/poly**

   (Balcázar, Gavaldà, Siegelmann, 1997)

3. arbitrary **real** weights: "super-Turing" computation

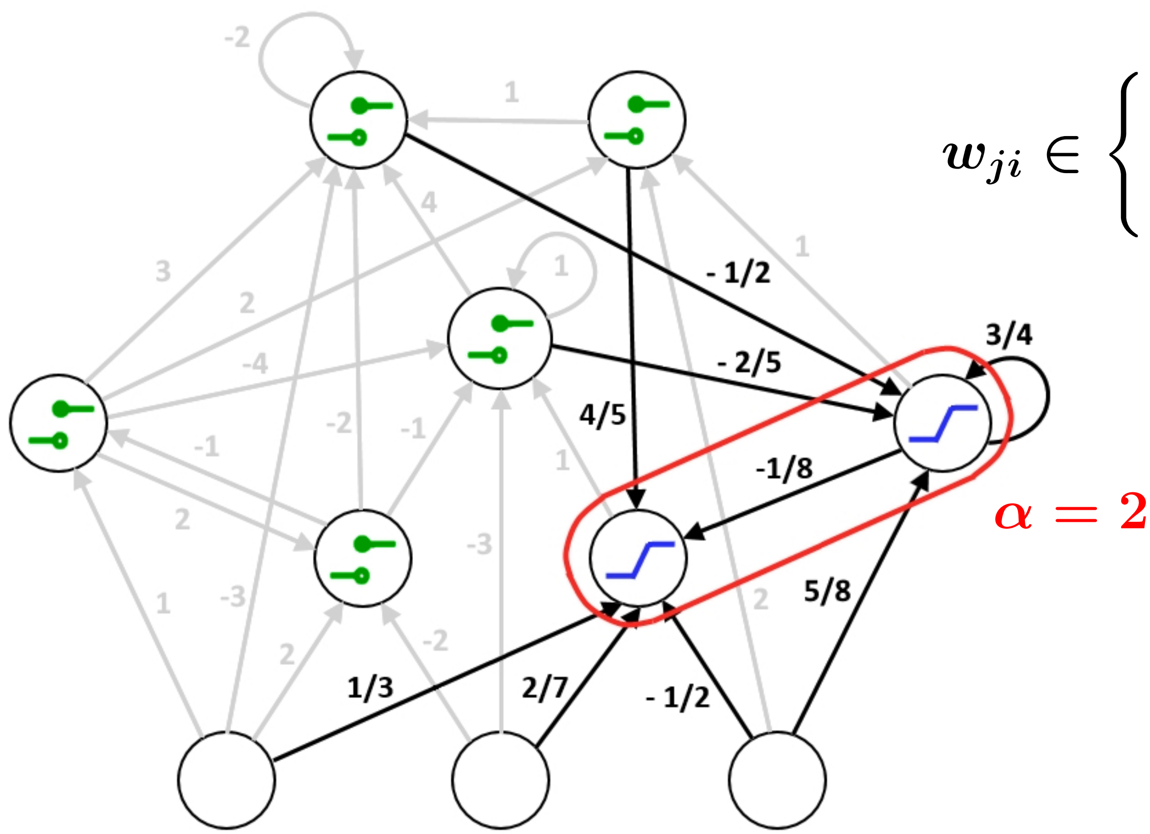
   polynomial time $\equiv$ **P/poly**

Neural Networks Between Integer and Rational Weights

from **integer** to **rational** weights

α**ANN** $=$ a **binary-state** NN with **integer** weights

$+\ \alpha$ **extra analog-state** neurons with **rational** weights



$$w_{ji} \in \begin{cases} \mathbb{Q} & j = 1, \ldots, \alpha \\ \mathbb{Z} & j = \alpha + 1, \ldots, s \end{cases}$$

$$i \in \{0, \ldots, s\}$$

Neural Networks with Increasing Analogicity

from **binary** $(\{0,1\})$ to **analog** $([0,1])$ states of neurons

α**ANN** $=$ a **binary-state** NN with **integer** weights

$+\ \alpha$ **extra** **analog-state** neurons with **rational** weights

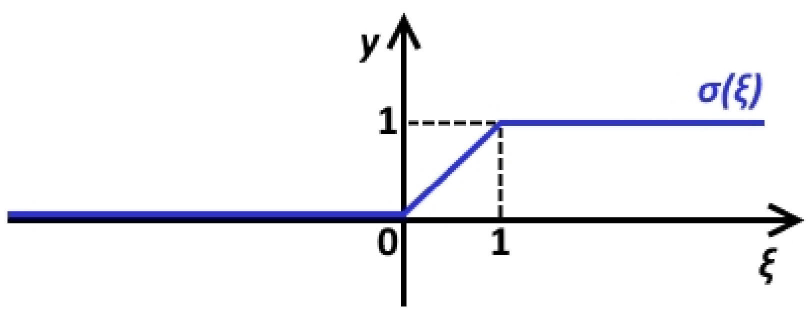
$$y_j^{(t+1)} = \sigma_j\left(\sum_{i=0}^{s} w_{ji} y_i^{(t)}\right) \qquad j = 1,\ldots,s$$

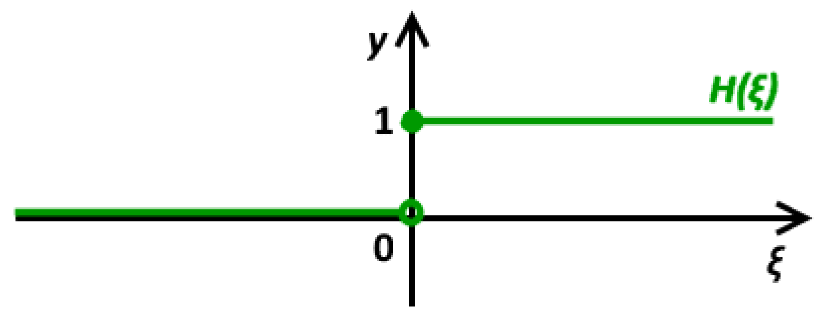
updating the states of neurons

$$\sigma_j(\xi) = \begin{cases} \sigma(\xi) = \begin{cases} 1 & \text{for } \xi \geq 1 \\ \xi & \text{for } 0 < \xi < 1 \\ 0 & \text{for } \xi \leq 0 \end{cases} & j = 1,\ldots,\alpha \\ \\ H(\xi) = \begin{cases} 1 & \text{for } \xi \geq 0 \\ 0 & \text{for } \xi < 0 \end{cases} & j = \alpha+1,\ldots,s \end{cases}$$

saturated-linear function (for $j = 1,\ldots,\alpha$)

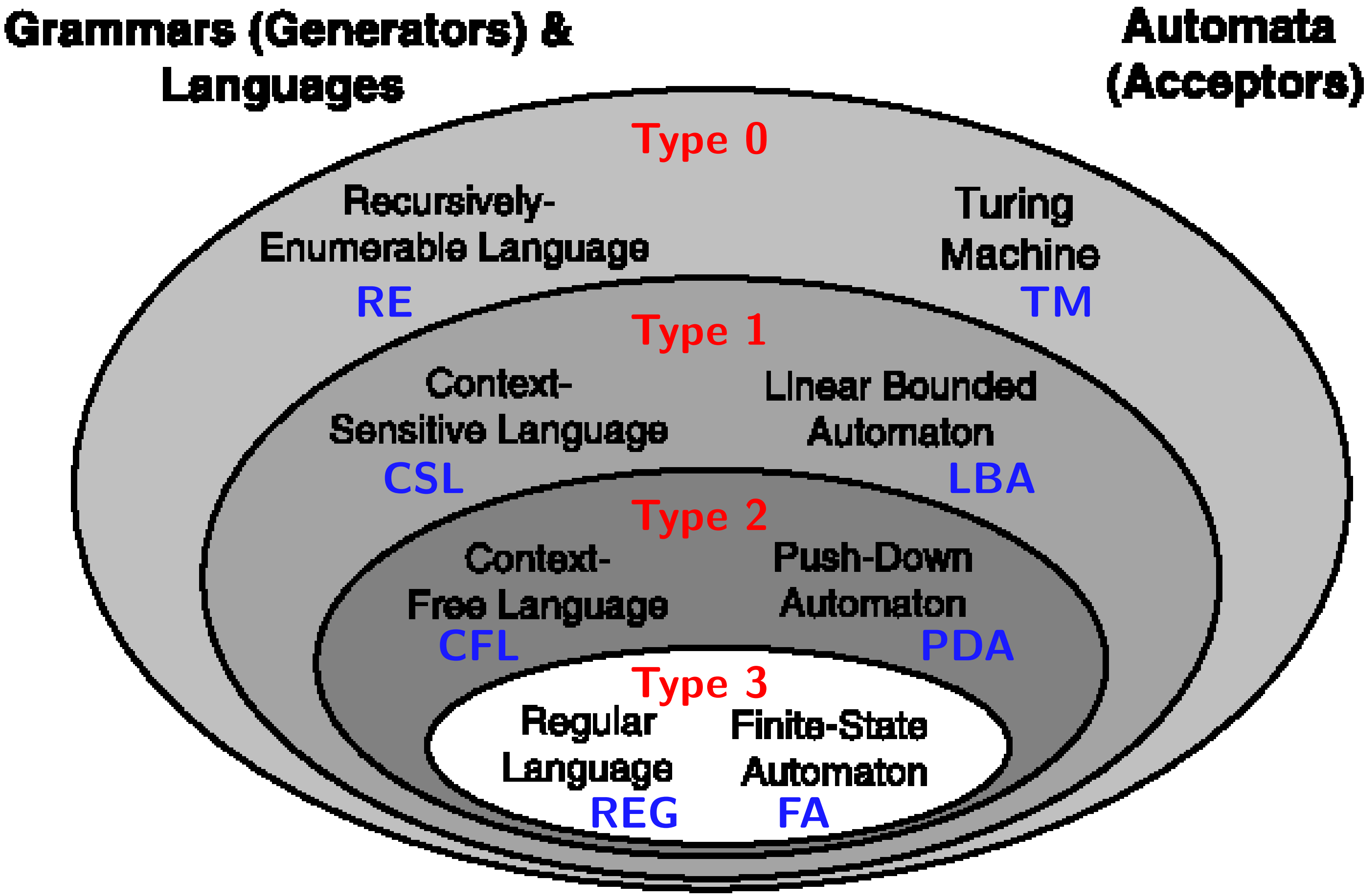
Heaviside function (for $j = \alpha+1,\ldots,s$)





The Chomsky Formal Language Hierarchy

from finite automata to Turing machines



The Analog Neuron Hierarchy (ANH)

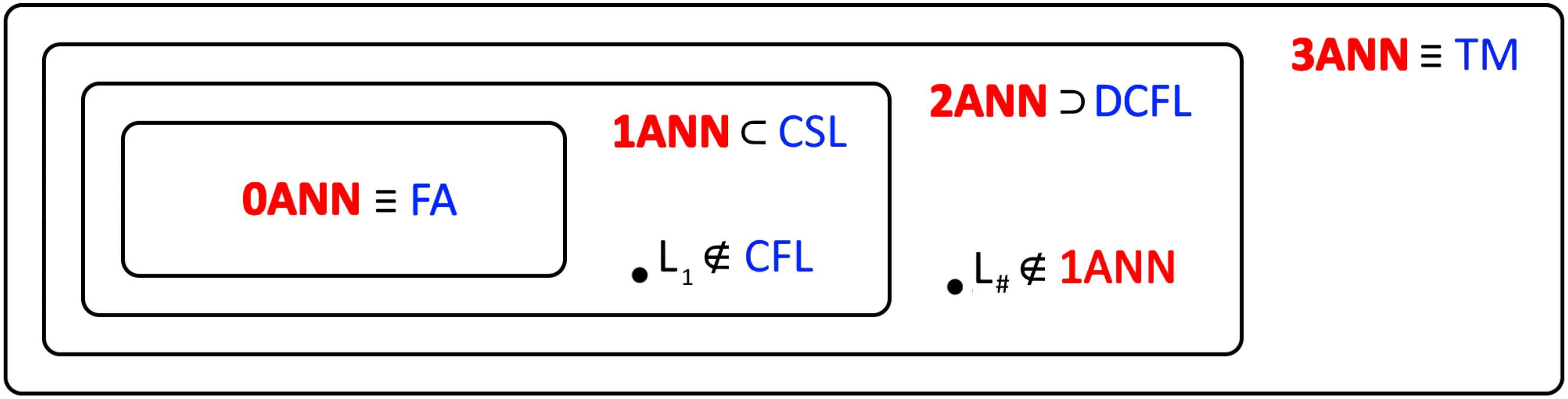
the computational power of α**ANNs**

increases with the number α of extra analog-state neurons:

$$
\begin{array}{ccccccccccccccc}
 & & \textbf{integer}\text{ weights} & & & & & & & & & & \textbf{rational}\text{ weights} & & \\
 & & \downarrow & & & & & & & & & & \downarrow & & \\
\textbf{FA} & \equiv & \textbf{REG} & = & \textbf{0ANN} & \subseteq & \textbf{1ANN} & \subseteq & \textbf{2ANN} & \subseteq & \textbf{3ANN} & \subseteq \ldots & = & \textbf{RE} & \equiv \ \textbf{TM} \\
 & & \uparrow & & & & & \times & & & & & & \uparrow & \\
 & & \textbf{Type 3} & & & & & \textbf{Chomsky hierarchy} & & & & & & \textbf{Type 0} & \\
 & & & & & & & \textbf{Type 1, 2} \ \ \textbf{?} & & & & & & &
\end{array}
$$

(the notation α**ANN** is also used for the class of languages accepted by αANNs)

The Analog Neuron Hierarchy as a Chomsky-Like NN Hierarchy



the **separation** of the first two levels **0ANN** $\overset{L_1}{\subsetneqq}$ **1ANN** $\overset{L_\#}{\subsetneqq}$ **2ANN**:

- LBA simulates 1ANN: **1ANN** $\subset$ **CSL** (Type 1)
- 1ANN accepts a non-CFL L_1: **1ANN** $\not\subset$ **CFL** (Type 2)

  $$L_1 = \left\{ x_1 \ldots x_n \in \{0,1\}^* \,\middle|\, \sum_{k=1}^{n} x_{n-k+1} \left(\tfrac{3}{2}\right)^{-k} < 1 \right\} \in \text{1ANN} \setminus \text{CFL}$$

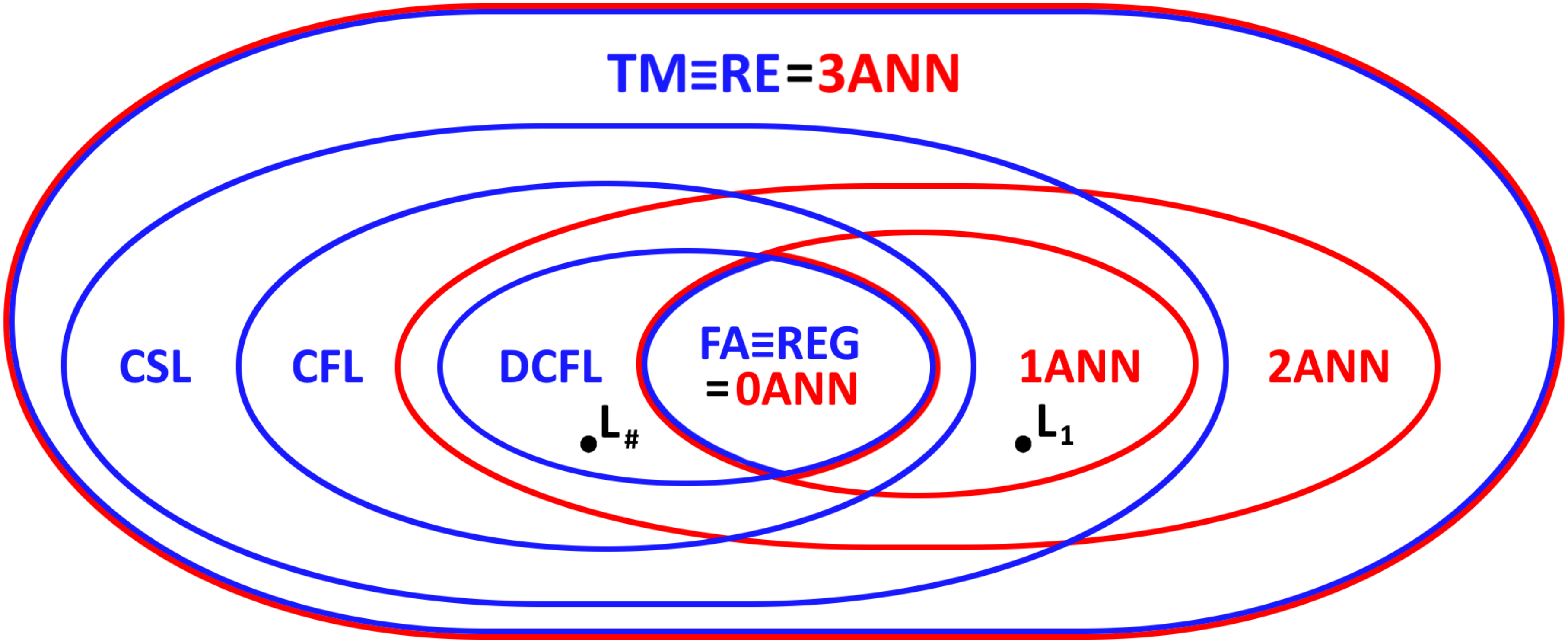
- 2ANN simulates deterministic PDA (DPDA $\equiv$ DCFL): **DCFL** $\subset$ **2ANN**
- 1ANN cannot count up to n (even with real weights): **DCFL** $\not\subset$ **1ANN**

  $$L_\# = \left\{ 0^n 1^n \,\middle|\, n \geq 1 \right\} \in \text{DCFL} \setminus \text{1ANN}$$

the **collapse** to the third level **3ANN** $=$ **4ANN** $= \ \ldots \ =$ **RE** $\equiv$ **TM** (Type 0):

- 3ANN simulates TM

The Chomsky Hierarchy vs. the Analog Neuron Hierarchy



the separation of some classes is still open, e.g. 2ANN $\overset{?}{\subsetneq}$ 3ANN, 1ANN $\cap$ CFL $\overset{?}{=}$ REG

the intermediate levels of the ANH and the Chomsky hierarchy seem incomparable

Positional Numeral Systems With Non-Integer Base

generalization of decimal expansions, which uses also non-integer numbers as the base and digits of a positional numeral system:

- $\beta \in \mathbb{R}$ is a real base (radix) such that $|\beta| > 1$
- $A \subset \mathbb{R}$ is a finite set of real digits such that $|A| \geq 2$

a finite β-expansion represents a number x in base β with digits a_i from A as

$$x = (0\,.\,a_1 \ldots a_n)_\beta = a_1\beta^{-1} + a_2\beta^{-2} + a_3\beta^{-3} + \cdots + a_n\beta^{-n} = \sum_{k=1}^{n} a_k\beta^{-k}$$

Examples:

1. $\beta = 10$, $A = \{0, 1, 2, \ldots, 9\}$
   decimal expansion of $\frac{3}{4} = (0\,.\,75)_{10} = 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$
2. $\beta = 2$, $A = \{0, 1\}$
   binary expansion of $\frac{3}{4} = (0\,.\,11)_2 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$
3. $\beta = \frac{5}{2}$, $A = \left\{\frac{5}{16}, \frac{7}{4}\right\}$
   $\frac{5}{2}$-expansion of $\frac{3}{4} \;=\; \left(0\;.\;\frac{7}{4}\;\frac{5}{16}\right)_{\frac{5}{2}} = \frac{7}{4} \cdot \left(\frac{5}{2}\right)^{-1} + \frac{5}{16} \cdot \left(\frac{5}{2}\right)^{-2}$

(Infinite) β-Expansions

introduced by Rényi (1957) and studied by Parry (1960); still an active research field with applications in coding theory, algorithmic complexity of arithmetic operations, models of quasicrystals, etc. (e.g. a research group at FNSPE CTU, Prague)

an infinite β-expansion of number x over digits a_i from A:

$$x = (0\,.\,a_1 a_2 a_3 \cdots)_\beta = a_1\beta^{-1} + a_2\beta^{-2} + a_3\beta^{-3} + \cdots = \sum_{k=1}^{\infty} a_k \beta^{-k}$$

which is a convergent power series due to $|\beta| > 1$

Example: $\beta = \frac{3}{2}$, $A = \{0, 1\}$

$\frac{3}{2}$-expansion of $\frac{16}{45}$: $(0\,.\,000\,10\,10\,10\,10\,10\,\ldots)_{\frac{3}{2}} = (0\,.\,000\,\overline{10})_{\frac{3}{2}}$

$$= \left(\frac{3}{2}\right)^{-4} + \left(\frac{3}{2}\right)^{-6} + \left(\frac{3}{2}\right)^{-8} + \cdots = \sum_{k=2}^{\infty} \left(\frac{3}{2}\right)^{-2k} = \sum_{k=2}^{\infty} \left(\frac{4}{9}\right)^{k} = \frac{16}{45}$$

(a geometric series)

Existence of β-Expansions

Let $\beta > 1$ and $A = \{\alpha_1, \ldots, \alpha_p\}$ where $\alpha_1 < \alpha_1 < \cdots < \alpha_p$.

Then every number in the interval $\left[\frac{\alpha_1}{\beta-1}, \frac{\alpha_p}{\beta-1}\right]$ has a β-expansion

iff $\max_{1<j\leq p}(\alpha_j - \alpha_{j-1}) \leq \frac{\alpha_p-\alpha_1}{\beta-1}$. (Pendicini, 2005)

Examples:

1. $\beta > 1$, $A = \{0, 1, \ldots, \lceil\beta\rceil - 1\}$ containing the standard integer digits

   every number in the interval $D_\beta = \left(0, \frac{\lceil\beta\rceil-1}{\beta-1}\right)$ (even $\overline{D_\beta}$) has a β-expansion,

   note that $(0,1) \subseteq D_\beta$, e.g. $D_\beta = (0,1)$ for integer base β

2. $\beta = 3$, $A = \{0, 2\}$ (i.e. $2 \not\leq \frac{2-0}{3-1} = 1$)

   any number from the complement of the Cantor ternary set

$$\bigcup_{n=0}^{\infty} \bigcup_{k=0}^{3^n-1} \left(\frac{3k+1}{3^n+1}, \frac{3k+2}{3^n+1}\right) \subset (0,1)$$ has no 3-expansion

   (including iteratively the open middle third from a set of line segments, starting with (0,1))

Uniqueness of β-Expansions for Integer Base β

for an integer base $\beta > 1$ and the standard digits, $A = \{0, 1, \ldots, \beta - 1\}$,

almost any number from the interval $D_\beta = (0, 1)$ has a **unique** β-expansion,

e.g. the unique decimal expansion of $\frac{\sqrt{2}}{2} = (0\,.\,70710678118\ldots)_{10}$,

except for numbers with a finite β-expansion, which have **two distinct** (infinite) β-expansions,

e.g. two (infinite) decimal expansions of

$$\frac{3}{4} = (0\,.\,75)_{10} = (0\,.\,75000\ldots)_{10} = (0\,.\,74999\ldots)_{10}$$

Uniqueness of β-Expansions for Non-Integer Base β

for a non-integer base, almost every number has **infinitely (uncountably) many distinct** β-expansions (Sidorov, 2003)

Example: $1 < \beta < 2, \quad A = \{0, 1\}, \quad D_\beta = \left(0\,,\, \frac{1}{\beta-1}\right)$

- $1 < \beta < \varphi$ where $\varphi = (1+\sqrt{5})/2 \approx 1.618034$ is the golden ratio:

  every $x \in D_\beta$ has uncountably many distinct β-expansions (Erdös et al.,1990)

- $\varphi \le \beta < q$ where $q \approx 1.787232$ is the Komornik-Loreti constant
  (i.e. $\sum_{k=1}^{\infty} t_k q^{-k} = 1$ where $t_k = \text{parity}(\text{bin}(k))$ is the Thue-Morse sequence):

  countably many $x \in D_\beta$ have unique β-expansions (Glendinning,Sidorov,2001),

  e.g. the unique $\frac{5}{3}$-expansions of $\frac{9}{16}\left(\frac{3}{5}\right)^{k-1} = \left(0\,.\,(0)^k\,\overline{10}\right)_{\frac{5}{3}}$ for $k \ge 0$

  vs. countably many distinct φ-expansions of $1 = \left(0\,.\,(10)^k\,0\overline{1}\right)_\varphi$ for $k \ge 0$

- $q \le \beta < 2$: uncountably many $x \in D_\beta$ have unique β-expansions

partially generalizes to $\beta > 2$ and arbitrary A: two critical bases $1 < \varphi_A \le q_A$ such that the number of unique β-expansions is finite if $1 < \beta < \varphi_A$, countable if $\varphi_A < \beta < q_A$, and uncountable if $\beta > q_A$ (Komornik,Pedicini,2016)

Eventually Periodic β-Expansions

$$(0\,.\,a_1a_2\ldots a_{k_1}\,\overline{a_{k_1+1}a_{k_1+2}\ldots a_{k_2}})_\beta = (0\,.\,a_1a_2\ldots a_{k_1})_\beta + \beta^{-k_1}\varrho$$

where

- $a_1a_2\ldots a_{k_1} \in A^{k_1}$ is a preperiodic part of length $k_1 \geq 0$
  (purely periodic β-expansions for $k_1 = 0$)
- $a_{k_1+1}a_{k_1+2}\ldots a_{k_2} \in A^m$ is a repetend of $m = k_2 - k_1 > 0$ repeating digits
- $\varrho = (0\,.\,\overline{a_{k_1+1}a_{k_1+2}\ldots a_{k_2}})_\beta = \dfrac{\sum_{k=1}^{m} a_{k_1+k}\,\beta^{-k}}{1-\beta^{-m}}$ is a periodic point

Example: $\beta = \frac{3}{2}$, $A = \{0,1\}$

$$\frac{22}{15} = (0\,.\,1\,\overline{10})_{\frac{3}{2}} = (0\,.\,1)_{\frac{3}{2}} + \left(\frac{3}{2}\right)^{-1}\cdot\varrho = \left(\frac{3}{2}\right)^{-1} + \left(\frac{3}{2}\right)^{-1}\cdot(0\,.\,\overline{10})_{\frac{3}{2}}$$

where $\varrho = (0\,.\,\overline{10})_{\frac{3}{2}} = \sum_{k=0}^{\infty}\left(\frac{3}{2}\right)^{-2k-1} = \dfrac{1\cdot\left(\frac{3}{2}\right)^{-1} + 0\cdot\left(\frac{3}{2}\right)^{-2}}{1-\left(\frac{3}{2}\right)^{-2}} = \dfrac{6}{5}$

Eventually Quasi-Periodic β-Expansions

$$(0\,.\,a_1 \ldots a_{k_1}\, a_{k_1+1} \ldots a_{k_2}\, a_{k_2+1} \ldots a_{k_3}\, a_{k_3+1} \ldots a_{k_4} \ldots)_\beta$$
$$= (0\,.\,a_1 a_2 \ldots a_{k_1})_\beta + \beta^{-k_1} \varrho$$

where

- $a_1 a_2 \ldots a_{k_1} \in A^{k_1}$ is a preperiodic part of length (purely quasi-periodic β-expansions for $k_1 = 0$)
- $a_{k_i+1} \ldots a_{k_{i+1}} \in A^{m_i}$ is a quasi-repetend of length $m_i = k_{i+1} - k_i > 0$
- $\varrho = (0\,.\,\overline{a_{k_i+1} \ldots a_{k_{i+1}}})_\beta = \dfrac{\sum_{k=1}^{m_i} a_{k_i+k}\, \beta^{-k}}{1 - \beta^{-m_i}}$ is the same periodic point for every $i \geq 1$

$\longrightarrow$ quasi-repetends can be interchanged with each other arbitrarily

- a generalization of eventually periodic β-expansions

$$a_{k_1+1} \ldots a_{k_2} = a_{k_2+1} \ldots a_{k_3} = a_{k_3+1} \ldots a_{k_4} = \cdots$$

Example: $\beta \approx 1.220744$ satisfying $\beta^4 - \beta - 1 = 0$ $(\star)$, $A = \{0, 1\}$

$$1 = (0\,.\,00\,010\,1000\,1000\,010 \ldots)_\beta = (0\,.\,00)_\beta + \beta^{-2} \varrho$$

where 00 is a preperiodic part and 010, 1000 are two quasi-repetends with same periodic point $\varrho = (0\,.\,\overline{010})_\beta = \frac{\beta^{-2}}{1-\beta^{-3}} \overset{\star}{=} \beta^2 \overset{\star}{=} \frac{\beta^{-1}}{1-\beta^{-4}} = (0\,.\,\overline{1000})_\beta$

An Example of Repetends With Unbounded Length

base $\beta = \frac{5}{2}$, digits $A = \{0\,,\ \frac{1}{2}\,,\ \frac{7}{4}\}$

for every $n \geq 0$, the quasi-repetends $\frac{7}{4}\ \underbrace{\frac{1}{2} \cdots \frac{1}{2}}_{n \text{ times}}\ 0 \in A^{n+2}$ have

the same periodic point $\varrho = \frac{3}{4}$:

$$\Big(0\,.\,\overline{\frac{7}{4}\ \underbrace{\frac{1}{2} \cdots \frac{1}{2}}_{n \text{ times}}\ 0}\Big)_{\frac{5}{2}} = \frac{\frac{7}{4} \cdot \left(\frac{5}{2}\right)^{-1} + \sum_{i=2}^{n+1} \frac{1}{2} \cdot \left(\frac{5}{2}\right)^{-i} + 0 \cdot \left(\frac{5}{2}\right)^{-n-2}}{1 - \left(\frac{5}{2}\right)^{-n-2}} = \frac{3}{4}$$

$\longrightarrow$ $\frac{3}{4}$ has uncountably many distinct quasi-periodic $\frac{5}{2}$-expansions:

$$\frac{3}{4} = \Big(0\,.\,\frac{7}{4}\ \underbrace{\frac{1}{2} \cdots \frac{1}{2}}_{n_1 \text{ times}}\ 0\ \frac{7}{4}\ \underbrace{\frac{1}{2} \cdots \frac{1}{2}}_{n_2 \text{ times}}\ 0\ \frac{7}{4}\ \underbrace{\frac{1}{2} \cdots \frac{1}{2}}_{n_3 \text{ times}}\ 0\ \frac{7}{4}\ \underbrace{\frac{1}{2} \cdots \frac{1}{2}}_{n_4 \text{ times}}\ 0\ \cdots\Big)_{\frac{5}{2}}$$

where $n_1, n_2, n_3, \ldots$ is any infinite sequence of nonnegative integers

(there are examples of exponentially many quasi-repetends in terms of their length)

Eventually Quasi-Periodic β-Expansions and Tail Sequences

$(r_n)_{n=0}^{\infty}$ is a tail sequence of β-expansion $\varepsilon = \big(0\,.\,a_1\,a_2\,a_3\,\ldots\big)_\beta$ if

$$r_n = (0\,.\,a_{n+1}a_{n+2}\ldots)_\beta = \sum_{k=1}^{\infty} a_{n+k}\beta^{-k} \quad \text{for every } n \geq 0$$

denote by $R_\varepsilon = \{r_n \mid n \geq 0\}$ its range

Lemma. *If R_ε is finite (i.e. the tail sequence contains a constant infinite subsequence), then the β-expansion ε is eventually quasi-periodic.*

Theorem. *Let β be a real algebraic number ($|\beta| > 1$) whose all conjugates β' (i.e. the other roots of minimal polynomial of β) meet $|\beta'| \neq 1$. Then a β-expansion ε is eventually quasi-periodic **iff** R_ε is finite.*

Theorem. *Let β be a real algebraic number ($|\beta| > 1$) whose conjugate β' meets $|\beta'| = 1$. Then there exists a finite set $A \subset \mathbb{Z}$ of integer digits and a quasi-periodic β-expansion ε over A of the number 0 that has infinite R_ε.*

(solves an important open problem in algebraic number theory)

Quasi-Periodic Numbers

a real number $x \in \mathbb{R}$ is β-quasi-periodic within A if **every** infinite β-expansion of x over A, is eventually quasi-periodic

Examples:

- x with no β-expansion at all, **is** formally quasi-periodic (e.g. any number from the complement of the Cantor ternary set is 3-quasi-periodic within $A = \{0, 2\}$)
- $x = \frac{3}{4}$ **is** $\frac{5}{2}$-quasi-periodic within $A = \left\{0\,,\,\frac{1}{2}\,,\,\frac{7}{4}\right\}$:
  all the $\frac{5}{2}$-expansions of $\frac{3}{4}$ using the digits from A, are eventually quasi-periodic
- $x = \frac{40}{57} = (0\,.\,0\,\overline{011})_{\frac{3}{2}}$ **is not** $\frac{3}{2}$-quasi-periodic within $A = \{0, 1\}$:
  the greedy (i.e. lexicographically maximal) $\frac{3}{2}$-expansion $(0\,.\,100000001\,\ldots)_{\frac{3}{2}}$ of $\frac{40}{57}$ is not eventually quasi-periodic

Theorem. *Let $\beta > 1$ be a Pisot number (i.e. a real algebraic integer whose all conjugates β' meet $|\beta'| < 1$) and $A \subset \mathbb{Q}(\beta)$. Then any $x \in \mathbb{Q}(\beta)$ is β-quasi-periodic within A.*

- $x = 1$ **is** β-quasi-periodic within $A = \{0, 1\}$ for the plastic constant $\beta \approx 1.324718$ (i.e. the minimal Pisot number satisfying $\beta^3 - \beta - 1 = 0$)

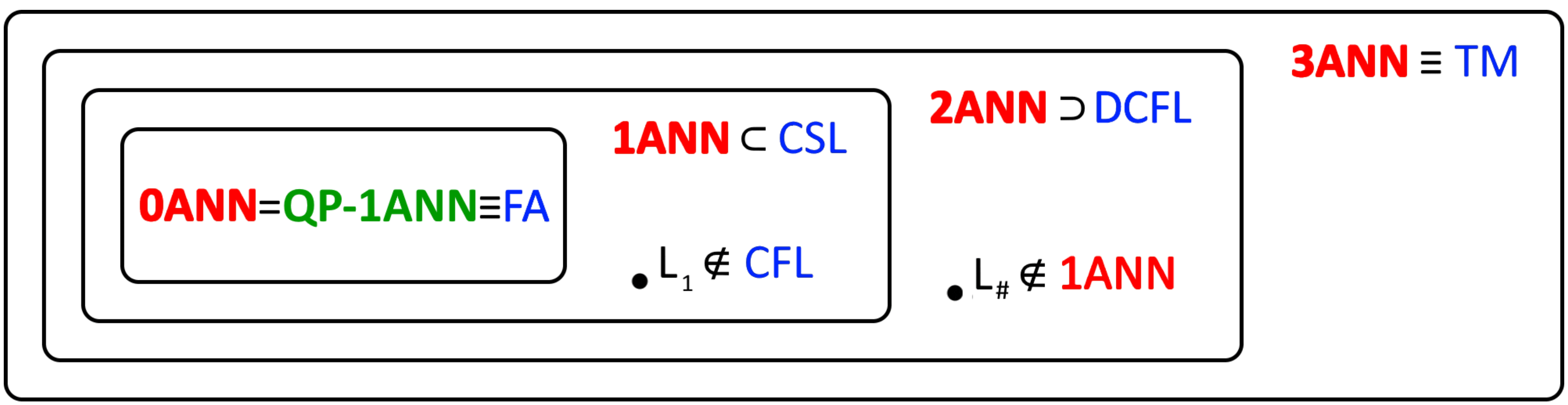
Quasi-Periodic 1ANN (QP-1ANN): for a 1ANN, denote:

- $\beta = 1/w_{11}$ is the base ($|\beta| > 1$) where w_{11} is the self-loop weight of the one analog-state neuron ($0 < |w_{11}| < 1$)
- $A = \left\{ \sum_{i=0\,;\,i\neq 1}^{s} \frac{w_{1i}}{w_{11}}\, y_i \;\middle|\; y_2,\ldots,y_s \in \{0,1\} \right\} \cup \{0,\beta\}$ are the digits
- $X = \left\{ \sum_{i=0\,;\,i\neq 1}^{s} \frac{w_{ji}}{w_{j1}}\, y_i \;\middle|\; j \neq 1\,,\; w_{j1} \neq 0\,,\; y_2,\ldots,y_s \in \{0,1\} \right\} \cup \{0,1\}$

we say that 1ANN (even with real weights) is quasi-periodic and denote **QP-1ANN** if every $x \in X$ is β-quasi-periodic within A

Example: 1ANN with rational weights $+$ the self-loop weight $w_{11} = 1/\beta$ where β is an integer or the plastic constant or the golden ratio

Theorem. QP-1ANN $=$ REG $=$ 0ANN $\equiv$ FA (Type 3)

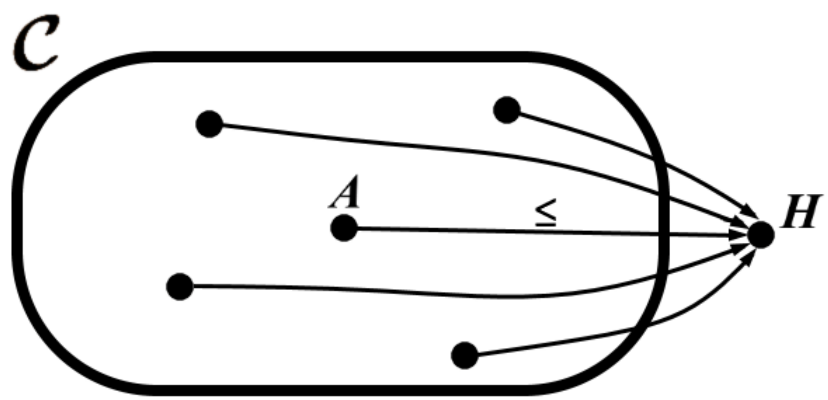


$\mathcal{C}$-Hard Problems

$\mathcal{C}$ is a complexity class of decision problems (i.e. formal languages)

$A \leq B$ is a reduction transforming a problem A to a problem B (a preorder), which is assumed not to have a higher computational complexity than $\mathcal{C}$

H is a $\mathcal{C}$-hard problem (under the reduction $\leq$) if for every $A \in \mathcal{C}$, $A \leq H$



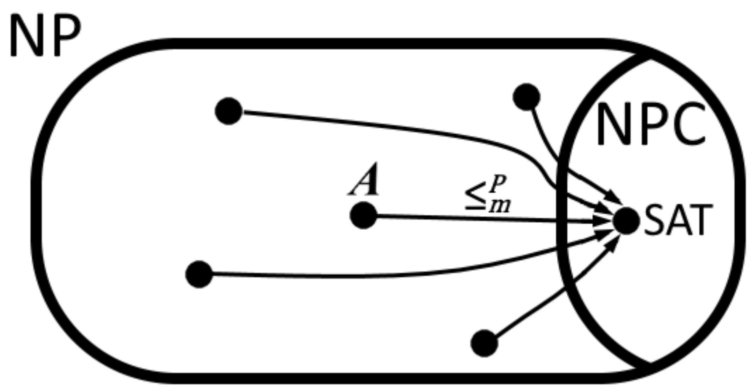
- If a $\mathcal{C}$-hard problem has a (computationally) "easy" solution, then each problem in $\mathcal{C}$ has an "easy" solution (via the reduction).
- If a $\mathcal{C}$-hard problem H is in $\mathcal{C}$ (a so-called $\mathcal{C}$-complete problem), then H belongs to the hardest problems in the class $\mathcal{C}$.

The Most Prominent Example: NP-Hard Problems

$\mathcal{C} =$ NP is the class of decision problems solvable in polynomial time by a nondeterministic Turing machine

$A \leq_m^P B$ is a polynomial-time many-one reduction (Karp reduction) from A to B

the satisfiability problem SAT is NP-hard: for every $A \in$ NP, $A \leq_m^P$ SAT

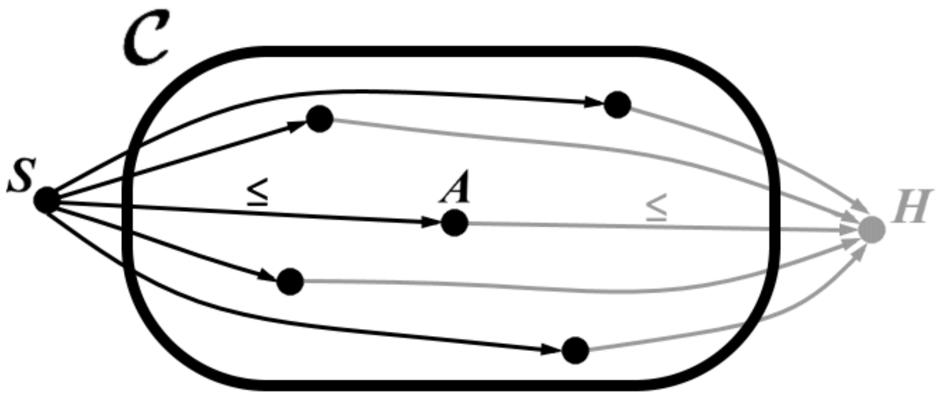


- If an NP-hard problem is polynomial-time solvable,
  then each NP problem would be solved in polynomial time (i.e. P = NP)
- The NP-hard problem SAT is in NP (i.e. SAT is NP-complete),
  that is, SAT belongs to the hardest problems (NPC) in the class NP.

$\mathcal{C}$-Simple Problems

a conceptual counterpart to $\mathcal{C}$-hard problems:

S is a $\mathcal{C}$-simple problem (under the reduction $\leq$) if for every $A \in \mathcal{C}$, $S \leq A$



- If a $\mathcal{C}$-simple problem S proves to be not "easy",
  e.g. S is not solvable by a machine M that can compute the reduction $\leq$,
  then all problems in $\mathcal{C}$ are not "easy", i.e. $\mathcal{C}$ cannot be solved by M.

  $\longrightarrow$ **New Proof Technique:** a lower bound known for one $\mathcal{C}$-simple problem S extends to the whole class of problems $\mathcal{C}$

- If a $\mathcal{C}$-simple problem S is in $\mathcal{C}$, then S is the simplest problem in the class $\mathcal{C}$.

A Trivial Example: SAT is simple for the class of NP-hard problems under $\leq_m^P$

A Nontrivial Example of a $\mathcal{C}$-Simple Problem

$\mathcal{C}$ = DCFL' = DCFL \ REG

is the class of non-regular deterministic context-free languages

$L_1 \leq_{tt}^{A} L_2$ is a truth-table reduction (a stronger Turing reduction) from L_1 to L_2

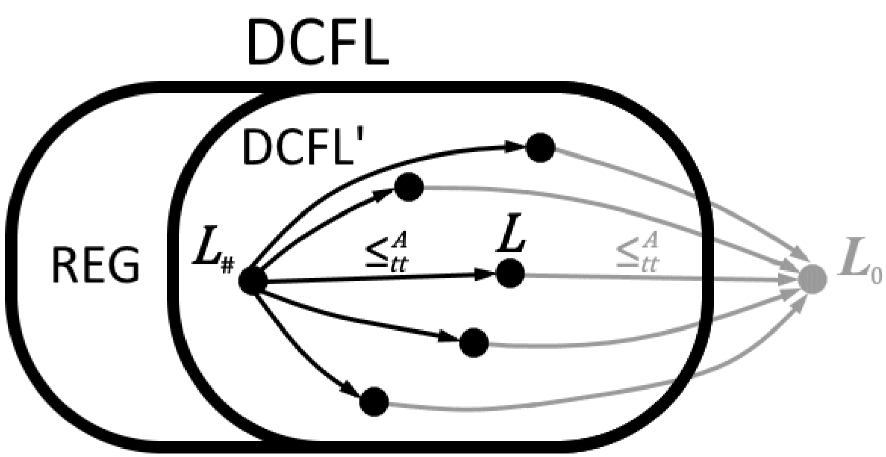
implemented by a Mealy machine with the oracle L_2

The Technical Result:

the language $L_\# = \{0^n 1^n \mid n \geq 1\}$ over the binary alphabet $\{0, 1\}$ is DCFL'-simple under the reduction $\leq_{tt}^{A}$: for every $L \in$ DCFL', $L_\# \leq_{tt}^{A} L$

$\longrightarrow$ $L_\# \in$ DCFL' is the *simplest* non-regular deterministic context-free languages

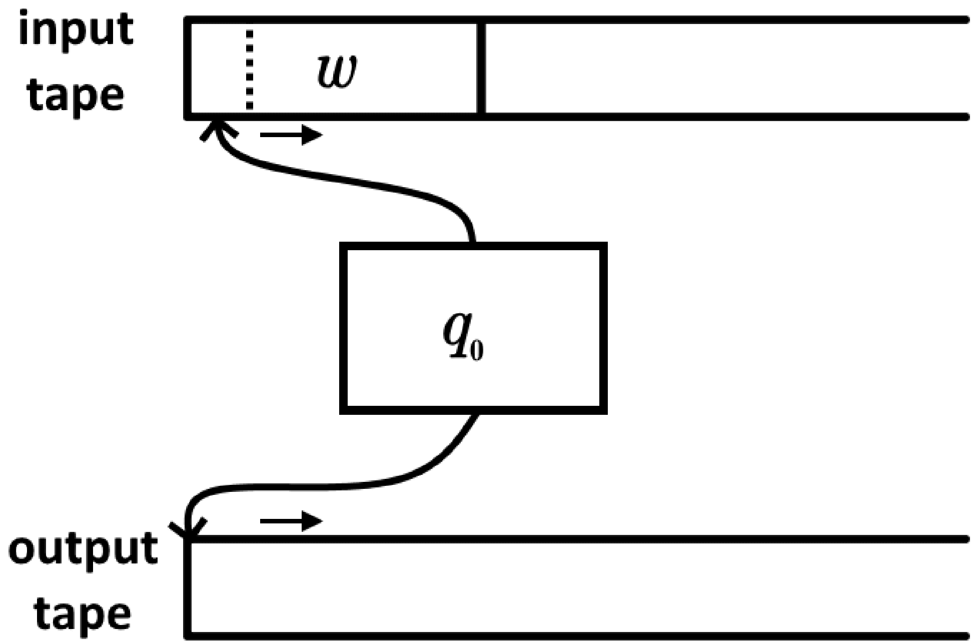
cf. the *hardest* context-free language L_0 due to `S. Greibach (1973)` is CFL-hard



Mealy Machines

$\mathcal{A}$ is a Mealy Machine with an input/output alphabet Σ/Δ

i.e. a deterministic finite automaton with an output tape:



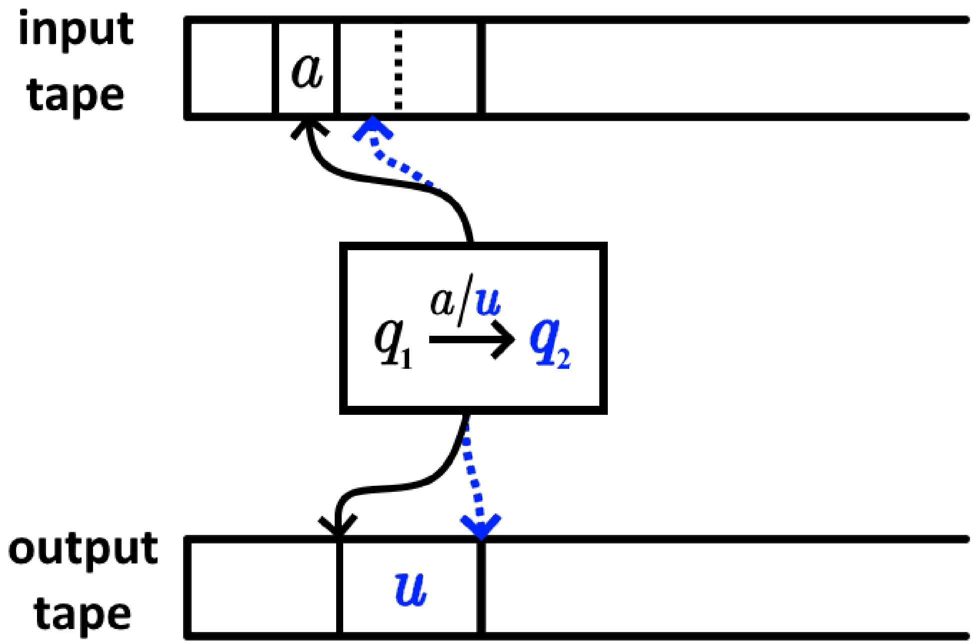
input word $w \in \Sigma^*$

initial state q_0

Mealy Machines

$\mathcal{A}$ is a Mealy Machine with an input/output alphabet Σ/Δ

i.e. a deterministic finite automaton with an output tape:



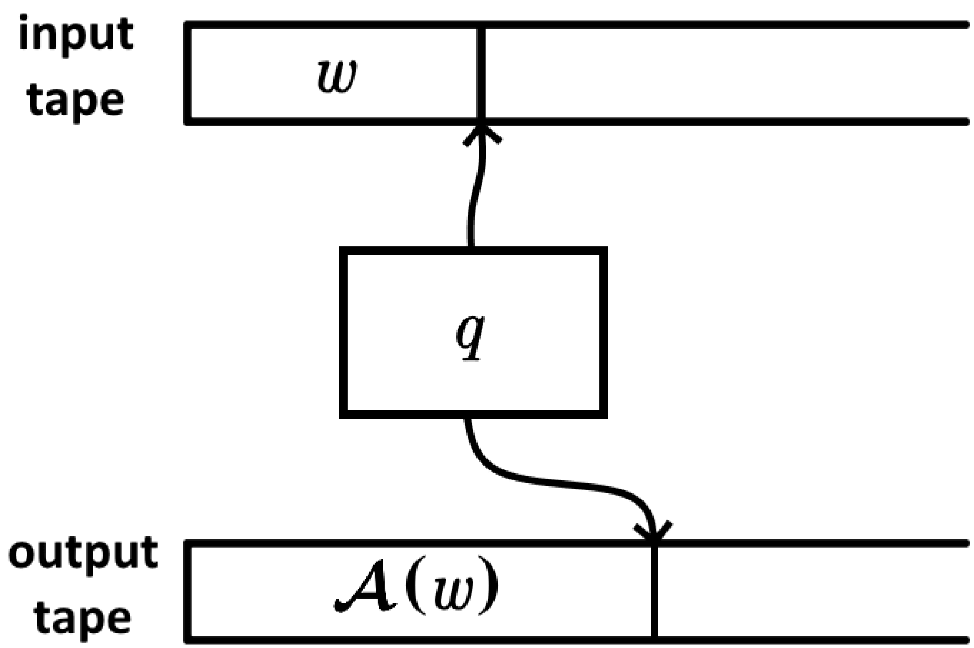
current input symbol $a \in \Sigma$

state transition
from q_1 to q_2

output string $u \in \Delta^*$

Mealy Machines

$\mathcal{A}$ is a Mealy Machine with an input/output alphabet Σ/Δ
i.e. a deterministic finite automaton with an output tape:



input $w \in \Sigma^*$

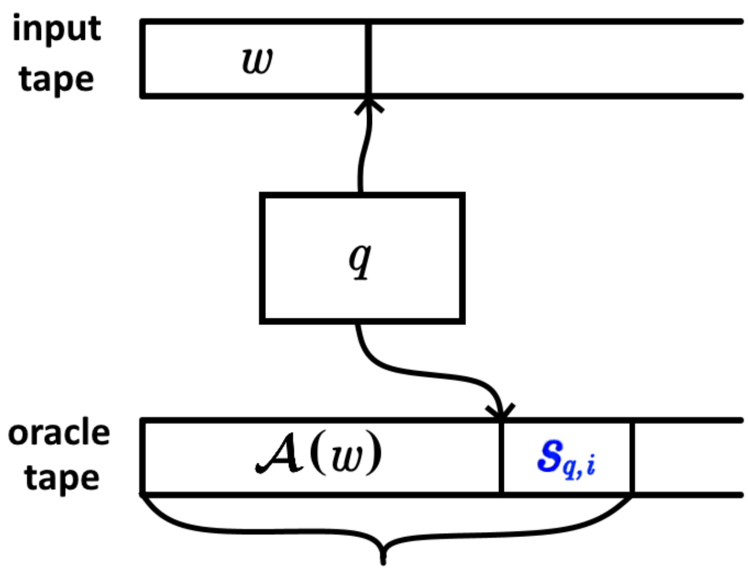
"final" state q

output $\mathcal{A}(w) \in \Delta^*$

$\longrightarrow$ a deterministic finite-state transducer: $w \in \Sigma^* \longmapsto \mathcal{A}(w) \in \Delta^*$

The Truth-Table Reduction by Oracle Mealy Machines

$\mathcal{A}^{L_2}$ is a Mealy Machine $\mathcal{A}$ with an oracle $L_2 \subseteq \Delta^*$:



for each state q of $\mathcal{A}$:

- r_q suffixes $s_{q,1}, \ldots, s_{q,r_q} \in \Delta^*$
- truth table $T_q : \{0,1\}^{r_q} \to \{0,1\}$ with r_q variables

r_q queries: $\overset{?}{\in} L_2$ for every $i = 1, \ldots, r_q$

$w \in \Sigma^*$ is accepted by $\mathcal{A}^{L_2}$ **iff** w brings $\mathcal{A}$ to the state q such that

$$T_q\left(\mathcal{A}(w) \cdot s_{q,1} \overset{?}{\in} L_2\,,\ \mathcal{A}(w) \cdot s_{q,2} \overset{?}{\in} L_2\,, \ldots,\ \mathcal{A}(w) \cdot s_{q,r_q} \overset{?}{\in} L_2\right) = 1$$

$L_1 \le_{tt}^{A} L_2$: $L_1 \subseteq \Sigma^*$ is truth-table reducible to $L_2 \subseteq \Delta^*$ **iff**
$L_1 = \mathcal{L}(\mathcal{A}^{L_2})$ is accepted by some Mealy machine $\mathcal{A}^{L_2}$ with oracle L_2

Proposition: *The relation* $\le_{tt}^{A}$ *is a preorder.*

Why $L_{\#} = \{0^n 1^n \mid n \geq 1\}$ Is the Simplest DCFL' language?

any reduced context-free grammar G generating a non-regular language $L \subseteq \Delta^*$

is self-embedding: there is a self-embedding nonterminal A admitting the derivation

$A \Rightarrow^* xAy$ for some non-empty strings $x, y \in \Delta^+$ (Chomsky, 1959)

G is reduced $\longrightarrow$ $S \Rightarrow^* vAz$ and $A \Rightarrow^* w$ for some $v, w, z \in \Delta^*$

$$\longrightarrow \qquad S \Rightarrow^* vx^m wy^m z \in L \text{ for every } m \geq 0 \qquad (1)$$

??? a conceivable (one-one) reduction from $L_{\#}$ to L: for every $m, n \geq 1$,

$$0^m 1^n \in \{0,1\}^* \longmapsto vx^m wy^n z \in \Delta^*$$

(the inputs outside 0^+1^+ are mapped onto some fixed string outside L)

since $0^m 1^n \in L_{\#}$ **implies** $vx^m wy^n z \in L$ by (1)

!!! however, the opposite implication may not be true:

Why $L_\#$ Is the Simplest DCFL' language? (cont.)

!!! however, the opposite implication may not be true:

for the DCFL' language $L_1 = \{a^m b^n \mid 1 \leq m \leq n\}$ over $\Delta = \{a, b\}$

there are **no** words $v, x, w, y, z \in \Delta^*$ such that for every $m, n \geq 1$,

$$vx^m wy^n z \in L_1 \text{ would ensure } m = n$$

nevertheless, already **two** inputs $a^m b^{n-1} \overset{?}{\in} L_1$ and $a^m b^n \overset{?}{\in} L_1$ decides $m \overset{?}{=} n$

$\longrightarrow$ the truth-table reduction from $L_\#$ to L_1 with two queries to the oracle L_1 :

$$0^m 1^n \in \{0,1\}^* \quad \longmapsto \quad vx^m wy^{n-1} z \in \Delta^*, \quad vx^m wy^n z \in \Delta^*$$

where $x = a$, $y = b$, $v = w = z = \varepsilon$ is the empty string

satisfying $0^m 1^n \in L_\#$ **iff** $(vx^m wy^{n-1} z \notin L_1$ **and** $vx^m wy^n z \in L_1)$

this can be generalized to any DCFL' language L :

The Main Technical Result

Theorem: *Let $L \subseteq \Delta^*$ be a non-regular deterministic context-free language over an alphabet Δ. There exist non-empty words $v, x, w, y, z \in \Delta^+$ and a language $L' \in \{L, \overline{L}\}$ (where $\overline{L} = \Delta^* \setminus L$ is the complement of L) such that*

1. **either** *for all* $m, n \geq 0$, $vx^m wy^n z \in L'$ **iff** $m = n$,
2. **or** *for all* $m, n \geq 0$, $vx^m wy^n z \in L'$ **iff** $m \leq n$.

1.

m \ n	0	1	2	3	…
0	$\in L'$	$\notin L'$	$\notin L'$	$\notin L'$	
1	$\notin L'$	$\in L'$	$\notin L'$	$\notin L'$	
2	$\notin L'$	$\notin L'$	$\in L'$	$\notin L'$	
3	$\notin L'$	$\notin L'$	$\notin L'$	$\in L'$	
⋮					⋱

2.

m \ n	0	1	2	3	…
0	$\in L'$	$\in L'$	$\in L'$	$\in L'$	
1	$\notin L'$	$\in L'$	$\in L'$	$\in L'$	
2	$\notin L'$	$\notin L'$	$\in L'$	$\in L'$	
3	$\notin L'$	$\notin L'$	$\notin L'$	$\in L'$	
⋮					⋱

In particular, for all $m \geq 0$ *and* $n > 0$,

$(vx^m wy^{n-1} z \notin L'$ **and** $vx^m wy^n z \in L')$ **iff** $m = n$.

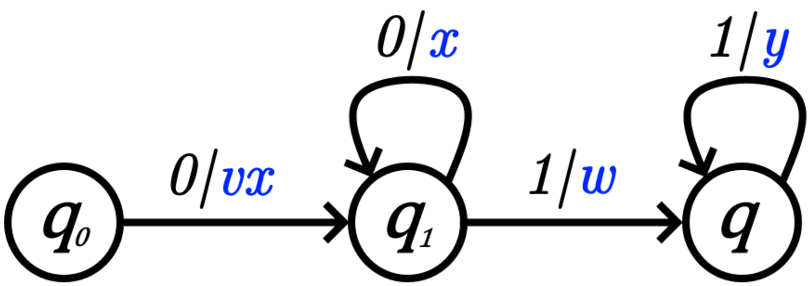
The Truth-Table Reduction From $L_\#$ to Any DCFL' L

implemented by a **Mealy machine** $\mathcal{A}^L$ with two queries to the oracle L:

For any DCFL' language $L \subseteq \Delta^*$, Theorem provides $v, x, w, y, z \in \Delta^+$ and $L' \in \{L, \overline{L}\}$, say $L' = L$ (analogously for $L' = \overline{L}$), such that

$$(\, vx^m wy^{n-1}z \notin L \;\; \textbf{and} \;\; vx^m wy^n z \in L \,) \;\; \textbf{iff} \;\; m = n\,. \qquad (2)$$

$\mathcal{A}^L$ transforms the input $0^m 1^n$ to the output $\mathcal{A}(0^m 1^n) = vx^m wy^{n-1} \in \Delta^+$ (the inputs outside 0^+1^+ are rejected), while moving to the state q with $r_q = 2$ suffixes $s_{q,1}$, $s_{q,2}$ and the truth table $T_q : \{0,1\}^2 \longrightarrow \{0,1\}$



$s_{q,1} = z$

$s_{q,2} = yz$

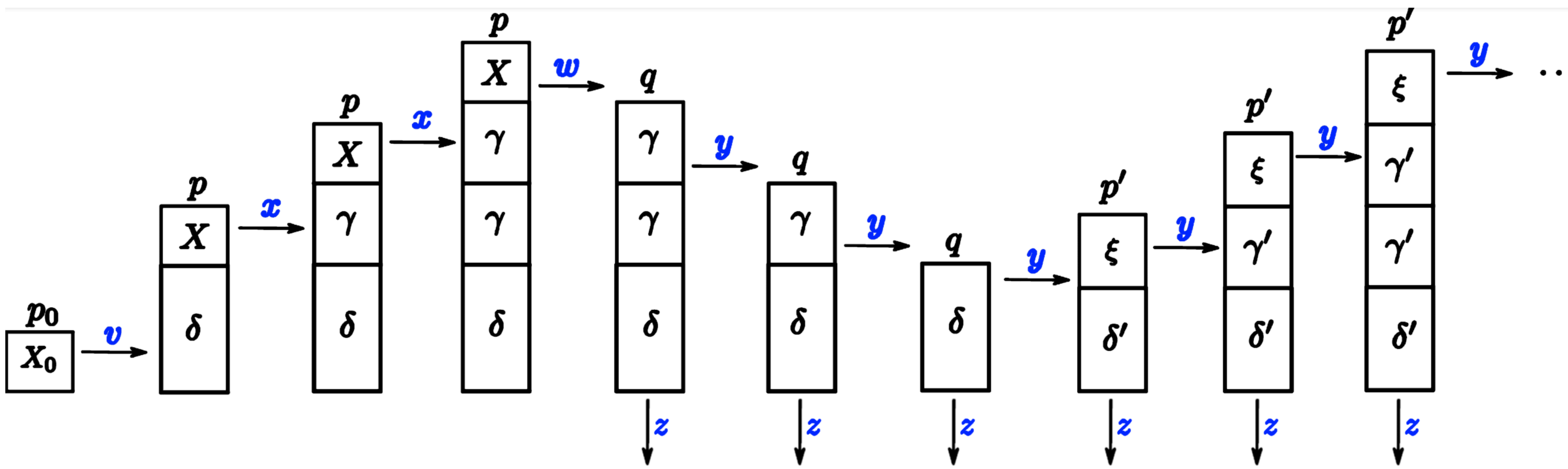
$\mathcal{A}(0^m1^n) \cdot z = vx^m wy^{n-1}z \overset{?}{\in} L$	$\mathcal{A}(0^m1^n) \cdot yz = vx^m wy^n z \overset{?}{\in} L$	T_q
0	1	1
0	0	0
1	1	0
1	0	0

It follows from (2) that $\mathcal{L}(\mathcal{A}^L) = L_\#$, i.e. $L_\# \leq_{tt}^{A} L$.

Ideas of the Proof of the Theorem

(inspired by some ideas on regularity of pushdown processes due to `Janar`, 2020)

- any non-regular DCFL language $L \subseteq \Delta^*$ is accepted
  by a deterministic pushdown automaton $\mathcal{M}$ by the empty stack
- since $L \notin$ REG, there is a computation by $\mathcal{M}$, reaching configurations with an arbitrary large stack which is being erased afterwards,
  corresponding to $v, x, w, y, z \in \Delta^+$ such that $vx^m wy^m z \in L$ for all $m \geq 1$
- in addition, we aim to ensure that for all $m \geq 0$ and $n > 0$,
  $(\, vx^m wy^{n-1} z \notin L'$ **and** $vx^m wy^n z \in L' \,)$ **iff** $m = n$



$(L' = L)$

case "$m = n$":	REJ	REJ	ACC	REJ	REJ	REJ
case "$m \leq n$":	REJ	REJ	ACC	ACC	ACC	ACC

Ideas of the Proof of the Theorem

(inspired by some ideas on regularity of pushdown processes due to Janar, 2020)

- any non-regular DCFL language $L \subseteq \Delta^*$ is accepted
  by a deterministic pushdown automaton $\mathcal{M}$ by the empty stack
- since $L \notin$ REG, there is a computation by $\mathcal{M}$, reaching configurations with an arbitrary large stack which is being erased afterwards, corresponding to $v, x, w, y, z \in \Delta^+$ such that $vx^m wy^m z \in L$ for all $m \geq 1$
- in addition, we aim to ensure that for all $m \geq 0$ and $n > 0$,
  $(vx^m wy^{n-1} z \notin L'$ **and** $vx^m wy^n z \in L')$ **iff** $m = n$
- we study the computation of $\mathcal{M}$ on an infinite word that traverses infinitely many pairwise non-equivalent configurations
- we use a natural congruence property of language equivalence on the set of configurations (determinism of $\mathcal{M}$ is essential)
- we apply Ramsey's theorem for extracting the required $v, x, w, y, z \in \Delta^+$ from the infinite computation

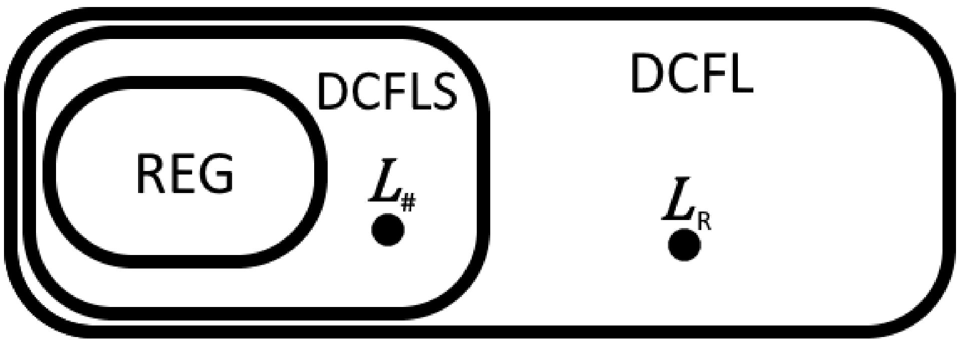
Basic Properties of DCFL'-Simple Problems

DCFLS is the class of DCFL'-simple problems

Proposition:

- REG $\subsetneq$ DCFLS $\subsetneq$ DCFL,

  e.g. $L_{\#} \in$ DCFLS, $L_R = \{wcw^R \mid w \in \{a,b\}^*\} \notin$ DCFLS



- The class DCFLS is closed under complement and intersection with regular languages.
- The class DCFLS is not closed under concatenation, intersection, and union.

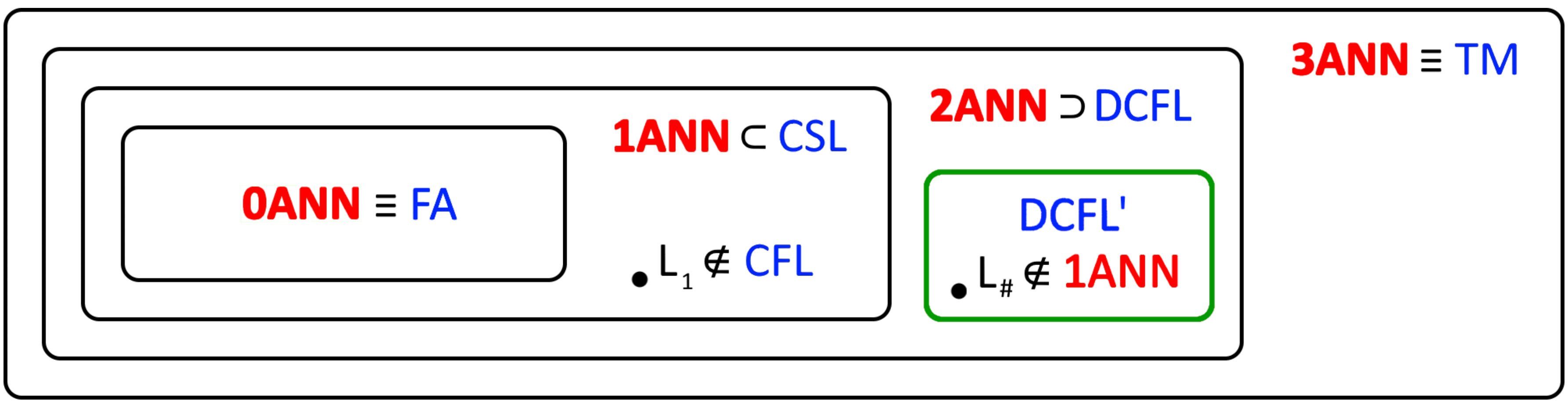
Application to the Analog Neuron Hierarchy

- $L_\# \notin$ **1ANN** by a nontrivial proof (based on the Bolzano–Weierstrass theorem) which can hardly be generalized to another DCFL' language

- $L_\#$ is DCFL'-simple under $\leq_{tt}^{A}$

- the reduction $\leq_{tt}^{A}$ to any $L \in$ 1ANN can be implemented by 1ANN

$\longrightarrow$ the known lower bound $L_\# \notin$ 1ANN for a single DCFL'-simple problem $L_\#$ is expanded to the whole class: **DCFL'** $\cap$ **1ANN** $= \emptyset$



$\longrightarrow$ **DCFL $\cap$ 1ANN = 0ANN**