Subrecursive Neural Networks

Jiří Šíma[a,*]

[a] *Institute of Computer Science of the Czech Academy of Sciences, P. O. Box 5, 18207 Prague 8, Czech Republic,*

Abstract

It has been known for discrete-time recurrent neural networks (NNs) that binary-state models using the Heaviside activation function (with Boolean outputs 0 or 1) are equivalent to finite automata (level 3 in the Chomsky hierarchy), while analog-state NNs with rational weights, employing the saturated-linear function (with real-number outputs in the interval $[0, 1]$), are Turing complete (Chomsky level 0) even for three analog units. However, it is as yet unknown whether there exist subrecursive (i.e. sub-Turing) NN models which occur on Chomsky levels 1 or 2. In this paper, we provide such a model which is a binary-state NN extended with one extra analog unit (1ANN). We achieve a syntactic characterization of languages that are accepted online by 1ANNs in terms of so-called cut languages which are combined in a certain way by usual operations. We employ this characterization for proving that languages accepted by 1ANNs with rational weights are context-sensitive (Chomsky level 1) and we present explicit examples of such languages that are not context-free (i.e. are above Chomsky level 2). In addition, we formulate a sufficient condition when a 1ANN recognizes a regular language (Chomsky level 3) in terms of quasi-periodicity of parameters derived from its real weights, which is satisfied e.g. for rational weights provided that the inverse of the real self-loop weight of the analog unit is a Pisot number.

Keywords: recurrent neural network, Chomsky hierarchy, cut language, quasi-periodic number

*Corresponding author

Email address: `sima@cs.cas.cz` (Jiří Šíma)

1. Introduction

The computational power of discrete-time recurrent neural networks (NNs) with the saturated-linear activation function[1] depends on the descriptive complexity of their weight parameters (Siegelmann, 1999; Šíma and Orponen, 2003). NNs with *integer* weights, corresponding to binary-state networks (with Boolean outputs 0 or 1), coincide with finite automata (Alon et al., 1991; Horne and Hush, 1996; Indyk, 1995; Minsky, 1967; Šíma, 2014a; Šíma and Wiedermann, 1998). *Rational* weights make the analog-state NNs (with real-number outputs in the interval $[0, 1]$) computationally equivalent to Turing machines (Indyk, 1995; Siegelmann and Sontag, 1995), and thus (by a real-time simulation due to Siegelmann and Sontag, 1995) polynomial-time computations of such networks are characterized by the complexity class P. Moreover, NNs with arbitrary *real* weights can even derive "super-Turing" computational capabilities (Siegelmann, 1999). In particular, their polynomial-time computations correspond to the nonuniform complexity class P/poly while any input/output mapping (including undecidable problems) can be computed within exponential time (Siegelmann and Sontag, 1994). In addition, a proper hierarchy of nonuniform complexity classes between P and P/poly has been established for polynomial-time computations of NNs with increasing Kolmogorov complexity of real weights (Balcázar et al., 1997).

As can be seen, our understanding of the computational power of super-recursive (i.e. super-Turing) NNs is satisfactorily fine-grained when changing from rational to arbitrary real weights. In contrast, there is still a gap between integer and rational weights which results in a jump from regular languages capturing the lowest level 3 in the Chomsky hierarchy to recursively enumerable languages on the highest Chomsky level 0. However, it is as yet unknown whether there exist so-called *subrecursive* (i.e. sub-Turing)[2] NNs that occur on Chomsky levels 1 or 2. In this paper, we provide such a model.

In order to refine the classification of subrecursive NNs which do not possess the full power of Turing machines (Chomsky level 0), we initiate the

[1] The results are partially valid for more general classes of activation functions (Koiran, 1996; Siegelmann, 1996; Šíma, 1997; Šorel and Šíma, 2004) including the logistic function (Kilian and Siegelmann, 1996).

[2] We use the term "subrecursive" to refer to recursion (computability) theory where a function is called *recursive* if it is computable by a Turing machine. Thus, by subrecursive neural networks we mean any class of NNs that are computationally less powerful than Turing machines.

study of binary-state NNs extended with a few extra analog neurons having real weights. It appears that already a binary-state NN to which only *three* analog-state units with rational weights are added, is Turing complete (Chomsky level 0), since it can implement two stacks of pushdown automata, a model equivalent to Turing machines (Šíma, 2018). Therefore, we consider a model of binary-state NNs including exactly *one* extra analog neuron with real weights. Such a model with an online input/output protocol has been shown to be computationally equivalent to so-called finite automata with a register whose domain is partitioned into a finite number of intervals, each associated with a local state-transition function (Šíma, 2014b). Although this study has been inspired by theoretical issues, NNs with different types of units/layers are widely used in practical applications, e.g. in deep learning (Schmidhuber, 2015), and they thus require a detailed mathematical analysis.

In this paper, we characterize syntactically the class of languages that are accepted online by binary-state neural networks with an extra analog unit (1ANN) in terms of so-called *cut languages* (Šíma and Savický, 2018) which are combined in a certain way by usual operations such as complementation, intersection, union, concatenation, Kleene star, reversal, the largest prefix-closed subset, and a letter-to-letter morphism. A cut language $L_{<c}$ contains finite representations of numbers in a real *base* β (so-called *β-expansions*) using real *digits* from a finite alphabet A, that are less than a given real *threshold* c (i.e. a Dedekind cut).

We have classified the class of cut languages within the Chomsky hierarchy (Šíma and Savický, 2018). For this purpose, we have introduced a new concept of a *quasi-periodic number* having all of its infinite β-expansions eventually quasi-periodic. In particular, we have proven that a cut language $L_{<c}$ is regular iff c is a quasi-periodic number, while $L_{<c}$ is not even context-free if it is not the case. Nevertheless, any cut language $L_{<c}$ with rational parameters (threshold, base, and digits) is context-sensitive. These results have revealed an interesting connection between the analysis of NN models and an active number-theoretic research field of non-standard numeral systems employing non-integer bases (e.g. Baker, 2014; Baker et al., 2017; de Vries and Komornik, 2009; Dombek et al., 2011; Erdös et al., 1990; Frougny and Lai, 2011; Frougny et al., 2014; Glendinning and Sidorov, 2001; Hare, 2007; Komornik et al., 2011; Komornik and Pedicini, 2017; Kong et al., 2010; Schmidt, 1980; Sidorov, 2003), which was established already in late 1950's (Parry, 1960; Rényi, 1957).

By using the syntactic characterization of 1ANNs achieved in this paper, we derive a sufficient condition when a 1ANN recognizes a regular language (Chomsky level 3), in terms of quasi-periodicity of some parameters depending on its real weights. For example, a 1ANN with weights from the smallest field extension[3] $\mathbb{Q}(\beta)$ over the rational numbers including a real number $\beta \in \mathbb{R}$, such that the self-loop weight w of its analog unit is $1/\beta$ for some Pisot base $\beta > 1$, is computationally equivalent to a finite automaton. For instance, since every integer $n > 1$ is a Pisot number (in fact, such integers are the only rational Pisot numbers), it follows that any 1ANN with rational weights such that $w = 1/n$, accepts a regular language.

Furthermore, we show examples of languages accepted by 1ANNs with rational weights that are not context-free (i.e. are above Chomsky level 2) while we prove that any language accepted by this model is context-sensitive (Chomsky level 1). Thus, these results refine the analysis of the computational power of subrecursive NNs with the weight parameters between integer and rational weights. Namely, the computational power of binary-state networks having integer weights can increase from regular languages (Chomsky level 3) to that between context-free (Chomsky level 2) and context-sensitive languages (Chomsky level 1), when an extra analog unit with rational weights is added, while a condition when this does not bring any additional power even for real weights, is formulated.

The paper is organized as follows. In Section 2, we introduce basic definitions concerning the language acceptors based on 1ANNs. In Section 3, we give a brief review of definitions and results related to quasi-periodic numbers and cut languages, including unique and eventually periodic β-expansions, examples of quasi-periodic numbers, and the classification of cut languages within the Chomsky hierarchy, which will later be used for classifying the 1ANN models. The main technical results of the paper is the representation theorem providing a syntactic characterization of languages accepted by 1ANNs, which is proven in Section 4 and partially reversed in Section 5. As a consequence of this characterization, in Section 6, we formulate a sufficient condition when a language accepted by 1ANN is regular. In addition, we

[3] Recall that in algebra, the rational numbers (fractions) form the field $\mathbb{Q}$ with the two usual operations, the addition and the multiplication over real numbers. For any real number $\beta \in \mathbb{R}$, the field extensions $\mathbb{Q}(\beta) \subset \mathbb{R}$ is the smallest set containing $\mathbb{Q} \cup \{\beta\}$ that is closed under these operations. For example, the golden ratio $\varphi = (1 + \sqrt{5})/2 \in \mathbb{Q}(\sqrt{5})$ whereas $\sqrt{2} \notin \mathbb{Q}(\sqrt{5})$. Note that $\mathbb{Q}(\beta) = \mathbb{Q}$ for every $\beta \in \mathbb{Q}$.

show a lower bound on the computational power of 1ANNs by providing an explicit example of non-context-free languages that are recognized by 1ANNs with rational weights, while any language accepted by this model proves to be context-sensitive, which represents a corresponding upper bound. Finally, we summarize the results and present some open problems in Section 7.

A preliminary version of this paper (Šíma, 2017) considered the model of 1ANNs restricted only to rational weights and exploited the equivalence of 1ANNs and finite automata with a register, while the present proofs use direct constructions of 1ANNs with real weights.

2. Neural Language Acceptors With an Extra Analog Unit

We specify a computational model of a *binary-state neural network with an extra analog unit* (shortly, 1ANN), $\mathcal{N}$, which will be used online as a formal language acceptor. The network $\mathcal{N}$ consists of $s \geq 3$ *units (neurons)*, indexed as $V = \{1, \ldots, s\}$. All the units in $\mathcal{N}$ are assumed to be binary-state (shortly *binary*) neurons (i.e. *perceptrons, threshold gates)* except for the last sth neuron which is an analog-state (shortly *analog*) unit. The neurons are connected into a directed graph representing an *architecture* of $\mathcal{N}$, in which each edge $(i, j) \in V^2$ leading from unit i to j is labeled with a real *weight* $w(i, j) = w_{ji} \in \mathbb{R}$. The absence of a connection within the architecture corresponds to a zero weight between the respective neurons, and vice versa.

The *computational dynamics* of $\mathcal{N}$ determines for each unit $j \in V$ its *state (output)* $y_j^{(t)}$ at discrete time instants $t = 0, 1, 2, \ldots$. The states $y_j^{(t)}$ of the first $s-1$ binary neurons $j \in V \setminus \{s\}$ are Boolean values 0 or 1, whereas the output $y_s^{(t)}$ from analog unit s is a real number from the unit interval $\mathbb{I} = [0, 1]$. This establishes the *network state* $\mathbf{y}^{(t)} = \left(y_1^{(t)}, \ldots, y_{s-1}^{(t)}, y_s^{(t)}\right) \in \{0, 1\}^{s-1} \times \mathbb{I}$ at each discrete time instant $t \geq 0$.

At the beginning of a computation, the neural network $\mathcal{N}$ is placed in an *initial state* $\mathbf{y}^{(0)} \in \{0, 1\}^s$ which may also include an external input, where, for simplicity, we assume $y_s^{(0)} = 0$. At discrete time instant $t \geq 0$, an *excitation* of any neuron $j \in V$ is defined as

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

including a real *bias* value $w_{j0} \in \mathbb{R}$ which can be viewed as the weight $w(0, j) = w_{j0}$ from a formal constant unit input $y_0^{(t)} \equiv 1$ (i.e. $0 \in V$). At the next instant $t+1$, the neurons $j \in V_{t+1}$ from a selected subset $V_{t+1} \subseteq V$ compute their new outputs $y_j^{(t+1)}$ in parallel by applying an *activation function* $\sigma_j : \mathbb{R} \longrightarrow \mathbb{I}$ to $\xi_j^{(t)}$, whereas the remaining units $j \notin V_{t+1}$ do not update their states, that is,

$$y_j^{(t+1)} = \begin{cases} \sigma_j\left(\xi_j^{(t)}\right) & \text{for } j \in V_{t+1} \\ y_j^{(t)} & \text{for } j \in V \setminus V_{t+1}\,. \end{cases} \tag{2}$$

For perceptron units $j \in V \setminus \{s\}$ with binary states $y_j \in \{0,1\}$, the *Heaviside* activation function $\sigma_j(\xi) = H(\xi)$ is used where

$$H(\xi) = \begin{cases} 1 & \text{for } \xi \geq 0 \\ 0 & \text{for } \xi < 0\,, \end{cases} \tag{3}$$

while the analog-state unit $s \in V$ employs the *saturated-linear* function $\sigma_s(\xi) = \sigma(\xi)$ where

$$\sigma(\xi) = \begin{cases} 1 & \text{for } \xi \geq 1 \\ \xi & \text{for } 0 < \xi < 1 \\ 0 & \text{for } \xi \leq 0\,. \end{cases} \tag{4}$$

In this way, the new network state $\mathbf{y}^{(t+1)} \in \{0,1\}^{s-1} \times \mathbb{I}$ at time $t+1$ is determined.

Without loss of efficiency (Orponen, 1997), we assume synchronous computations for which the sets V_t that define the computational dynamics of $\mathcal{N}$ according to (2), are predestined deterministically. Usually, the sets V_t correspond to layers in the architecture of $\mathcal{N}$ which are updated one by one (e.g., a feedforward subnetwork). In particular, we use a systematic periodic choice of V_t so that $V_{t+d} = V_t$ for every $t \geq 1$ where an integer parameter $d \geq 1$ represents the number of updates within one *macroscopic time* step (e.g., d is the number of layers). For notational simplicity, we assume that the state of the analog unit $s \in V$ is updated exactly once in every macroscopic time step, say $s \in V_{d\tau}$ for every $\tau \geq 1$.

The computational power of NNs has been studied analogously to the traditional models of computations so that the networks are exploited as acceptors of formal languages $L \subseteq \Sigma^*$ over a finite alphabet $\Sigma = \{\lambda_1, \ldots \lambda_q\}$ of q letters (symbols). For the finite networks the following online input/output protocol has been used (Alon et al., 1991; Horne and Hush, 1996; Indyk, 1995; Siegelmann, 1996; Šíma and Orponen, 2003; Šíma and Wiedermann, 1998). An input word (string) $\mathbf{x} = x_1 \ldots x_n \in \Sigma^n$ of arbitrary length $n \geq 0$ is sequentially presented to the network, symbol after symbol, via the first $q < s$ so-called *input neurons* $X = \{1, \ldots, q\} \subset V \setminus \{s\}$. We employ the popular *one-hot encoding* of alphabet Σ. This means that each letter $\lambda_i \in \Sigma$ is represented by one input neuron $i \in X$ which is activated when symbol λ_i is being read. Thus, the states of input neurons X are externally set (and clamped) at macroscopic time steps so that they represent a current input symbol regardless of any influence from the remaining neurons in the network, that is,

$$y_i^{(\tau d+k)} = \begin{cases} 1 & \text{if } x_{\tau+1} = \lambda_i \\ 0 & \text{otherwise} \end{cases} \qquad \text{for} \quad i \in X, \qquad \begin{array}{l} \tau = 0, \ldots, n-1, \\ k = 0 \ldots, d-1, \end{array} \tag{5}$$

where an integer $d \geq 1$ is the *time overhead* for processing a single input symbol which coincides with the macroscopic time step. Then, the so-called *output neuron* $\text{out} \in V \setminus (X \cup \{s\})$ signals at the macroscopic time instant $\tau = n$ whether the input word belongs to the underlying language L, that is,

$$y_{\text{out}}^{(nd)} = \begin{cases} 1 & \text{for } \mathbf{x} \in L \\ 0 & \text{for } \mathbf{x} \notin L\,. \end{cases} \tag{6}$$

Note that the states $\mathbf{y}_X^{(nd)} = \left(y_1^{(nd)}, \ldots, y_q^{(nd)}\right) \in \{0,1\}^q$ of input neurons X at the macroscopic time instant $\tau = n$ when the network output (6) is produced, are not specified by (5), since they do not influence this output. We thus define them formally so that they represent any symbol $x_{n+1} \in \Sigma$ added to the input string in order to ensure the formal correctness of dynamics equations (2). We say that a language $L \subseteq \Sigma^*$ is *accepted (recognized)* by a 1ANN $\mathcal{N}$, which is denoted by $L = \mathcal{L}(\mathcal{N})$, if for any input word $\mathbf{x} \in \Sigma^*$, $\mathbf{x}$ is accepted by $\mathcal{N}$ iff $\mathbf{x} \in L$.

Example 1 We illustrate the definition of the 1ANN language acceptor and its input/output protocol on a simple 1ANN $\mathcal{N}$, which will further be used in this paper as a running example for clarifying the theorems. The network $\mathcal{N}$ is composed of only $s = 5$ neurons, that is, $V = \{0, 1, \ldots, 5\}$ where the last 5-th unit is the analog neuron. The architecture of $\mathcal{N}$ is depicted in Figure 1 where the directed edges connecting units are labeled with the respective weights $w_{34} = -1$, $w_{45} = 1$, $w_{52} = \frac{2}{3}$, and $w_{55} = \frac{1}{3}$, while the edge

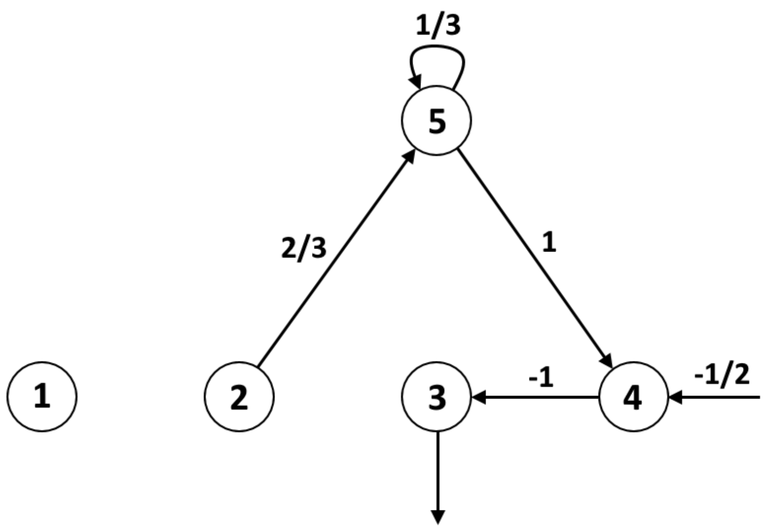

Figure 1: Example of a 1ANN language acceptor.

drawn without the originating unit $0 \in V$ corresponds to the bias parameter $w_{40} = -\frac{1}{2}$.

The 1ANN $\mathcal{N}$ is employed for recognizing a language $L = \mathcal{L}(\mathcal{N})$ over the binary alphabet $\Sigma = \{\lambda_1, \lambda_2\} = \{0, 1\}$ including $q = |\Sigma| = 2$ symbols, $\lambda_1 = 0$ and $\lambda_2 = 1$. For this purpose, the first two units serve as the input neurons $X = \{1, 2\} \subset V$ using the one-hot encoding of Σ, whereas the third unit is the output neuron out $= 3 \in V \setminus (X \cup \{s\})$, which implements the input/output protocol (5) and (6). Notice that the first neuron $1 \in V$ corresponding to the input symbol $\lambda_1 = 0$, is here in fact disconnected from the network, since in the one-hot encoding, its role is taken by the second unit $2 \in V$ corresponding to $\lambda_2 = 1$, whose state is 0 iff the first neuron outputs 1. In addition, the non-input neurons start at the initial states $y_3^{(0)} = 1$, $y_4^{(0)} = y_5^{(0)} = 0$, and we assume that one macroscopic step of $\mathcal{N}$ consists of $d = 2$ updates which corresponds to the time overhead for processing a single input symbol. Namely, $V_1 = \{4\}$, $V_2 = \{3, 5\}$, and the set of neurons V_t that are updated at time instant $t \geq 1$, satisfy $V_t = V_{t+2}$ for every $t \geq 1$.

For instance, suppose that the input word $\mathbf{x} = 1100 \in \{0, 1\}^4$ of length $n = 4$ is externally presented to $\mathcal{N}$ where $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$, and let e.g. $x_5 = 0$. Table 1 shows the sequential schedule of presenting

τ	$x_{\tau+1}$	t	$y_1^{(t)}$	$y_2^{(t)}$	$y_3^{(t)}$	$y_4^{(t)}$	$y_5^{(t)}$	the result of recognition
0	1	0	$\mathbf{0}$	$\mathbf{1}$	1	0	0	$\varepsilon \in \mathcal{L}(\mathcal{N})$
0	1	1	0	1	1	$\mathbf{0}$	0	
1	1	2	$\mathbf{0}$	$\mathbf{1}$	$\mathbf{1}$	0	$\mathbf{\frac{2}{3}}$	$1 \in \mathcal{L}(\mathcal{N})$
1	1	3	0	1	1	$\mathbf{1}$	$\frac{2}{3}$	
2	0	4	$\mathbf{1}$	$\mathbf{0}$	$\mathbf{0}$	1	$\mathbf{\frac{8}{9}}$	$11 \notin \mathcal{L}(\mathcal{N})$
2	0	5	1	0	0	$\mathbf{1}$	$\frac{8}{9}$	
3	0	6	$\mathbf{1}$	$\mathbf{0}$	$\mathbf{0}$	1	$\mathbf{\frac{8}{27}}$	$110 \notin \mathcal{L}(\mathcal{N})$
3	0	7	1	0	0	$\mathbf{0}$	$\frac{8}{27}$	
4	0	8	$\mathbf{1}$	$\mathbf{0}$	$\mathbf{1}$	0	$\mathbf{\frac{8}{81}}$	$1100 \in \mathcal{L}(\mathcal{N})$

Table 1: The accepting computation by the 1ANN $\mathcal{N}$ from Figure 1 on the input 1100.

the symbols x_1, x_2, x_3, x_4 of $\mathbf{x}$ to $\mathcal{N}$ through the input neurons X at the macroscopic time steps $\tau = 0, 1, 2, 3$, respectively. Namely, the state values $y_1^{(2\tau)} = 1$ and $y_2^{(2\tau)} = 0$ encode the input symbol $x_{\tau+1} = 0$ whereas $y_1^{(2\tau)} = 0$ and $y_2^{(2\tau)} = 1$ translates to $x_{\tau+1} = 1$ in the one-hot encoding, according to (5). The computation by $\mathcal{N}$ on the input 1100 produces a sequence of network states which are listed in Table 1 where the updated states of neurons in V_t, including the externally set inputs, are indicated in boldface. This includes the result of the recognition which is reported by the output neuron out $= 3$, even for each of the five prefixes of $\mathbf{x}$, the empty string ε, 1, 11, 110 and 1100, at the respective macroscopic time steps $\tau = 0, 1, 2, 3, 4$, according to (6).

3. Quasi-Periodic Numbers and Cut Languages

In this section, we recall the notions of quasi-periodic numbers and cut languages, which have been introduced by the author (Šíma and Savický, 2018) within the context of 1ANN analysis. Nevertheless, these concepts also appear to be interesting in numeral systems within algebraic number theory (see the references throughout this section). The following detailed exposition is thus motivated by the fact that the positional numeral systems provide a connection between the world of NNs and the world of formal languages. Namely, the idea of periodicity in the non-standard numeral systems with non-integer bases appears to be related to regular cut languages (see Theorem 2 below). The bridge between these two worlds is then built in the representation theorem in Section 4, providing a syntactic characterization of languages accepted by 1ANNs in terms of cut languages.

The section is organized as follows. In Paragraph 3.1, we first review the definitions and results concerning β-expansions. In Paragraph 3.2, we introduce and illustrate by examples the concept of quasi-periodic numbers which is then used for classifying the cut languages within the Chomsky hierarchy in Paragraph 3.3.

3.1. β-Expansions: Uniqueness and Periodicity

Hereafter, let β be a real number such that $|\beta| > 1$, which represents a *base (radix)* of non-standard positional numeral system, and let $A \neq \emptyset$ be a finite set of real *digits*. We say that a word (string) $\mathbf{a} = a_1 \ldots a_n \in A^*$ over alphabet A is a *finite β-expansion* of a real number x if

$$x = (\mathbf{a})_\beta = (a_1 \ldots a_n)_\beta = \sum_{k=1}^{n} a_k \beta^{-k} \,. \tag{7}$$

Note that we use only negative powers of β while omitting the radix point at the left of β-expansions. Moreover, $\mathbf{a} = a_1 a_2 a_3 \ldots \in A^\omega$ is an (*infinite*) β-expansion of a real number x, where A^ω denotes the set of all countably infinite words over A, if

$$x = (\mathbf{a})_\beta = (a_1 a_2 a_3 \ldots)_\beta = \sum_{k=1}^{\infty} a_k \beta^{-k} \,. \tag{8}$$

Observe that the infinite sum in (8) can be viewed as a power series in variable β^{-1} which is convergent due to $|\beta| > 1$.

Obviously, β-expansions are a generalization of the integer-base numeral representations such as the classical *decimal* expansions in base $\beta = 10$ with the digits from $A = \{0, 1, \ldots, 9\}$ or the *binary* expansions in base $\beta = 2$ with the digit alphabet $A = \{0, 1\}$. For instance, $(75)_{10} = 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$ and $(11)_2 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$ are the finite β-expansions of $\frac{3}{4}$ for these two integer bases, respectively, while the same number can be represented by the finite β-expansion $\frac{3}{4} = \frac{7}{4} \cdot \left(\frac{5}{2}\right)^{-1} + \frac{5}{16} \cdot \left(\frac{5}{2}\right)^{-2}$ using the non-integer base $\beta = \frac{5}{2}$ and alphabet $A = \left\{\frac{5}{16}, \frac{7}{4}\right\}$. The representations in non-integer bases have systematically been studied since late 1950's, starting with the seminal papers due to Rényi (1957) and Parry (1960), and nowadays, they are still the subject of active research with important applications. For example, recent results on non-integer numeral systems find their applications in the algorithms for fast parallel addition (Frougny et al., 2014).

Furthermore, an infinite β-expansion $\mathbf{a} \in A^\omega$ is called *eventually periodic* if $\mathbf{a} = a_1 a_2 \dots a_{k_1} (a_{k_1+1} a_{k_1+2} \dots a_{k_2})^\omega$ where $m = k_2 - k_1 > 0$ is the length of a *repetend* (repeating digits) $a_{k_1+1} a_{k_1+2} \dots a_{k_2} \in A^m$, while k_1 is the length of *preperiodic part* $a_1 a_2 \dots a_{k_1} \in A^{k_1}$. For $k_1 = 0$, we call such a β-expansion (purely) *periodic*. Any eventually periodic β-expansion can be evaluated as

$$(a_1 a_2 \dots a_{k_1} (a_{k_1+1} a_{k_1+2} \dots a_{k_2})^\omega)_\beta = (a_1 \dots a_{k_1})_\beta + \beta^{-k_1} \varrho \tag{9}$$

where a so-called *periodic point* $\varrho = ((a_{k_1+1} a_{k_1+2} \dots a_{k_2})^\omega)_\beta \in \mathbb{R}$ satisfies

$$(a_{k_1+1} a_{k_1+2} \dots a_{k_2})_\beta = \sum_{k=1}^{m} a_{k_1+k} \beta^{-k} = \varrho \left(1 - \beta^{-m}\right) \tag{10}$$

by the sum of geometric series with the common ratio β^{-m}.

For simplicity, it is usually assumed in the literature that a real base meets $\beta > 1$ and the standard digits are integers from $A = \{0, 1, \dots, \lceil \beta \rceil - 1\}$, which ensures that a β-expansion exists for every real number $x \in D_\beta$ where D_β is the interval $D_\beta = [0\,,\, (\lceil \beta \rceil - 1)\,/(\beta - 1)]$. For an integer base $\beta \in \mathbb{N}$ when $D_\beta = \mathbb{I}$, it is well known that every irrational number $x \in D_\beta \setminus \mathbb{Q}$ has a *unique* infinite β-expansion, while any rational $x \in D_\beta \cap \mathbb{Q}$ has either a *unique* eventually periodic β-expansion (e.g., the endpoints 0 and 1 of D_β have the trivial β-expansions 0^ω and $(\beta - 1)^\omega$, respectively), or exactly the *two* distinct eventually periodic β-expansions, $a_1 a_2 \dots a_n 0^\omega$ and $a_1 a_2 \dots a_{n-1} (a_n - 1)(\beta - 1)^\omega$, if there exists a finite β-expansion $a_1 a_2 \dots a_n \in A^*$ of $x = (a_1 a_2 \dots a_n)_\beta$ such that $a_n \neq 0$. An example of such an ambiguity is $\frac{3}{4} = (75)_{10} = (750^\omega)_{10} = (749^\omega)_{10}$.

For a non-integer base β, by contrast, *almost every* number $x \in D_\beta$ has a continuum of distinct β-expansions (Sidorov, 2003). In particular, for $1 < \beta < 2$ when $A = \{0, 1\}$, *every* number $x \in D_\beta = [0, 1/(\beta - 1)]$ except for the endpoints 0 and $1/(\beta - 1)$, has a continuum of distinct β-expansions if $1 < \beta < \varphi$ where $\varphi = (1 + \sqrt{5})/2 \approx 1.618034$ is the golden ratio (Erdös et al., 1990). On the other hand, for $\varphi \leq \beta < q$ where $q \approx 1.787232$ is the (transcendental) *Komornik-Loreti constant* (i.e. the unique solution of equation $\sum_{k=1}^{\infty} t_k q^{-k} = 1$ where $(t_k)_{k=1}^{\infty}$ is the *Thue-Morse sequence* in which

$t_k \in \{0,1\}$ is the parity of the number of 1s in the binary representation of k), there are *countably* many numbers in D_β that have *unique* β-expansions which are eventually periodic (Glendinning and Sidorov, 2001). Moreover, for $q \leq \beta < 2$, there is a *continuum* of numbers in D_β with unique β-expansions.

These results have further been generalized to any non-integer base $\beta > 2$ combined with the standard integer digits from $A = \{0, 1, \ldots, \lceil\beta\rceil - 1\}$, by using generalized golden ratios (Baker, 2014) and generalized Komornik-Loreti constants (de Vries and Komornik, 2009; Kong et al., 2010). For a general reasonable alphabet A (when D_β is an interval) that may include non-integer digits, which is the case considered in this paper, there exist two critical bases φ_A and q_A such that $1 < \varphi_A \leq q_A$ and the number of unique β-expansions is finite if $1 < \beta < \varphi_A$, countable if $\varphi_A < \beta < q_A$, and uncountable if $\beta > q_A$. Nevertheless, the determination of these critical bases (and even the existence of q_A) for arbitrary A is still not complete even for three digits (Komornik et al., 2011; Komornik and Pedicini, 2017).

The β-expansions of a given number can be ordered lexicographically and its maximal (resp. minimal) β-expansion is called *greedy* (resp. *lazy*). Obviously, any unique β-expansion is simultaneously greedy and lazy. For simplicity, assume $\beta > 1$ and $A = \{0, 1, \ldots, \lceil\beta\rceil - 1\}$, and denote by $\mathrm{Per}(\beta)$ the set of numbers whose greedy β-expansion using the digits from A, is eventually periodic. For any integer base $\beta \in \mathbb{Z}$, it is well known that $\mathrm{Per}(\beta) = \mathbb{Q} \cap \mathbb{I}$. For a non-integer base β, we have $\mathrm{Per}(\beta) \subseteq \mathbb{Q}(\beta) \cap D_\beta$ where $\mathbb{Q}(\beta)$ denotes the smallest field extension[3] over $\mathbb{Q}$ including β, according to (9) and (10).

On the other hand, if $\mathbb{Q} \cap \mathbb{I} \subset \mathrm{Per}(\beta)$, then β must be either a Pisot or a Salem number (Schmidt, 1980) where a *Pisot* number is a real algebraic integer (a root of some monic polynomial with integer coefficients) greater than 1 such that all its Galois conjugates (other roots of such a unique monic polynomial with minimal degree) are in absolute value less than 1. A characteristic property of Pisot numbers is that their powers approach integers at an exponential rate. Similarly, a *Salem* number is a real algebraic integer greater than 1 such that all its Galois conjugates are in absolute value less or equal to 1 and at least one equals 1. In particular, for any $\beta \in \mathbb{Q} \setminus \mathbb{Z}$ which cannot be a Pisot nor Salem number by the integral root theorem, there exists a rational number from $D_\beta \cap \mathbb{Q}$ whose greedy β-expansion is not eventually periodic. Nevertheless, it was shown for any Pisot number β that $\mathrm{Per}(\beta) = \mathbb{Q}(\beta) \cap D_\beta$ (Schmidt, 1980), while for Salem numbers this is still an open problem (Hare, 2007). Recently, these results have partially been generalized to negative base $\beta < -1$ and non-integer digits in A (Baker et al., 2017; Dombek et al., 2011; Frougny and Lai, 2011).

3.2. Quasi-Periodic Numbers

We have generalized the notion of eventual periodicity defined by (9) and (10) (Šíma and Savický, 2018). We say that an infinite β-expansion $a_1a_2a_3\ldots \in A^\omega$ is *eventually quasi-periodic* with a *periodic point* $\varrho \in \mathbb{R}$ if there is an increasing infinite sequence of indices, $0 \leq k_1 < k_2 < \cdots$, such that for every $i \geq 1$,

$$(a_{k_i+1}\ldots a_{k_{i+1}})_\beta = \sum_{k=1}^{m_i} a_{k_i+k}\beta^{-k} = \varrho\left(1-\beta^{-m_i}\right) \tag{11}$$

(cf. (10)) where $m_i = k_{i+1} - k_i > 0$ is the length of *quasi-repetend* $a_{k_i+1}\ldots a_{k_{i+1}} \in A^{m_i}$, while k_1 is the length of *preperiodic part* $a_1\ldots a_{k_1} \in A^{k_1}$. For $k_1 = 0$, we call such a β-expansion (purely) *quasi-periodic*. Any eventually quasi-periodic β-expansion can be evaluated as

$$(a_1a_2a_3\ldots)_\beta = \sum_{k=1}^{\infty} a_k\beta^{-k} = (a_1\ldots a_{k_1})_\beta + \beta^{-k_1}\varrho \tag{12}$$

(cf. (9)) by using (11) since

$$\begin{aligned}
\sum_{k=k_1+1}^{\infty} a_k\beta^{-k} &= \sum_{i=1}^{\infty}\beta^{-k_i}\sum_{k=1}^{m_i} a_{k_i+k}\beta^{-k} = \varrho\sum_{i=1}^{\infty}\beta^{-k_i}\left(1-\beta^{-m_i}\right) \\
&= \varrho\sum_{i=1}^{\infty}\left(\beta^{-k_i}-\beta^{-k_{i+1}}\right) \\
&= \varrho\left(\beta^{-k_1}+\sum_{i=2}^{\infty}\beta^{-k_i}-\sum_{i=1}^{\infty}\beta^{-k_{i+1}}\right) = \beta^{-k_1}\varrho
\end{aligned} \tag{13}$$

is an absolutely convergent series. In fact, condition (11) is equivalent to the statement that every quasi-repetend creates a periodic β-expansion of ϱ, that is, for every $i \geq 1$,

$$((a_{k_i+1}\ldots a_{k_{i+1}})^\omega)_\beta = \varrho\,. \tag{14}$$

It follows from (12) that the preperiodic part together with an arbitrary sequence of quasi-repetends satisfying (11) respectively (14) for the same periodic point ϱ, yields a β-expansion of the same number. The quasi-repetends in a single eventually quasi-periodic β-expansion can be distinct having different length which can even be unbounded (see Example 3). Clearly, every eventually periodic β-expansion is eventually quasi-periodic with a sequence of identical quasi-repetends. For illustration, we present a simple example of eventually quasi-periodic β-expansions:

Example 2 (Šíma and Savický, 2018, Example 1) Assume $A = \{0, 1\}$ and let $\beta \approx 1.220744$ be the real root of the polynomial $x^4 - x - 1$, which means

$$\beta^4 - \beta - 1 = 0\,, \tag{15}$$

such that $1 < \beta < 2$. Any infinite word $\mathbf{a} \in A^\omega$ generated by the ω-regular expression[4] $00\,(010 + 1000)^\omega$ is an eventually quasi-periodic β-expansion of the number 1 with the periodic point $\varrho = \beta^2$. In particular, the prefix 00 is the preperiodic part of length $k_1 = 2$, while 010 and 1000 represent two quasi-repetends of length 3 and 4, respectively, satisfying condition (11):

$$\begin{aligned} (010)_\beta &= \beta^{-2} = \beta^2\left(1 - \beta^{-3}\right) && (16)\\ (1000)_\beta &= \beta^{-1} = \beta^2\left(1 - \beta^{-4}\right) && (17) \end{aligned}$$

since both equations (16) and (17) for β reduce to (15), when multiplied by β^2 and β, respectively. For every word $\mathbf{a}$, formula (12) is instantiated here as

$$(\mathbf{a})_\beta = (00)_\beta + \beta^{-2}\beta^2 = 1\,. \tag{18}$$

Observe that for instance, $\mathbf{a} = 00\,(010\,1000\,010)^\omega = 000\,(1010000\,100)^\omega$ can also be decomposed into the preperiodic part 000 and two quasi-repetends 1010000 and 100 with the periodic point $\varrho = \beta^3$.

We have introduced so-called quasi-periodic numbers (Šíma and Savický, 2018). We say that a real number c is *β-quasi-periodic within A* if *every* infinite β-expansion of c using the digits from A, is eventually quasi-periodic. Note that a number c that has no β-expansion at all, or has, in addition, a finite β-expansion whereas $0 \notin A$, is also considered formally to be β-quasi-periodic. For example, the numbers from the complement of the Cantor set are formally 3-quasi-periodic within $\{0, 2\}$, since they have no 3-expansion for the alphabet $\{0, 2\}$.

[4] We use the plus sign for denoting the alternation in regular expressions, which is the union of two sets of strings, corresponding to the logical OR operation.

Thus, a β-quasi-periodic number within A with at least two different quasi-repetends has uncountably many eventually quasi-periodic β-expansions, while at most countably many of them are eventually periodic which are generated using individual quasi-repetends as repetends. Moreover, any greedy eventually quasi-periodic β-expansion of a number can employ only one identical repetend, and hence it must be eventually periodic, although the number itself need not be β-quasi-periodic within A. This implies $\mathrm{QPer}(\beta) \subseteq \mathrm{Per}(\beta)$ where $\mathrm{QPer}(\beta)$ denotes the set of β-quasi-periodic numbers within A, which have an infinite β-expansion. In general, a number that is not β-quasi-periodic within A can still have some of its β-expansions eventually quasi-periodic.

For the bases β that are or are not Salem or Pisot numbers, we present examples of numbers that are β-quasi-periodic within the binary alphabet $A = \{0, 1\}$ which is the case widely used in the literature. Simpler examples can easily be found for larger alphabets and/or for arbitrary real digits.

Example 3 (Šíma and Savický, 2018, Example 4) Let $\beta \approx 1.722084$ be the real root of the polynomial $x^4 - x^3 - x^2 - x + 1$ such that $\beta > 1$, which is a Salem number. One can show that all β-expansions of the real number

$$c = \tfrac{1}{9}\left(4\beta^3 - 2\beta^2 - 2\beta - 5\right) \approx 0.672505 \tag{19}$$

using the binary alphabet $A = \{0, 1\}$, are generated by the ω-regular expression

$$\begin{aligned}&(100010 + 011(011101)^*100)^\omega \\ &\quad + (100010 + 011(011101)^*100)^*011(011101)^\omega ,\end{aligned} \tag{20}$$

and prove to be eventually quasi-periodic, which ensures that c is β-quasi-periodic within A. This example also illustrates that there can be constant, polynomial, or exponential number of distinct quasi-repetends of a given length. For instance, for a preperiodic part of the form $(100010)^k011$ for $k \geq 0$, we have only one quasi-repetend of length $6n$, for $n \geq 2$, namely $100(100010)^{n-1}011$, while for the preperiodic part $(100010)^k0$, we have $n \geq 1$ distinct quasi-repetends of length $6n$, namely $11(011101)^{n_1}100(100010)^{n_2}0$ where $n_1, n_2 \geq 0$ such that $n - 1 = n_1 + n_2$. Moreover, for the preperiodic part $1(000101)^k$, we have 2^{n-2} quasi-repetends of the form $00010(011(011101)^*100)^*1$ having the length $6n$, for $n \geq 2$.

Another example employs the *plastic constant* (i.e. the minimal Pisot number) $\beta \approx 1.324718$ which is the unique real root of the polynomial $x^3 -$

$x-1$, for which $c = 1$ proves to be β-quasi-periodic number within $A = \{0, 1\}$. In addition, there are also β-quasi-periodic numbers within $A = \{0, 1\}$ for the base β that is neither Pisot nor Salem number. For instance, let $\beta \approx 1.685137$ be the unique real root of the polynomial $x^5 - x^4 - x^2 - x - 1$, whose some Galois conjugates are in absolute value greater than 1. Then all the β-expansions of the real number $c = \frac{1}{3}(-\beta^4 + 3\beta^3 - 2\beta - 1) \approx 0.640563$, which are generated by the ω-regular expression $(10000+01(01111)^*10)^\omega + (10000+01(01111)^*10)^*01(01111)^\omega$, are eventually quasi-periodic.

On the other hand, we provide examples of rational and irrational numbers that are not β-quasi-periodic within A for rational and irrational bases β, despite their greedy and/or lazy β-expansion is eventually periodic.

Example 4 (Šíma and Savický, 2018, Example 3 and 5) Let $A = \{0, 1\}$. We first consider rational $\beta = \frac{3}{2} < \varphi$ which ensures there are uncountably many infinite β-expansions of the number 1 (see Paragraph 3.1) which all can be shown to be not eventually periodic. Thus, the number 1 is not $\frac{3}{2}$-quasi-periodic within $\{0, 1\}$.

Further consider irrational $\beta = \sqrt{2} \approx 1.414214$. The infinite words 110^ω and 001^ω are eventually periodic greedy and lazy β-expansions of $c = \frac{1}{2}(\beta + 1) \approx 1.207107$, respectively. Nevertheless, c is not β-quasi-periodic within A since any β-expansion of c with the prefix 0111 proves to be not eventually quasi-periodic. In addition, $c = \frac{1}{3}$ is an example of a rational number with the periodic greedy β-expansion $(0001)^\omega$ whose lazy β-expansion is not eventually periodic.

For Pisot bases β and digits from $\mathbb{Q}(\beta)$, the following theorem shows that $\mathrm{QPer}(\beta) = \mathrm{Per}(\beta) \subseteq \mathbb{Q}(\beta) \subseteq \overline{\mathrm{QPer}}(\beta)$ where $\overline{\mathrm{QPer}}(\beta)$ denotes the set of β-quasi-periodic numbers within A, including those with no infinite β-expansion.

Theorem 1 *(Šíma and Savický, 2018, Theorem 5) Let β be a Pisot number and assume $A \subset \mathbb{Q}(\beta)$. Then any number $c \in \mathbb{Q}(\beta)$ is β-quasi-periodic within A.*

3.3. Cut Languages Within the Chomsky hierarchy

Let $\Gamma \neq \emptyset$ be a finite alphabet which represents the digits through the mapping $\alpha : \Gamma \longrightarrow A$. We introduce a so-called *cut language* $L_{<c} \subseteq \Gamma^*$ over

alphabet Γ, which contains the representations of all finite β-expansions of the numbers that are less than a given real *threshold* c, that is,

$$L_{<c} = \left\{ z_1 \ldots z_n \in \Gamma^* \,\middle|\, \sum_{k=1}^{n} \alpha(z_k)\beta^{-k} < c \right\} . \tag{21}$$

The cut language $L_{>c}$ with the greater-than symbol is defined analogously. In other words, a cut language is composed of finite β-expansions of a *Dedekind cut* from which its name comes from. For instance, assume the base $\beta = 10$, the digits $A = \{0, 1, \ldots, 9\}$, and let α be the identity mapping on $\Gamma = A$. Then the cut language

$$L_{<\sqrt{2}-1} = \left\{ a_1 \ldots a_n \in A^* \,\middle|\, (a_1 \ldots a_n)_{10} = \tfrac{a_1}{10} + \tfrac{a_2}{100} + \cdots + \tfrac{a_n}{10^n} < \sqrt{2} - 1 \right\} \tag{22}$$

with the threshold $c = \sqrt{2} - 1$, contains all the finite decimal expansions of the (eventually non-periodic rational) numbers that are less than $\sqrt{2} - 1 \approx 0.414214$, e.g. the words ε, 4, 41, 414, 4142 belong to the cut language $L_{<\sqrt{2}-1}$, whereas the words 5, 42, 415, 4143 are not in $L_{<\sqrt{2}-1}$.

We have classified the cut languages within the Chomsky hierarchy by using the concept of quasi-periodic numbers introduced in Paragraph 3.2, which is summarized in the following Theorems 2 and 3.

Theorem 2 *(Šíma and Savický, 2018, Corollary 2) Any cut language $L_{<c} \subseteq \Gamma^*$ over alphabet Γ with base β is either regular if c is β-quasi-periodic within $A = \alpha(\Gamma)$, or non-context-free otherwise.*

Theorem 3 *(Šíma and Savický, 2018, Theorem 8) Let $\beta \in \mathbb{Q}$ be a rational base and $A = \alpha(\Gamma) \subset \mathbb{Q}$ be a set of rational digits encoded by alphabet Γ. Every cut language $L_{<c} \subseteq \Gamma^*$ with rational threshold $c \in \mathbb{Q}$ is context-sensitive.*

4. The Representation Theorem for 1ANNs

The main technical result of the paper is a syntactic characterization of languages accepted by 1ANNs, which is formulated in the following representation Theorem 4. The statement of this theorem is necessarily complicated because of complex 1ANN computational dynamics. Therefore, we will illustrate the theorem and its proof on the 1ANN $\mathcal{N}$ from running Example 1. Nevertheless, the resulting formula describing the languages recognized by

1ANNs is mainly based on the cut languages introduced in Paragraph 3.3, which are combined by usual operations including complementation, intersection, union, concatenation, Kleene star, reversal, the largest prefix-closed subset, and a letter-to-letter morphism. The analysis of the computational power of 1ANNs in Section 6 can thus be reduced to that of cut languages since regular and context-sensitive languages are closed under these operations.

The idea of the representation theorem is to characterize a language $L = \mathcal{L}(\mathcal{N}) \subseteq \Sigma^*$ over the input alphabet Σ, that is accepted by a 1ANN $\mathcal{N}$, in terms of simpler cut languages over an alphabet Γ. Namely, a word $\mathbf{x} \in L$ is characterized by a sequence of the network states $\mathbf{y}^{(0)}\mathbf{y}^{(1)} \ldots \mathbf{y}^{(t^*)} \in (\{0,1\}^{s-1} \times \mathbb{I})^*$ which $\mathcal{N}$ traverses during its accepting computation on the input $\mathbf{x}$, within the time t^* of update steps. However, the state alphabet $\{0,1\}^{s-1} \times \mathbb{I}$ of $\mathcal{N}$ is infinite since output values from the analog neuron are real numbers in the unit interval $\mathbb{I}$. Nevertheless, this unit interval $\mathbb{I}$ can be partitioned into a finite number p of subintervals such that the infinite state alphabet of $\mathcal{N}$ is represented by the finite alphabet Γ which is roughly $\{0,1\}^{s-1} \times \{1,\ldots,p\}$ where an analog output value from $\mathbb{I}$ is now replaced by an index of the subinterval to which this output value belongs. Moreover, the partition of $\mathbb{I}$ is such that the words over Γ that yield analog values in one of the subintervals, form an intersection of two cut languages (or their complements). This provides the desired characterization of L in terms of cut languages.

Theorem 4 *Let $\mathcal{N}$ be a 1ANN such that $0 < |w_{ss}| < 1$. For any $j \in V \setminus X$ such that $w_{js} \neq 0$, denote*

$$c_j(\tilde{\mathbf{y}}) = -\sum_{i=0}^{s-1} \frac{w_{ji}}{w_{js}}\, y_i \quad \textit{for } \tilde{\mathbf{y}} = (y_1, \ldots, y_{s-1}) \in \{0,1\}^{s-1} . \tag{23}$$

We define a base

$$\beta = \frac{1}{w_{ss}} \tag{24}$$

and a digit alphabet $A = A_1 \cup A_2$ *where*

$$\begin{aligned} A_1 &= \left\{ -c_s(\tilde{\mathbf{y}}) \,\middle|\, \tilde{\mathbf{y}} \in \{0,1\}^{s-1} \right\} && (25)\\ A_2 &= \left\{ a_0 + a_1\beta^{-1} + a_2\beta^{-2} \,\middle|\, a_0, a_1 \in A_1 \,,\; a_2 \in \{0,\beta\} \right\} . && (26) \end{aligned}$$

In addition, we introduce a set [5]

$$
\begin{aligned}
C \;&=\; \left\{ (c_j(\tilde{\mathbf{y}}) \,,\, -\operatorname{sgn}(w_{js})) \;\middle|\; \begin{array}{l} j \in V \setminus (X \cup \{s\}) \,,\, \tilde{\mathbf{y}} \in \{0,1\}^{s-1} \\ s.t.\;\; w_{js} \neq 0 \;\;\&\;\; c_j(\tilde{\mathbf{y}}) \in \mathbb{I} \end{array} \right\} \\
&\qquad \cup\; \big\{ (0,-1) \,,\, (0,1) \,,\, (1,-1) \,,\, (1,1) \big\} \subset \mathbb{I} \times \{-1,1\} \qquad (27) \\
&=\; \big\{ (c_1,s_1), (c_2,s_2), \ldots, (c_{p+1}, s_{p+1}) \big\} \,,
\end{aligned}
$$

encoding the $p+1$ *closed half-lines* $[0,+\infty)$, $(-\infty,0]$, $[1,+\infty)$, $(-\infty,1]$, *and*

$$
H_j(\tilde{\mathbf{y}}) = \left\{ \begin{array}{ll} [c_j(\tilde{\mathbf{y}}) \;,\; +\infty) & \textit{if } (c_j(\tilde{\mathbf{y}}) \,,\, -1) \in C \\ (-\infty \,,\, c_j(\tilde{\mathbf{y}})] & \textit{if } (c_j(\tilde{\mathbf{y}}) \,,\, 1) \in C \end{array} \right. \qquad (28)
$$

with the finite end-points called thresholds *from the unit interval* $\mathbb{I}$, *which is sorted lexicographically as*

$$
(c_1, s_1) < (c_2, s_2) < \cdots < (c_{p+1}, s_{p+1}) \,. \qquad (29)
$$

Then any language $L = \mathcal{L}(\mathcal{N}) \subseteq \Sigma^*$ *that is accepted by* $\mathcal{N}$, *can be written as*

$$
L = h\left((\mathcal{L} \cap R_0)^* \cdot \mathcal{L} \cap R \right) \qquad (30)
$$

where the dot operator denotes the concatenation and
- $h : \Gamma^* \longrightarrow \Sigma^*$ *is a letter-to-letter morphism (i.e.* $h(\Gamma) \subseteq \Sigma$*) from a set of strings over a finite alphabet* $\Gamma = \Gamma' \cup \Gamma''$ *such that* Γ' *is partitioned into* $\Gamma_1, \ldots, \Gamma_p$ *and* $\Gamma'' = \Gamma_\lambda \times \Gamma_\sigma$ *where* $\Gamma_\sigma = \Gamma_1 \cup \Gamma_p$ *and* $\Gamma_\lambda = \Gamma' \setminus \Gamma_\sigma$,
- $R \subseteq \Gamma^*$ *and* $R_0 = \Gamma'' \cdot \Gamma_\lambda^* \cdot \Gamma_\sigma \cup \Gamma_\sigma$ *are regular languages,*
- *language* $\mathcal{L}$ *is defined as*

$$
\mathcal{L} = \left(\bigcup_{r=1}^{p} L_r^R \cdot \Gamma_r \right)^{Pref} \qquad (31)
$$

in which S^{Pref} *denotes the largest prefix-closed subset of* $S \cup \Gamma \cup \{\varepsilon\}$, ε *is the*

[5] Clearly, the cardinality of C is upper bounded as $|C| = p+1 \leq (s-q-1)2^{s-1}+4$. Note also that the thresholds $c_j(\tilde{\mathbf{y}})$ in the definition of C are restricted to the unit interval $\mathbb{I}$ in order to fall in the state domain of the analog neuron s, according to (4).

empty string, S^R is the reversal of language S,

$$
\begin{aligned}
L_1 &= \overline{L_{>0}} \\
L_r &= \begin{cases} \overline{L_{<c_r}} \cap L_{<c_{r+1}} & \textit{if } s_r = -1 \ \& \ s_{r+1} = -1 \\ \overline{L_{<c_r}} \cap \overline{L_{>c_{r+1}}} & \textit{if } s_r = -1 \ \& \ s_{r+1} = 1 \\ L_{>c_r} \cap \overline{L_{<c_{r+1}}} & \textit{if } s_r = 1 \ \& \ s_{r+1} = -1 \\ L_{>c_r} \cap \overline{L_{>c_{r+1}}} & \textit{if } s_r = 1 \ \& \ s_{r+1} = 1 \end{cases} \quad \textit{for } r = 2, \ldots, p-1 \quad (32) \\
L_p &= \overline{L_{<1}}
\end{aligned}
$$

where $\overline{L_{<c}} = \Gamma_\lambda^* \cdot \Gamma'' \setminus L_{<c}$ denotes the complement,

$$L_{<c} = \left\{ z_1 \ldots z_n \in \Gamma_\lambda^* \cdot \Gamma'' \;\middle|\; \sum_{k=1}^{n} \alpha(z_k) \beta^{-k} < c \right\} \tag{33}$$

is a cut language (analogously $L_{>c}$), and $\alpha : \Gamma \longrightarrow A$ is a mapping to the digit alphabet such that $\alpha(\Gamma') \subseteq A_1$, $\alpha(\Gamma'') \subseteq A_2$, and for every $(z_1, z_2) \in \Gamma''$,

$$\alpha\big((z_1, z_2)\big) = \begin{cases} \alpha(z_1) + \alpha(z_2)\beta^{-1} & \textit{if } z_1 \in \Gamma_1 \\ \alpha(z_1) + \alpha(z_2)\beta^{-1} + \beta \cdot \beta^{-2} & \textit{if } z_1 \in \Gamma_p \,. \end{cases} \tag{34}$$

Example 1 (continuing from p. 7) We illustrate Theorem 4 by applying it to the 1ANN $\mathcal{N}$ from running Example 1, which is depicted in Figure 1. Thus, we have $\beta = 1/w_{55} = 3$ due to (24), and $A_1 = \{0, 2\}$ according to (25) where $c_5(0,0,0,0) = 0$ and $c_5(0,1,0,0) = -w_{52}/w_{55} = -2$ are the only distinct values of $c_5(\tilde{\mathbf{y}})$ for $\tilde{\mathbf{y}} \in \{0,1\}^4$ by (23) because $w_{5i} \neq 0$ only for $i = 2$ when $i \in \{0, \ldots, 4\}$. It follows that

$$A_2 = \left\{0, \tfrac{1}{3}, \tfrac{2}{3}, 1, 2, \tfrac{7}{3}, \tfrac{8}{3}, 3\right\} = A \tag{35}$$

which is produced by (26) for $a_0, a_1 \in \{0, 2\}$ and $a_2 \in \{0, 3\}$. Furthermore,

$$C = \left\{(0, -1), (0, 1), \left(\tfrac{1}{2}, -1\right), (1, -1), (1, 1)\right\} \tag{36}$$

according to (27), since $w_{j5} \neq 0$ only for $j = 4$ when $j \in V \setminus (X \cup \{s\}) = \{3, 4\}$, and $c_4(0,0,0,0) = -w_{40}/w_{45} = \frac{1}{2}$ is the only possible value of $c_4(\tilde{\mathbf{y}})$ for $\tilde{\mathbf{y}} \in \{0,1\}^4$ because $w_{4i} = 0$ for every $i = 1, \ldots, 4$, and $\operatorname{sgn}(w_{45}) = 1$. This encodes the corresponding $p + 1 = 5$ half-lines $[0, +\infty)$, $(-\infty, 0]$, $[1, +\infty)$, $(-\infty, 1]$, and $H_4(0,0,0,0) = [\frac{1}{2}, +\infty)$ by (28). It follows that the characterization (30) of $L = \mathcal{L}(\mathcal{N})$ is based on the languages

$$L_1 = \overline{L_{>0}}\,, \quad L_2 = L_{>0} \cap L_{<\frac{1}{2}}\,, \quad L_3 = \overline{L_{<\frac{1}{2}}} \cap L_{<1}\,, \quad L_4 = \overline{L_{<1}}\,, \tag{37}$$

which are defined using the cut languages according to (32). We will continue below in this running example by instantiating the alphabet Γ and the mapping α in the proof of Theorem 4.

PROOF (THEOREM 4). An input string $\mathbf{x} = x_1 \ldots x_n \in \Sigma^*$ belongs to $L = \mathcal{L}(\mathcal{N})$ iff there is an accepting computation of $\mathcal{N}$ on $\mathbf{x}$ which is described by a sequence of its states $\mathbf{y}^{(0)}, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(nd)}$ with $y_{\text{out}}^{(nd)} = 1$, following the computational dynamics (1), (2), and the input/output protocol (5), (6). For every time instant $t = 0, 1, \ldots, nd - 1$ and for any non-input binary neuron $j \in V_{t+1} \setminus (X \cup \{s\})$ whose state is updated at time $t + 1$, we have

$$y_j^{(t+1)} = 1 \quad \text{iff} \quad \sum_{i=0}^{s-1} w_{ji} y_i^{(t)} + w_{js} y_s^{(t)} \geq 0 \tag{38}$$

according to (1)–(3). For $w_{js} \neq 0$, this condition can be rewritten as

$$y_j^{(t+1)} = 1 \quad \text{iff} \quad y_s^{(t)} \in H_j \left(\tilde{\mathbf{y}}^{(t)} \right) \tag{39}$$

where $H_j \left(\tilde{\mathbf{y}}^{(t)} \right)$ is a half-line defined by (28), (27), and (23) for the states of binary neurons $\tilde{\mathbf{y}}^{(t)} = \left(y_1^{(t)}, \ldots, y_{s-1}^{(t)} \right) \in \{0, 1\}^{s-1}$ at time $t \geq 0$.

By using the order of thresholds (29), we partition the unit interval $\mathbb{I}$ into the p intervals,

$$I_r = \begin{cases} [c_r \, , \, c_{r+1}) & \text{if } \; s_r = -1 \;\; \& \;\; s_{r+1} = -1 \\ [c_r \, , \, c_{r+1}] & \text{if } \; s_r = -1 \;\; \& \;\; s_{r+1} = 1 \\ (c_r \, , \, c_{r+1}) & \text{if } \; s_r = 1 \;\; \& \;\; s_{r+1} = -1 \\ (c_r \, , \, c_{r+1}] & \text{if } \; s_r = 1 \;\; \& \;\; s_{r+1} = 1 \end{cases} \quad \text{for } \; r = 1, \ldots, p \tag{40}$$

corresponding to (32). Note that if $c_r = c_{r+1}$ for some $1 \leq r \leq p$, we know $-1 = s_r < s_{r+1} = 1$ due to C is a lexicographically ordered set, which produces the degenerate interval $[c_r, c_r]$. Thus, $I_1 = [0, 0]$ and $I_p = [1, 1]$ because $(0, -1), (0, 1), (1, -1), (1, 1) \in C$ according to (27). It follows from (39) and (40) that the states of non-input binary neurons $y_{q+1}^{(t+1)}, \ldots, y_{s-1}^{(t+1)} \in \{0, 1\}$ at time instant $t + 1$, are uniquely determined by both $\tilde{\mathbf{y}}^{(t)} \in \{0, 1\}^{s-1}$ and the interval I_r ($1 \leq r \leq p$) to which the analog state $y_s^{(t)}$ belongs.

Since we assume that the states of input neurons $X \subset V$ and analog unit $s \in V_{d\tau}$ are simultaneously updated once in a macroscopic time step $\tau \geq 1$, we have

$$\begin{array}{l} \mathbf{y}_X^{(\tau d)} = \mathbf{y}_X^{(\tau d+1)} = \mathbf{y}_X^{(\tau d+2)} = \cdots = \mathbf{y}_X^{((\tau+1)d-1)} \\ y_s^{(\tau d)} = y_s^{(\tau d+1)} = y_s^{(\tau d+2)} = \cdots = y_s^{((\tau+1)d-1)} \end{array} \quad \text{for } \tau = 0, \ldots, n-1\,, \tag{41}$$

by (2) and (5). For every $\tau = 0, \ldots, n$, we define the index $r_\tau \in \{1, \ldots, p\}$ of interval I_{r_τ} so that $y_s^{(\tau d)} \in I_{r_\tau}$, especially, $r_0 = 1$ because $y_s^{(0)} = 0$ is assumed. As the computational dynamics of $\mathcal{N}$ is deterministic satisfying $V_{t+d} = V_t$ for every $t \geq 1$, and because of (41), the states of binary neurons, $\tilde{\mathbf{y}}^{((\tau+1)d-1)} \in \{0,1\}^{s-1}$ at the time instant $(\tau+1)d-1$ for any macroscopic time step $\tau \geq 0$, are uniquely determined by both $\tilde{\mathbf{y}}^{(\tau d)} \in \{0,1\}^{s-1}$ and $r_\tau \in \{1, \ldots, p\}$, which defines the mapping $\nu : \{0,1\}^{s-1} \times \{1, \ldots, p\} \longrightarrow \{0,1\}^{s-1}$ that meets

$$\tilde{\mathbf{y}}^{((\tau+1)d-1)} = \nu\left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right)\,. \tag{42}$$

For the non-input binary neurons $q+1, \ldots, s-1 \in V \setminus (X \cup \{s\})$, this mapping extends uniquely to the next time instant by $\overline{\nu} : \{0,1\}^{s-1} \times \{1, \ldots, p\} \longrightarrow \{0,1\}^{s-q-1}$, satisfying

$$\left(y_{q+1}^{((\tau+1)d)}, \ldots, y_{s-1}^{((\tau+1)d)}\right) = \overline{\nu}\left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right)\,. \tag{43}$$

Thus, we encode the underlying computation by using a string $z_0 \ldots z_n \in \Gamma^*$ over a finite alphabet $\Gamma = \Gamma' \cup \Gamma''$ which consists of *basic* letters

$$\Gamma' = \bigcup_{r=1}^{p} \Gamma_r \quad \text{where} \quad \Gamma_r = \{0,1\}^{s-1} \times \{r\} \quad \text{for } r = 1, \ldots, p\,, \tag{44}$$

and so-called *contextual* symbols

$$\Gamma'' = \Gamma_\lambda \times \Gamma_\sigma \quad \text{where} \quad \Gamma_\sigma = \Gamma_1 \cup \Gamma_p \quad \text{and} \quad \Gamma_\lambda = \Gamma' \setminus \Gamma_\sigma\,. \tag{45}$$

In particular, the underlying string is composed of $z_0 = \left(\tilde{\mathbf{y}}^{(0)}, r_0\right)$ with $r_0 = 1$ and

$$z_\tau = \begin{cases} \left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right) \in \Gamma' \\ \qquad\qquad \text{if } r_{\tau-1} \notin \{1,p\} \text{ or } r_\tau \in \{1,p\} \\ \left(\left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right), \left(\tilde{\mathbf{y}}^{((\tau-1)d)}, r_{\tau-1}\right)\right) \in \Gamma'' \\ \qquad\qquad \text{if } r_{\tau-1} \in \{1,p\}\ \&\ r_\tau \notin \{1,p\} \end{cases} \quad \text{for } \tau = 1, \ldots, n\,. \tag{46}$$

Clearly, the basic letter $z_\tau \in \Gamma_{r_\tau} \subset \Gamma'$ encodes the network state $\mathbf{y}^{(\tau d)} \in \{0,1\}^{s-1} \times \mathbb{I}$ at the macroscopic time step $\tau \geq 0$, by using the states of binary neurons $\tilde{\mathbf{y}}^{(\tau d)} \in \{0,1\}^{s-1}$ and the index $r_\tau \in \{1,\ldots,p\}$. Similarly for the contextual symbols $z_\tau \in \Gamma''$ whose additional role is to mark the points of the computation at which the state value $y_s^{(\tau d)} \in \mathbb{I}$ of the analog unit s enters the linear part the activation function (4), which means $y_s^{(\tau d)} \in (0,1)$ (i.e. $1 < r_\tau < p$) whereas $y_s^{((\tau-1)d)} \in \{0,1\}$ (i.e. $r_{\tau-1} \in \{1,p\}$). Thus, any contextual symbol followed by basic letters from Γ_λ, is the starting letter of a reverse word in a cut language as will be argued below. Therefore, the contextual symbols encode an extra information about the previous network state which is needed for a proper initialization of the sum in the cut language definition (33) (for $k=n$). We define the mapping $\alpha : \Gamma \longrightarrow A$ for any $z \in \Gamma$ as

$$\alpha(z) = \begin{cases} -c_s\left(\nu(z)\right) \quad \text{if } z \in \Gamma' \\ -c_s\left(\nu(\tilde{\mathbf{y}},r)\right) - c_s\left(\nu(\tilde{\mathbf{y}}',1)\right)\beta^{-1} \\ \qquad\qquad \text{if } z = ((\tilde{\mathbf{y}},r),(\tilde{\mathbf{y}}',1)) \in \Gamma'' \\ -c_s\left(\nu(\tilde{\mathbf{y}},r)\right) - c_s\left(\nu(\tilde{\mathbf{y}}',p)\right)\beta^{-1} + \beta\cdot\beta^{-2} \\ \qquad\qquad \text{if } z = ((\tilde{\mathbf{y}},r),(\tilde{\mathbf{y}}',p)) \in \Gamma'' \end{cases} \tag{47}$$

by using (23) and (42), which satisfies $\alpha(\Gamma') \subseteq A_1$, $\alpha(\Gamma'') \subseteq A_2$, and (34).

Example 1 (continuing from p. 20) The ongoing proof of Theorem 4 is accompanied by the running example of the 1ANN language acceptor $\mathcal{N}$ from Figure 1, for which the preceding definitions are now instantiated. Thus, consider the accepting computation by $\mathcal{N}$ on the input word 1100, as is presented in Table 1. Since $w_{j5} \neq 0$ only for the binary neuron $j=4$, condition (39) applies to $4 \in V_{t+1}$ for $t \geq 0$:

$$y_4^{(t+1)} = 1 \quad \text{iff} \quad y_5^{(t)} \in H_4(0,0,0,0) = [\tfrac{1}{2}, +\infty) \tag{48}$$

(cf. Table 1). The unit interval $\mathbb{I}$ is partitioned into the $p=4$ subintervals

$$I_1 = [0,0], \quad I_2 = \left(0, \tfrac{1}{2}\right), \quad I_3 = \left[\tfrac{1}{2}, 1\right), \quad I_4 = [1,1], \tag{49}$$

according to (40) and (36). It follows that

$$r_0 = 1, \quad r_1 = 3, \quad r_2 = 3, \quad r_3 = 2, \quad r_4 = 2, \tag{50}$$

satisfying $y_5^{(2\tau)} \in I_{r_\tau}$ by Table 1. The mapping $\nu : \{0,1\}^4 \times \{1,2,3,4\} \longrightarrow \{0,1\}^4$ meets

$$\tilde{\mathbf{y}}^{(1)} = \nu\left(\tilde{\mathbf{y}}^{(0)}, r_0\right) = \nu((0,1,1,0),1) = (0,1,1,0) \tag{51}$$

$$\tilde{\mathbf{y}}^{(3)} = \nu\left(\tilde{\mathbf{y}}^{(2)}, r_1\right) = \nu((0,1,1,0),3) = (0,1,1,1) \tag{52}$$

$$\tilde{\mathbf{y}}^{(5)} = \nu\left(\tilde{\mathbf{y}}^{(4)}, r_2\right) = \nu((1,0,0,1),3) = (1,0,0,1) \tag{53}$$

$$\tilde{\mathbf{y}}^{(7)} = \nu\left(\tilde{\mathbf{y}}^{(6)}, r_3\right) = \nu((1,0,0,1),2) = (1,0,0,0) \tag{54}$$

by (42).

The sequence of states of $\mathcal{N}$ at the macroscopic time steps $\tau = 0, 1, 2, 3, 4$ during the computation on the input $\mathbf{x} = 1100$ is encoded by a string $z_0 z_1 z_2 z_3 z_4 \in \Gamma^*$ over the alphabet $\Gamma = \Gamma' \cup \Gamma''$ which is composed of the basic letters $\Gamma' = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ where $\Gamma_r = \{0,1\}^4 \times \{r\}$ for $r = 1, 2, 3, 4$, and the contextual symbols $\Gamma'' = \Gamma_\lambda \times \Gamma_\sigma$ where $\Gamma_\lambda = \Gamma_2 \cup \Gamma_3$ and $\Gamma_\sigma = \Gamma_1 \cup \Gamma_4$, according to (44) and (45), respectively. It follows from (46) that

$$\begin{aligned} z_0 &= \left(\tilde{\mathbf{y}}^{(0)}, r_0\right) = ((0,1,1,0),1) \in \Gamma_1 \subset \Gamma' && (55) \\ z_1 &= \left(\left(\tilde{\mathbf{y}}^{(2)}, r_1\right), \left(\tilde{\mathbf{y}}^{(0)}, r_0\right)\right) \\ &= (((0,1,1,0),3),((0,1,1,0),1)) \in \Gamma_3 \times \Gamma_1 \subset \Gamma'' && (56) \\ z_2 &= \left(\tilde{\mathbf{y}}^{(4)}, r_2\right) = ((1,0,0,1),3) \in \Gamma_3 \subset \Gamma' && (57) \\ z_3 &= \left(\tilde{\mathbf{y}}^{(6)}, r_3\right) = ((1,0,0,1),2) \in \Gamma_2 \subset \Gamma' && (58) \\ z_4 &= \left(\tilde{\mathbf{y}}^{(8)}, r_4\right) = ((1,0,1,0),2) \in \Gamma_2 \subset \Gamma' && (59) \end{aligned}$$

for which the mapping $\alpha : \Gamma \longrightarrow \left\{0, \frac{1}{3}, \frac{2}{3}, 1, 2, \frac{7}{3}, \frac{8}{3}, 3\right\}$ produces

$$\begin{aligned} \alpha(z_0) &= -c_5(\nu(z_0)) = -c_5(\nu((0,1,1,0),1)) = -c_5(0,1,1,0) = 2 && (60) \\ \alpha(z_1) &= \alpha(((0,1,1,0),3),((0,1,1,0),1)) \\ &= -c_5(\nu((0,1,1,0),3)) - c_5(\nu((0,1,1,0),1))\beta^{-1} \\ &= -c_5(0,1,1,1) - c_5(0,1,1,0) \cdot 3^{-1} = \tfrac{8}{3} && (61) \\ \alpha(z_2) &= -c_5(\nu(z_2)) = -c_5(\nu((1,0,0,1),3)) = -c_5(1,0,0,1) = 0 && (62) \\ \alpha(z_3) &= -c_5(\nu(z_3)) = -c_5(\nu((1,0,0,1),2)) = -c_5(1,0,0,0) = 0 && (63) \end{aligned}$$

by (47), (55)–(59), (51)–(54), and (23).

We further continue in the proof of Theorem 4. Let $0 = \tau_1 < \tau_2 < \cdots < \tau_m \leq n$ be all the indices such that $z_{\tau_\ell} \in \Gamma_\sigma$, which implies $r_{\tau_\ell} \in \{1, p\}$ and $y_s^{(\tau_\ell d)} \in \{0,1\}$, for $\ell = 1, \ldots, m$, and formally denote $\tau_0 = -1$ and $\tau_{m+1} = n$. For each $\ell \in \{1, \ldots, m\}$ such that $\tau_\ell + 1 < \tau_{\ell+1}$, and for every $\tau = \tau_\ell, \ldots, \tau_{\ell+1} - 2$, we know $z_{\tau+1} \notin \Gamma_\sigma$ which ensures

$$0 < y_s^{((\tau+1)d)} = \xi_s^{((\tau+1)d-1)} = \sum_{i=0}^{s-1} w_{si} y_i^{((\tau+1)d-1)} + w_{ss} y_s^{(\tau d)} < 1\,. \qquad (64)$$

The recursive formula (64) for the analog state $y_s^{(\tau d)}$ is applied $(\tau - \tau_\ell + 1)$

times, which results in

$$\xi_s^{((\tau+1)d-1)} = \sum_{k=1}^{\tau-\tau_\ell+1} w_{ss}^k \left(\sum_{i=0}^{s-1} \frac{w_{si}}{w_{ss}} \, y_i^{((\tau-k+2)d-1)} \right) + \, {w_{ss}}^{\tau-\tau_\ell+1} y_s^{(\tau_\ell d)} \,, \tag{65}$$

for every $\tau = \tau_\ell, \ldots, \tau_{\ell+1}-1$. By using (23), (24), and (42), this yields

$$\xi_s^{((\tau+1)d-1)} = - \sum_{k=1}^{\tau-\tau_\ell+1} c_s \left(\nu \left(\tilde{\mathbf{y}}^{((\tau-k+1)d)}, r_{\tau-k+1} \right) \right) \beta^{-k} \, + \, \frac{y_s^{(\tau_\ell d)}}{w_{ss}} \, \beta^{-(\tau-\tau_\ell+2)}, \tag{66}$$

which can be written down as

$$\begin{aligned} \xi_s^{((\tau+1)d-1)} \quad &= \quad - \sum_{k=1}^{\tau-\tau_\ell-1} c_s \left(\nu \left(\tilde{\mathbf{y}}^{((\tau-k+1)d)}, r_{\tau-k+1} \right) \right) \beta^{-k} \\ &\quad + \Bigg(- c_s \left(\nu \left(\tilde{\mathbf{y}}^{((\tau_\ell+1)d)}, r_{\tau_\ell+1} \right) \right) - c_s \left(\nu \left(\tilde{\mathbf{y}}^{(\tau_\ell d)}, r_{\tau_\ell} \right) \right) \beta^{-1} \\ &\qquad + \frac{y_s^{(\tau_\ell d)}}{w_{ss}} \, \beta^{-2} \Bigg) \beta^{-(\tau-\tau_\ell)} \,. \end{aligned} \tag{67}$$

Hence,

$$\xi_s^{((\tau+1)d-1)} = \quad \sum_{k=1}^{\tau-\tau_\ell} \alpha(z_{\tau-k+1}) \beta^{-k} \tag{68}$$

according to (46) and (47), as $z_{\tau_\ell+1} \in \Gamma''$ due to $r_{\tau_\ell} \in \{1,p\}$ and $r_{\tau_\ell+1} \notin \{1,p\}$ for $\tau_\ell + 1 < \tau_{\ell+1}$.

For each $\ell \in \{1,\ldots,m\}$ such that $\tau_\ell + 1 < \tau_{\ell+1}$, and for every $\tau = \tau_\ell+1, \ldots, \tau_{\ell+1}-1$, we know $z_{\tau_\ell+1} \ldots z_\tau \in \Gamma'' \cdot \Gamma_\lambda^*$. According to (4), (32), (33), (40), and (68), $z_{\tau_\ell+1} \ldots z_\tau \in L_r^R$ iff $y_s^{((\tau+1)d)} = \sigma\left(\xi_s^{((\tau+1)d-1)}\right) = \sigma\left(\sum_{k=1}^{\tau-\tau_\ell} \alpha(z_{\tau-k+1})\beta^{-k}\right) \in I_r$ iff $r_{\tau+1} = r$ by the definition of $r_{\tau+1}$, iff $z_{\tau+1} \in \Gamma_r$ due to (46). It follows that for every $\ell = 1, \ldots, m-1$, substring

$$z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in \mathcal{L} \cap R_0 = \left(\bigcup_{r=1}^{p} L_r^R \cdot \Gamma_r \right)^{Pref} \cap \Gamma'' \cdot \Gamma_\lambda^* \cdot \Gamma_\sigma \tag{69}$$

since any of its prefix $z_{\tau_\ell+1} \ldots z_\tau \in L_r^R \subseteq \Gamma'' \cdot \Gamma_\lambda^*$ for $\tau_\ell + 1 \le \tau < \tau_{\ell+1}$, is followed by $z_{\tau+1} \in \Gamma_r$, including $z_{\tau_{\ell+1}} \in \Gamma_\sigma$ for $\tau = \tau_{\ell+1}-1$. Analogously,

$z_{\tau_m+1} \dots z_n \in \mathcal{L}$. In addition, for any $\ell \in \{0, \dots, m-1\}$ such that $\tau_\ell + 1 = \tau_{\ell+1}$, also $z_{\tau_{\ell+1}} \in \Gamma_\sigma \subseteq \mathcal{L} \cap R_0$. It follows that any computation by $\mathcal{N}$ is encoded by

$$z_0 \dots z_n = \left(\prod_{\ell=0}^{m-1} z_{\tau_\ell+1} \dots z_{\tau_{\ell+1}} \right) \cdot z_{\tau_m+1} \dots z_n \in (\mathcal{L} \cap R_0)^* \cdot \mathcal{L} \tag{70}$$

where the product symbol is used for the repeated concatenation of strings.

Example 1 (continuing from p. 23) The preceding formulas are again illustrated on the running example of the 1ANN language acceptor $\mathcal{N}$ (Figure 1) and its accepting computation on the input word 1100 of length $n = 4$ (Table 1), which is encoded by the string $z_0 z_1 z_2 z_3 z_4 \in \Gamma^*$ specified in (55)–(59). In this string, $z_0 \in \Gamma_1$ is the only symbol that belongs to $\Gamma_\sigma = \Gamma_1 \cup \Gamma_4$, which means $m = 1$ and $\tau_1 = 0$, whereas formally $\tau_0 = -1$ and $\tau_2 = 4$. Thus, let $\ell = 1$ which meets $1 = \tau_1 + 1 = \tau_\ell + 1 < \tau_{\ell+1} = \tau_2 = 4$. For every $\tau = 0, 1, 2$ we have $z_{\tau+1} \notin \Gamma_\sigma$ which validates the recursive formula $0 < y_5^{(2\tau+2)} = \xi_5^{(2\tau+1)} = w_{52} y_2^{(2\tau+1)} + w_{55} y_5^{(2\tau)} = \frac{2}{3} y_2^{(2\tau+1)} + \frac{1}{3} y_5^{(2\tau)} < 1$, by (64). According to (68), this formula is used for deriving the equation $y_5^{(2\tau+2)} = \xi_5^{(2\tau+1)} = \sum_{k=1}^{\tau} \alpha(z_{\tau-k+1}) \beta^{-k}$ for every $\tau = 0, 1, 2, 3$, which is instantiated as

$$\begin{aligned}
y_5^{(2)} &= 0 && (71)\\
y_5^{(4)} &= \alpha(z_1)\beta^{-1} = \tfrac{8}{3} \cdot \tfrac{1}{3} = \tfrac{8}{9} && (72)\\
y_5^{(6)} &= \alpha(z_2)\beta^{-1} + \alpha(z_1)\beta^{-2} = \tfrac{8}{3} \cdot \tfrac{1}{9} = \tfrac{8}{27} && (73)\\
y_5^{(8)} &= \alpha(z_3)\beta^{-1} + \alpha(z_2)\beta^{-2} + \alpha(z_1)\beta^{-3} = \tfrac{8}{3} \cdot \tfrac{1}{27} = \tfrac{8}{81} && (74)
\end{aligned}$$

(cf. Table 1) by using (60)–(63). Obviously, $z_0 \in \Gamma_1 \subset \Gamma_\sigma \subset \mathcal{L} \cap R_0$. Moreover, we know that the strings z_1, $z_1 z_2$, $z_1 z_2 z_3$ belong to $\Gamma'' \cdot \Gamma_\lambda^*$ and

$$\begin{aligned}
z_1 \in L_3^R \quad &\text{iff} \quad y_5^{(4)} \in I_3 \quad \text{iff} \quad z_2 \in \Gamma_3 && (75)\\
z_1 z_2 \in L_2^R \quad &\text{iff} \quad y_5^{(6)} \in I_2 \quad \text{iff} \quad z_3 \in \Gamma_2 && (76)\\
z_1 z_2 z_3 \in L_2^R \quad &\text{iff} \quad y_5^{(8)} \in I_2 \quad \text{iff} \quad z_4 \in \Gamma_2 && (77)
\end{aligned}$$

due to (37), (49), (50), and (72)–(74), which ensures

$$z_1 z_2 \in L_3^R \cdot \Gamma_3 \,, \quad z_1 z_2 z_3 \in L_2^R \cdot \Gamma_2 \,, \quad z_1 z_2 z_3 z_4 \in L_2^R \cdot \Gamma_2 \,, \tag{78}$$

implying $z_1 z_2 z_3 z_4 \in \mathcal{L}$ by (31). Hence $z_0 z_1 z_2 z_3 z_4 \in (\mathcal{L} \cap R_0) \cdot \mathcal{L}$.

We will finish the proof of Theorem 4. The role of language $R \subseteq \Gamma^*$ in (30) is to restrict the strings $z_0 \ldots z_n \in \mathcal{L}^*$ only to those encoding valid accepting computations of $\mathcal{N}$, mainly with respect to the macroscopic computational dynamics of non-input binary neurons and to the consistency of symbols from Γ_σ and Γ''. In particular, these strings (if nonempty) must start with an initial letter $z_0 = \left(\tilde{\mathbf{y}}^{(0)}, r_0\right) \in \Gamma'$ such that $\mathbf{y}^{(0)} = \left(\tilde{\mathbf{y}}^{(0)}, y_s^{(0)}\right) \in \{0,1\}^{s-1} \times \mathbb{I}$ is the initial state of $\mathcal{N}$, including the one-hot encoding $\mathbf{y}_X^{(0)}$ of the first input symbol according to (5), and the initial output $y_s^{(0)} = 0$ of analog unit s, which corresponds to $r_0 = 1$. Any subsequent letter $z_\tau \in \Gamma$, for $1 \leq \tau \leq n$, has to be either a basic symbol $z_\tau = \left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right) \in \Gamma'$ if $r_{\tau-1} \notin \{1,p\}$ or $r_\tau \in \{1,p\}$, or a contextual symbol $z_\tau = \left(\left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right), \left(\tilde{\mathbf{y}}^{((\tau-1)d)}, r_{\tau-1}\right)\right) \in \Gamma''$ which comes after letter $z_{\tau-1} = \left(\tilde{\mathbf{y}}^{((\tau-1)d)}, r_{\tau-1}\right) \in \Gamma_\sigma$, if $r_{\tau-1} \in \{1,p\}$ and $r_\tau \notin \{1,p\}$. Each symbol z_τ, for $0 \leq \tau < n$, must be followed by $\left(\tilde{\mathbf{y}}^{((\tau+1)d)}, r_{\tau+1}\right) \in \Gamma'$ or $\left(\left(\tilde{\mathbf{y}}^{((\tau+1)d)}, r_{\tau+1}\right), \left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right)\right) \in \Gamma''$ such that $\tilde{\mathbf{y}}^{((\tau+1)d)} = \left(\mathbf{y}_X^{(\tau d)}, \overline{\nu}\left(\tilde{\mathbf{y}}^{(\tau d)}, r_\tau\right)\right)$ according to (5) and (43), and z_n satisfying $y_{\text{out}}^{(\tau n)} = 1$ terminates the string by (6). In addition, $\varepsilon \in R$ if $y_{\text{out}}^{(0)} = 1$. Furthermore, for any $\ell \in \{1, \ldots, m-1\}$ such that $\tau_\ell + 1 = \tau_{\ell+1}$, which ensures $z_{\tau_\ell} = \left(\tilde{\mathbf{y}}^{(\tau_\ell d)}, r_{\tau_\ell}\right) \in \Gamma_\sigma$ and $z_{\tau_{\ell+1}} = \left(\tilde{\mathbf{y}}^{(\tau_{\ell+1} d)}, r_{\tau_{\ell+1}}\right) \in \Gamma_\sigma$ with $r_{\tau_\ell}, r_{\tau_{\ell+1}} \in \{1,p\}$, the valid computation must satisfy

$$r_{\tau_{\ell+1}} = \begin{cases} 1 & \text{if } c_s\left(\nu\left(z_{\tau_\ell}\right)\right)\beta^{-1} \geq 0 & \& \quad r_{\tau_\ell} = 1 \\ 1 & \text{if } \left(1 - c_s\left(\nu\left(z_{\tau_\ell}\right)\right)\right)\beta^{-1} \leq 0 & \& \quad r_{\tau_\ell} = p \\ p & \text{if } c_s\left(\nu\left(z_{\tau_\ell}\right)\right)\beta^{-1} \leq -1 & \& \quad r_{\tau_\ell} = 1 \\ p & \text{if } \left(1 - c_s\left(\nu\left(z_{\tau_\ell}\right)\right)\right)\beta^{-1} \geq 1 & \& \quad r_{\tau_\ell} = p \end{cases} \tag{79}$$

according to (4) and (40), since $\xi_s^{((\tau+1)d)} = \left(y_s^{(\tau d)} - c_s\left(\nu\left(z_\tau\right)\right)\right)\beta^{-1}$ due to (1), (23), (24), (41), and (42), where $y_s^{(\tau d)} = 0$ if $r_{\tau_\ell} = 1$ while $y_s^{(\tau d)} = 1$ if $r_{\tau_\ell} = p$. Obviously, language R can be recognized by a finite automaton and hence it is regular.

Finally, the letter-to-letter morphism $h : \Gamma^* \longrightarrow \Sigma^*$ is defined as $h(z) = \lambda_i \in \Sigma$ for $z = (\tilde{\mathbf{y}}, r) \in \Gamma'$ or for $y = ((\tilde{\mathbf{y}}, r), (\tilde{\mathbf{y}}', r')) \in \Gamma''$ where $\tilde{\mathbf{y}} = (y_1, \ldots, y_{s-1})$ such that $y_i = 1$ for $i \in X$, which extracts the input strings accepted by $\mathcal{N}$ according to (5). This completes the proof that L can be written as (30). $\square$

5. The Reverse Implication of the Representation Theorem

In this section, we attempt to complete the syntactic characterization of 1ANNs from Section 4. Since it is unclear whether the languages accepted by 1ANNs are closed under morphism, the implication in the representation Theorem 4 can only be partially reversed, cf. (30) and (80):

Theorem 5 *Let β be a real base such that $|\beta| > 1$, $A \subset \mathbb{R}$ be a finite set of real digits, and $\{0,1\} \times \{-1,1\} \subseteq C = \big\{(c_1, s_1), (c_2, s_2), \ldots, (c_{p+1}, s_{p+1})\big\} \subset \mathbb{I} \times \{-1,1\}$ be sorted lexicographically according to (29). Then any language $L \subseteq \Gamma^*$ that can be written as*

$$L = (\mathcal{L} \cap R_0)^* \cdot \mathcal{L} \cap R \tag{80}$$

where languages $\mathcal{L}, R, R_0 \subset \Gamma^*$ are defined as in Theorem 4, can be recognized by a 1ANN, $\mathcal{N}$, that is, $L = \mathcal{L}(\mathcal{N})$.

Proof. Let $L \subseteq \Gamma^*$ be a language that can be written as (80). We will construct a 1ANN $\mathcal{N}$ such that $L = \mathcal{L}(\mathcal{N})$. The network $\mathcal{N}$ is schematically depicted in Figure 2 where the layers or their parts are indicated and only a few representative units and connections are drawn while the precise definition of $\mathcal{N}$ follows step by step throughout the proof. The directed edges connecting units are labeled with the corresponding weights whereas the edges drawn without an originating unit correspond to the bias parameters.

According to Horne and Hush (1996), one can construct a binary-state (size-optimal) neural network $\mathcal{N}'$ that simulates a finite automaton $\mathcal{A}$ recognizing the regular language $R \subseteq \Gamma^*$ within the time overhead $d = 4$ for processing one input symbol. Denote by V' the set of binary units of $\mathcal{N}'$, including the input neurons $X \subset V'$ and the output neuron $\mathrm{out} \in V'$ which implement the input/output protocol (5) and (6), respectively, using the notation $\Gamma = \{\lambda_i \,|\, i \in X\}$. Moreover, $V'_t \subseteq V'$ are the sets of neurons that update their states at time $t \geq 1$ according to (2), which satisfy $V'_{t+4} = V'_t$ for every $t \geq 1$, where $V'_1, V'_2, V'_3, V'_4 \subseteq V'$ correspond to the four layers in $\mathcal{N}'$ that evaluate the transition function of $\mathcal{A}$ using the method of threshold circuit synthesis (Lupanov, 1973; cf. Šíma, 2014a). Thus, $\mathrm{out} \in V'_4$, and we have initially $y_{\mathrm{out}}^{(0)} = 1$ iff $\varepsilon \in L$ iff $\varepsilon \in R$ due to $\varepsilon \in \mathcal{L}$ by the definition.

We add a subnetwork that recognizes the language $(\mathcal{L} \cap R_0)^* \cdot \mathcal{L}$, to the neural network $\mathcal{N}'$, by extending the respective layers of $\mathcal{N}'$ with binary-state

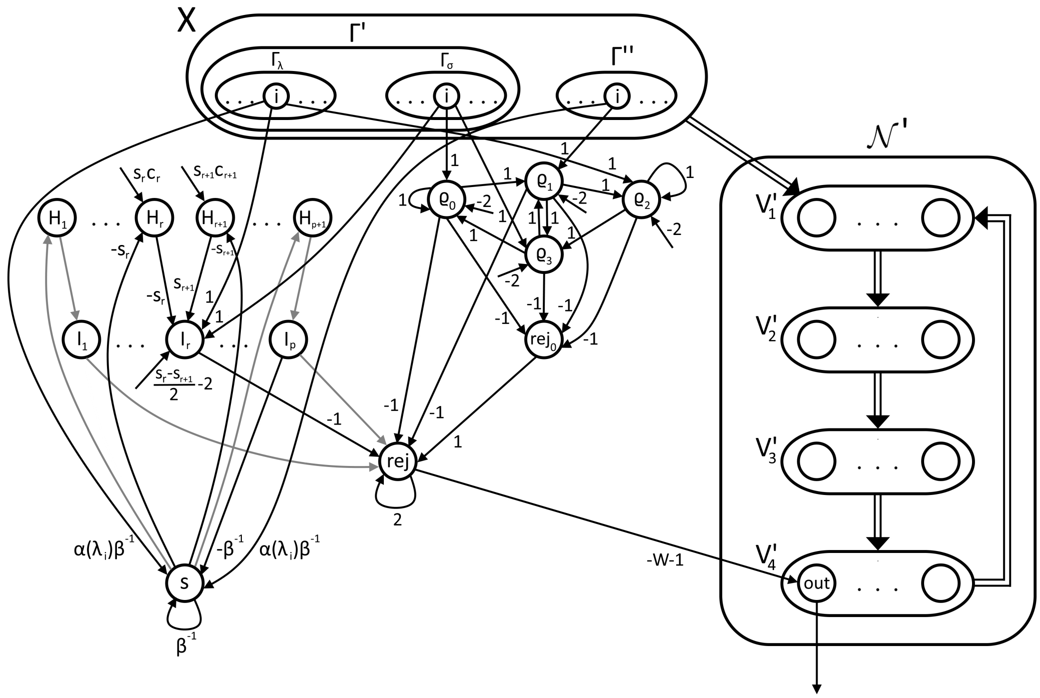

Figure 2: A schema of the neural network $\mathcal{N}$ that accepts the language (80).

neurons and one extra analog unit s as

$$V_1 = V_1' \cup \{H_r \,|\, r = 1, \ldots, p+1\} \cup \{\varrho_0, \varrho_1, \varrho_2, \varrho_3\} \tag{81}$$
$$V_2 = V_2' \cup \{I_r \,|\, r = 1, \ldots, p\} \cup \{\mathrm{rej}_0\} \tag{82}$$
$$V_3 = V_3' \cup \{\mathrm{rej}\} \tag{83}$$
$$V_4 = V_4' \cup \{s\}\,, \tag{84}$$

which determines the computational dynamics (2) of $\mathcal{N}$ because $V_{t+4} = V_t$ for every $t \geq 1$. For simplicity, we identify the names of added neurons in V_1 and V_2 with the intervals H_r and I_r, respectively, having thus two meanings which can clearly be distinguished by the context.

The binary-state neurons $\varrho_0, \varrho_1, \varrho_2, \varrho_3 \in V_1$ implement a finite automaton $\mathcal{A}_0$ that recognizes the regular language $R_0 = \Gamma'' \cdot (\Gamma_\lambda)^* \cdot \Gamma_\sigma \cup \Gamma_\sigma$ iteratively within $(\mathcal{L} \cap R_0)^* \cdot \mathcal{L}$ by using the following weights and biases:

$$w(i, \varrho_0) = w(i, \varrho_3) = 1 \quad \text{for } i \in X \text{ s.t. } \lambda_i \in \Gamma_\sigma \tag{85}$$
$$w(i, \varrho_1) = 1 \quad \text{for } i \in X \text{ s.t. } \lambda_i \in \Gamma'' \tag{86}$$
$$w(i, \varrho_2) = 1 \quad \text{for } i \in X \text{ s.t. } \lambda_i \in \Gamma_\lambda \tag{87}$$
$$w(\varrho_0, \varrho_0) = w(\varrho_3, \varrho_0) = 1 \tag{88}$$
$$w(\varrho_0, \varrho_1) = w(\varrho_3, \varrho_1) = 1 \tag{89}$$
$$w(\varrho_1, \varrho_2) = w(\varrho_2, \varrho_2) = 1 \tag{90}$$
$$w(\varrho_1, \varrho_3) = w(\varrho_2, \varrho_3) = 1 \tag{91}$$
$$w(0, \varrho_0) = w(0, \varrho_1) = w(0, \varrho_2) = w(0, \varrho_3) = -2\,. \tag{92}$$

The initial state of $\mathcal{N}$ is defined so that at the beginning, the state of ϱ_0 is activated, which means $y_{\varrho_0}^{(0)} = 1$, whereas $y_{\varrho_1}^{(0)} = y_{\varrho_2}^{(0)} = y_{\varrho_3}^{(0)} = 0$. This single activation traverses the units $\varrho_0, \varrho_1, \varrho_2, \varrho_3 \in V_1$ whose states are updated at time instants $4\tau + 1$ during the macroscopic steps $\tau \geq 0$, while they indicate whether the current input symbol $z_{\tau+1} \in \Gamma$ is from $\Gamma_\sigma, \Gamma'', \Gamma_\lambda, \Gamma_\sigma$, respectively, as a part of a string from R_0:

$$y_{\varrho_0}^{(4\tau+1)} = 1 \quad \text{iff} \quad z_{\tau+1} \in \Gamma_\sigma \text{ and either } \tau = 0 \text{ or } z_\tau \in \Gamma_\sigma \tag{93}$$
$$y_{\varrho_1}^{(4\tau+1)} = 1 \quad \text{iff} \quad z_{\tau+1} \in \Gamma'' \text{ and either } \tau = 0 \text{ or } z_\tau \in \Gamma_\sigma \tag{94}$$
$$y_{\varrho_2}^{(4\tau+1)} = 1 \quad \text{iff} \quad z_{\tau+1} \in \Gamma_\lambda \text{ and either } z_\tau \in \Gamma'' \text{ or } z_\tau \in \Gamma_\lambda \tag{95}$$
$$y_{\varrho_3}^{(4\tau+1)} = 1 \quad \text{iff} \quad z_{\tau+1} \in \Gamma_\sigma \text{ and either } z_\tau \in \Gamma_\lambda \text{ or } z_\tau \in \Gamma''\,. \tag{96}$$

The conditions (93)–(96) are proven by induction on $\tau \geq 0$. We verify (93), whereas the argument for (94)–(96) is analogous. Recall that at most one neuron $i \in X$ such that $\lambda_i \in \Gamma_\sigma$ can be activated due to (5). Thus, $y_{\varrho_0}^{(4\tau+1)} = 1$ iff

$$\begin{aligned} \xi_{\varrho_0}^{(4\tau)} &= w(0, \varrho_0) + \sum_{i \in X \,:\, \lambda_i \in \Gamma_\sigma} w(i, \varrho_0) y_i^{(4\tau)} + w(\varrho_0, \varrho_0) y_{\varrho_0}^{(4\tau)} \\ &\quad + w(\varrho_3, \varrho_0) y_{\varrho_3}^{(4\tau)} \geq 0 \end{aligned} \tag{97}$$

iff $\sum_{i \in X \,:\, \lambda_i \in \Gamma_\sigma} y_i^{(4\tau)} + y_{\varrho_0}^{(4\tau)} + y_{\varrho_3}^{(4\tau)} \geq 2$ iff $z_{\tau+1} \in \Gamma_\sigma$ and either $\tau = 0$ when we know $y_{\varrho_0}^{(4\tau)} = 1$ or either $y_{\varrho_0}^{(4(\tau-1)+1)} = 1$ or $y_{\varrho_3}^{(4(\tau-1)+1)} = 1$ iff $z_{\tau+1} \in \Gamma_\sigma$ and either $\tau = 0$ or $z_\tau \in \Gamma_\sigma$ by using the induction hypothesis for (93) or (96), which completes the proof of (93). Finally, the neuron $\text{rej}_0 \in V_2$ represents the reject state of finite automaton $\mathcal{A}_0$, which is implemented by the weights and bias

$$\begin{aligned} w(\varrho_i, \mathrm{rej}_0) &= -1 \quad \text{for } i = 1, 2, 3, 4 & (98)\\ w(0, \mathrm{rej}_0) &= 0\,. & (99) \end{aligned}$$

It follows that

$$y_{\mathrm{rej}_0}^{(4\tau+2)} = 1 \quad \text{iff} \quad y_{\varrho_i}^{(4\tau+1)} = 0 \text{ for every } i = 1, 2, 3, 4\,. \tag{100}$$

For each $\tau \geq 0$ and $r \in \{1, \ldots, p+1\}$, the binary output $y_{H_r}^{(4\tau+1)} \in \{0, 1\}$ from neuron $H_r \in V_1$ at time instant $4\tau + 1$ indicates whether the state $y_s^{(4\tau)} \in \mathbb{I}$ of analog unit $s \in V_4$ at macroscopic time τ, falls into the corresponding half-line H_r (cf. (28)), that is,

$$y_{H_r}^{(4\tau+1)} = \left\{ \begin{array}{ll} 1 & \text{if } y_s^{(4\tau)} \in H_r \\ 0 & \text{if } y_s^{(4\tau)} \notin H_r \end{array} \right. \quad \text{where } H_r = \left\{ \begin{array}{ll} [c_r, \infty) & \text{if } s_r = -1 \\ (-\infty, c_r] & \text{if } s_r = 1\,. \end{array} \right. \tag{101}$$

This is implemented by the weights and biases of units $H_r \in V_1$,

$$w(s, H_r) = -s_r\,, \quad w(0, H_r) = s_r c_r \qquad \text{for } r = 1 \ldots, p+1\,, \tag{102}$$

since $y_{H_r}^{(4\tau+1)} = 1$ iff $\xi_{H_r}^{(4\tau)} = w(0, H_r) + w(s, H_r) y_s^{(4\tau)} \geq 0$ iff $s_r c_r - s_r y_s^{(4\tau)} = s_r \left(c_r - y_s^{(4\tau)} \right) \geq 0$ iff either $s_r = -1$ and $y_s^{(4\tau)} \geq c_r$, or $s_r = 1$ and $y_s^{(4\tau)} \leq c_r$ iff $y_s^{(4\tau)} \in H_r$.

By using the intervals I_r introduced in (40), we define the weights and biases of neurons $I_r \in V_2$ as

$$\begin{aligned} w(i, I_r) &= 1 \quad \text{for } i \in X \text{ s.t. } \lambda_i \in \Gamma_r \\ w(H_r, I_r) &= -s_r\,, \quad w(H_{r+1}, I_r) = s_{r+1} \qquad \text{for } r = 1 \ldots, p\,. \\ w(0, I_r) &= \frac{s_r - s_{r+1}}{2} - 2 \end{aligned} \tag{103}$$

We show that

$$y_{I_r}^{(4\tau+2)} = 1 \quad \text{iff} \quad y_s^{(4\tau)} \in I_r \text{ and } z_{\tau+1} \in \Gamma_r\,. \tag{104}$$

Recall that at most one neuron $i \in X$ such that $\lambda_i \in \Gamma_r$ can be activated due to (5). We have $y_{I_r}^{(4\tau+2)} = 1$ iff $\xi_{I_r}^{(4\tau+1)} = w(0, I_r) + \sum_{i \in X\,:\,\lambda_i \in \Gamma_r} w(i, I_r) y_i^{(4\tau+1)} +$

$w(H_r, I_r)y_{H_r}^{(4\tau+1)} + w(H_{r+1}, I_r)y_{H_{r+1}}^{(4\tau+1)} \geq 0$ iff $\frac{s_r - s_{r+1}}{2} - 2 + \sum_{i \in X\,:\,\lambda_i \in \Gamma_r} y_i^{(4\tau)} - s_r y_{H_r}^{(4\tau+1)} + s_{r+1} y_{H_{r+1}}^{(4\tau+1)} \geq 0$. For example, consider the case when $s_r = -1$ and $s_{r+1} = -1$, which means $H_r = [c_r, \infty)$, $H_{r+1} = [c_{r+1}, \infty)$, and $I_r = [c_r, c_{r+1})$, while the argument in the remaining cases is similar. Thus, $y_{I_r}^{(4\tau+2)} = 1$ iff $\sum_{i \in X\,:\,\lambda_i \in \Gamma_r} y_i^{(4\tau)} + y_{H_r}^{(4\tau+1)} - y_{H_{r+1}}^{(4\tau+1)} \geq 2$ iff $z_{\tau+1} \in \Gamma_r$ and $y_{H_r}^{(4\tau+1)} = 1$ and $y_{H_{r+1}}^{(4\tau+1)} = 0$ iff $z_{\tau+1} \in \Gamma_r$ and $y_s^{(4\tau)} \in H_r \setminus H_{r+1} = I_r$.

Since the intervals I_r for $r = 1, \ldots, p$, create the partition of the unit interval $\mathbb{I}$, at most one neuron among the units $I_r \in V_2$ is activated according to (104). The neuron $\mathrm{rej} \in V_3$ activates and remains activated if no neuron $I_r \in V_2$ is activated and, at the same time, the current input symbol is either from Γ_λ or from Γ_σ preceded by the symbol from $\Gamma_\lambda \cup \Gamma''$. In addition, $\mathrm{rej} \in V_3$ also activates when the unit $\mathrm{rej}_0 \in V_2$ is activated. Thus, $y_{\mathrm{rej}}^{(t)} = 1$ for all $t \geq 4\tau + 3$ iff $y_{I_r}^{(4\tau+2)} = 0$ for every $r = 1, \ldots, p$, and either $y_{\varrho_2}^{(4\tau+1)} = 1$ or $y_{\varrho_3}^{(4\tau+1)} = 1$, or $y_{\mathrm{rej}_0}^{(4\tau+2)} = 1$. This is implemented by the weights and bias of unit $\mathrm{rej} \in V_3$,

$$
\begin{aligned}
w(I_r, \mathrm{rej}) &= -1 \quad \text{for } r = 1 \ldots, p && (105)\\
w(\varrho_0, \mathrm{rej}) &= w(\varrho_1, \mathrm{rej}) = -1 && (106)\\
w(\mathrm{rej}_0, \mathrm{rej}) &= 1 && (107)\\
w(\mathrm{rej}, \mathrm{rej}) &= 2 && (108)\\
w(0, \mathrm{rej}) &= 0\,. && (109)
\end{aligned}
$$

Clearly, $y_{\mathrm{rej}}^{(4\tau+3)} = 1$ iff

$$
\begin{aligned}
\xi_{\mathrm{rej}}^{(4\tau+2)} = {} & w(0, \mathrm{rej}) + \sum_{r=1}^{p} w(I_r, \mathrm{rej}) y_{I_r}^{(4\tau+2)} + w(\varrho_0, \mathrm{rej}) y_{\varrho_0}^{(4\tau+2)} \\
& + w(\varrho_1, \mathrm{rej}) y_{\varrho_1}^{(4\tau+2)} + w(\mathrm{rej}_0, \mathrm{rej}) y_{\mathrm{rej}_0}^{(4\tau+2)} \\
& + w(\mathrm{rej}, \mathrm{rej}) y_{\mathrm{rej}}^{(4\tau+2)} \geq 0 \qquad (110)
\end{aligned}
$$

iff $\sum_{r=1}^{p} y_{I_r}^{(4\tau+2)} + y_{\varrho_0}^{(4\tau+1)} + y_{\varrho_1}^{(4\tau+1)} - y_{\mathrm{rej}_0}^{(4\tau+2)} - 2y_{\mathrm{rej}}^{(4(\tau-1)+3)} \leq 0$ iff $y_{I_r}^{(4\tau+2)} = 0$ for every $r = 1, \ldots, p$, and $y_{\varrho_0}^{(4\tau+1)} = y_{\varrho_1}^{(4\tau+1)} = 0$, or $y_{\mathrm{rej}_0}^{(4\tau+2)} = 1$ or $y_{\mathrm{rej}}^{(4(\tau-1)+3)} = 1$. Recall that $y_{\varrho_0}^{(4\tau+1)} = y_{\varrho_1}^{(4\tau+1)} = y_{\mathrm{rej}_0}^{(4\tau+2)} = 0$ implies either $y_{\varrho_2}^{(4\tau+1)} = 1$ or $y_{\varrho_3}^{(4\tau+1)} = 1$. Moreover, we know $\sum_{r=1}^{p} y_{I_r}^{(4\tau+2)} + y_{\varrho_0}^{(4\tau+1)} + y_{\varrho_1}^{(4\tau+1)} \leq 2$ which ensures $y_{\mathrm{rej}}^{(t)} = 1$ for all $t \geq 4\tau + 3$, once $y_{\mathrm{rej}}^{(4\tau+3)} = 1$.

The neuron rej $\in V_3$ is connected to the output neuron out $\in V_4$ via the dominating negative weight $w(\mathrm{rej},\mathrm{out}) = -W - 1$ where

$$W = \sum_{i \in V'} |w(i,\mathrm{out})| \;, \tag{111}$$

which ensures $y_{\mathrm{out}}^{(4(\tau+1))} = 0$ whenever $y_{\mathrm{rej}}^{(4\tau+3)} = 1$. Finally, we define the weights and bias of analog neuron $s \in V_4$ as

$$\begin{aligned} w(i,s) &= \alpha(\lambda_i)\beta^{-1} \quad \text{for } i \in X \text{ s.t. } \lambda_i \in \Gamma_\lambda \cup \Gamma'' & (112)\\ w(s,s) &= \beta^{-1} & (113)\\ w(I_p,s) &= -\beta^{-1} & (114)\\ w(0,s) &= 0\,, & (115)\end{aligned}$$

which completes the definition of 1ANN $\mathcal{N}$.

We show that $L = \mathcal{L}(\mathcal{N})$. Let $z_1 \ldots z_n \in \Gamma^*$ be an input string to $\mathcal{N}$. Denote by $1 \le \tau_1 < \tau_2 < \cdots < \tau_m \le n$ all the indices such that $z_{\tau_\ell} \in \Gamma_\sigma = \Gamma_1 \cup \Gamma_p$ for $\ell = 1,\ldots,m$, and formally define $\tau_0 = 0$ and $\tau_{m+1} = n$. We proceed by induction on $\ell \ge 0$ up to $\ell = m$. Assume that $y_{\mathrm{rej}}^{(4\tau_\ell)} = 0$, $y_s^{(4\tau_\ell)} = 0$, and either $y_{\varrho_0}^{(4\tau_\ell)} = 1$ or $y_{\varrho_3}^{(4\tau_\ell)} = 1$, which holds for $\ell = 0$. We prove that $z_{\tau_\ell+1} z_{\tau_\ell+2} \ldots z_{\tau_{\ell+1}} \in \mathcal{L} \cap R_0$ iff $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 0$. According to (93)–(96), and (100), we already know that $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \notin R_0$ iff $y_{\mathrm{rej}_0}^{(4(\tau_{\ell+1}-1)+2)} = 1$ implying $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 1$. Thus, assume $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in R_0$ which ensures $y_{\mathrm{rej}_0}^{(4\tau)} = 0$ for every $\tau = \tau_\ell, \ldots, \tau_{\ell+1}$, and it suffices to show that $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in \mathcal{L}$ iff $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 0$.

First consider the case when $\tau_\ell + 1 = \tau_{\ell+1}$, which means $z_{\tau_{\ell+1}} \in \Gamma_\sigma \subset \mathcal{L} \cap R_0$. We have $y_{\varrho_0}^{(4\tau_\ell+1)} = 1$ by (93). If $z_{\tau_{\ell+1}} \in \Gamma_1$, then $y_{I_1}^{(4\tau_\ell+2)} = 1$ due to $0 = y_s^{(4\tau_\ell)} \in I_1 = [0,0]$, which implies $\xi_{\mathrm{rej}}^{(4\tau_\ell+2)} = w(I_1,\mathrm{rej})y_{I_1}^{(4\tau_\ell+2)} + w(\varrho_0,\mathrm{rej})y_{\varrho_0}^{(4\tau_\ell+1)} = -2$ according to (105)–(110), and hence $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 0$. If $z_{\tau_{\ell+1}} \in \Gamma_p$, then $y_{I_p}^{(4\tau_\ell+2)} = 0$ due to $0 = y_s^{(4\tau_\ell)} \notin I_p = [1,1]$, which implies $\xi_{\mathrm{rej}}^{(4\tau_\ell+2)} = w(\varrho_0,\mathrm{rej})y_{\varrho_0}^{(4\tau_\ell+1)} = -1$, and hence, $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 0$. In addition, $\xi_s^{(4\tau_\ell+3)} = w(0,s) + \sum_{i\in X\,:\,\lambda_i\in\Gamma_\lambda\cup\Gamma''} w(i,s)y_i^{(4\tau_\ell)} + w(s,s)y_s^{(4\tau_\ell)} + w(I_p,s)y_{I_p}^{(4\tau_\ell+2)} = 0$ according to (112)–(115), because $y_i^{(4\tau_\ell)} = 0$ for all $i \in X$ due to $z_{\tau_\ell+1} \notin \Gamma_\lambda \cup \Gamma''$, and $y_s^{(4\tau_\ell)} = y_{I_p}^{(4\tau_\ell+2)} = 0$. Hence, the assumption

$y_s^{(4\tau_{\ell+1})} = 0$ and $y_{\varrho_0}^{(4\tau_{\ell+1})} = 1$ is preserved for $\ell + 1$. This completes the proof that $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in \mathcal{L}$ iff $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 0$ for $\tau_\ell + 1 = \tau_{\ell+1}$.

Further consider the case when $\tau_\ell+1 < \tau_{\ell+1}$, which implies $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in \Gamma'' \cdot (\Gamma_\lambda)^* \cdot \Gamma_\sigma$ due to $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in R_0$. Hence, $y_{\varrho_1}^{(4\tau_\ell+1)} = y_{\varrho_3}^{(4(\tau_{\ell+1}-1)+1)} = 1$, and $y_{\varrho_2}^{(4\tau+1)} = 1$ for every $\tau = \tau_\ell + 1, \ldots, \tau_{\ell+1} - 2$. We show by induction on $\tau \geq \tau_\ell$ up to $\tau = \tau_{\ell+1} - 1$ that $z_{\tau_\ell+1} \ldots z_{\tau+1} \in \mathcal{L}$ iff $y_{\mathrm{rej}}^{(4\tau+3)} = 0$, and $\xi_s^{(4\tau+3)} = \sum_{k=1}^{\tau-\tau_\ell+1} \alpha(z_{\tau-k+2})\beta^{-k}$ for $z_{\tau_\ell+1} \ldots z_{\tau+1} \in \mathcal{L}$ such that $\tau < \tau_{\ell+1} - 1$. For $\tau = \tau_\ell$, we have $z_{\tau_\ell+1} \in \Gamma'' \subseteq \mathcal{L}$ which means $y_{I_r}^{(4\tau_\ell+2)} = 0$ for every $r = 1, \ldots, p$, by (104). According to (105)–(110), we thus have $\xi_{\mathrm{rej}}^{(4\tau_\ell+2)} = w(\varrho_1, \mathrm{rej}) y_{\varrho_1}^{(4\tau_\ell+1)} = -1$, and hence $y_{\mathrm{rej}}^{(4\tau_\ell+3)} = 0$. Moreover, $\xi_s^{(4\tau_\ell+3)} = w(0,s) + \sum_{i \in X : \lambda_i \in \Gamma_\lambda \cup \Gamma''} w(i,s) y_i^{(4\tau_\ell)} + w(s,s) y_s^{(4\tau_\ell)} + w(I_p, s) y_{I_p}^{(4\tau_\ell+2)} = \alpha(z_{\tau_\ell+1})\beta^{-1}$ by (112) due to $y_s^{(4\tau_\ell)} = 0$ and $y_{I_p}^{(4\tau_\ell+2)} = 0$.

In the induction step when $\tau > \tau_\ell$, we assume $z_{\tau_\ell+1} \ldots z_\tau \in \mathcal{L}$ iff $y_{\mathrm{rej}}^{(4(\tau-1)+3)} = 0$, and $\xi_s^{(4(\tau-1)+3)} = \sum_{k=1}^{\tau-\tau_\ell} \alpha(z_{\tau-k+1})\beta^{-k}$ for $z_{\tau_\ell+1} \ldots z_\tau \in \mathcal{L}$. If $z_{\tau_\ell+1} \ldots z_\tau \notin \mathcal{L}$, then $z_{\tau_\ell+1} \ldots z_{\tau+1} \notin \mathcal{L}$ due to $\mathcal{L}$ is prefix-closed by (31), and by induction hypothesis we have $y_{\mathrm{rej}}^{(4(\tau-1)+3)} = 1$ which ensures $y_{\mathrm{rej}}^{(4\tau+3)} = 1$. Thus, further assume $z_{\tau_\ell+1} \ldots z_\tau \in \mathcal{L}$ which means $y_{\mathrm{rej}}^{(4(\tau-1)+3)} = 0$ and $\xi_s^{(4(\tau-1)+3)} = \sum_{k=1}^{\tau-\tau_\ell} \alpha(z_{\tau-k+1})\beta^{-k}$. By using (4) we obtain $y_s^{(4\tau)} = \sigma\left(\xi_s^{(4(\tau-1)+3)}\right) \in I_r$ for some $r \in \{1, \ldots, p\}$, and hence, $z_\tau z_{\tau-1} \ldots z_{\tau_\ell+1} \in L_r$. It follows that $z_{\tau_\ell+1} \ldots z_{\tau+1} \in \mathcal{L}$ iff $z_{\tau_\ell+1} \ldots z_{\tau+1} \in L_r^R \cdot \Gamma_r$ iff $z_{\tau+1} \in \Gamma_r$ iff $y_{I_r}^{(4\tau+2)} = 1$ iff $y_{\mathrm{rej}}^{(4\tau+3)} = 0$ because $\xi_{\mathrm{rej}}^{(4\tau+2)} = -y_{I_r}^{(4\tau+2)}$ due to $y_{\varrho_0}^{(4\tau+1)} = y_{\varrho_2}^{(4\tau+1)} = y_{\mathrm{rej}_0}^{(4\tau+2)} = y_{\mathrm{rej}}^{(4(\tau-1)+3)} = 0$. For $z_{\tau_\ell+1} \ldots z_{\tau+1} \in \mathcal{L}$ such that $\tau < \tau_{\ell+1} - 1$, we know $z_{\tau+1} \in \Gamma_\lambda$, which ensures $0 < y_s^{(4\tau)} = \xi_s^{(4(\tau-1)+3)} < 1$ and $y_{I_p}^{(4\tau+2)} = 0$, and hence,

$$
\begin{aligned}
\xi_s^{(4\tau+3)} &= w(0,s) + \sum_{i \in X : \lambda_i \in \Gamma_\lambda \cup \Gamma''} w(i,s) y_i^{(4\tau)} + w(s,s) y_s^{(4\tau)} + w(I_p, s) y_{I_p}^{(4\tau+2)} \\
&= \alpha\left(z_{\tau+1}\right)\beta^{-1} + \beta^{-1} \sum_{k=1}^{\tau-\tau_\ell} \alpha(z_{\tau-k+1})\beta^{-k} = \sum_{k=1}^{\tau-\tau_\ell+1} \alpha(z_{\tau-k+2})\beta^{-k} \qquad (116)
\end{aligned}
$$

according to (112)–(115). This completes the induction on τ, which proves $z_{\tau_\ell+1} \ldots z_{\tau_{\ell+1}} \in \mathcal{L}$ iff $y_{\mathrm{rej}}^{(4(\tau_{\ell+1}-1)+3)} = 0$ also for $\tau_\ell + 1 < \tau_{\ell+1}$.

In addition, for $\tau = \tau_{\ell+1} - 1$, we know $z_{\tau_{\ell+1}} \in \Gamma_\sigma = \Gamma_1 \cup \Gamma_p$, which implies $y_{\varrho_3}^{(4(\tau_{\ell+1}-1)+1)} = 1$ and $\xi_s^{(4(\tau_{\ell+1}-1)+3)} = w(s,s) y_s^{(4(\tau_{\ell+1}-1))} + w(I_p, s) y_{I_p}^{(4(\tau_{\ell+1}-1)+2)}$.

If $z_{\tau_{\ell+1}} \in \Gamma_1$, then $y_s^{(4(\tau_{\ell+1}-1))} = 0$ and $y_{I_p}^{(4(\tau_{\ell+1}-1)+2)} = 0$, which gives $\xi_s^{(4(\tau_{\ell+1}-1)+3)} = 0$. If $z_{\tau_{\ell+1}} \in \Gamma_p$, then $y_s^{(4(\tau_{\ell+1}-1))} = 1$ and $y_{I_p}^{(4(\tau_{\ell+1}-1)+2)} = 1$, which also produces $\xi_s^{(4(\tau_{\ell+1}-1)+3)} = \beta^{-1} - \beta^{-1} = 0$. It follows that $y_s^{(4\tau_{\ell+1})} = 0$ which together with $y_{\varrho_3}^{(4\tau_{\ell+1})} = 1$, preserves the assumption for $\ell + 1$.

Thus by induction on $\ell = 1, \ldots, m$, we obtain $z_1 \ldots z_n \in (\mathcal{L} \cap R_0)^* \cdot \mathcal{L}$ iff $y_{\mathrm{rej}}^{(4(n-1)+3)} = 0$, which implies $z_1 \ldots z_n \in L = (\mathcal{L} \cap R_0)^* \cdot \mathcal{L} \cap R$ iff $y_{\mathrm{out}}^{(4n)} = 1$. Hence, $L = \mathcal{L}(\mathcal{N})$. □

6. 1ANNs Within the Chomsky Hierarchy

In this section, we analyze the computational power of 1ANNs by using the representation Theorem 4 and the classification of cut languages within the Chomsky hierarchy presented in Paragraph 3.3. We first formulate a sufficient condition when a 1ANN with real weights recognizes a regular language.

Theorem 6 *Let $\mathcal{N}$ be a 1ANN such that $0 < |w_{ss}| < 1$. Define the base, $\beta_{\mathcal{N}} = \frac{1}{w_{ss}}$, the digit alphabet,*

$$A_{\mathcal{N}} = \left\{ -c_s\left(\tilde{\mathbf{y}}\right) \,\middle|\, \tilde{\mathbf{y}} \in \{0,1\}^{s-1} \right\} \cup \{0, \beta_{\mathcal{N}}\}\,, \tag{117}$$

and the set of thresholds,

$$C_{\mathcal{N}} = \left\{ c_j\left(\tilde{\mathbf{y}}\right) \in \mathbb{I} \,\middle|\, j \in V \setminus (X \cup s) : w_{js} \neq 0\,,\, \tilde{\mathbf{y}} \in \{0,1\}^{s-1} \right\} \cup \{0,1\} \tag{118}$$

where $c_j\left(\tilde{\mathbf{y}}\right)$ for $j \in V \setminus X$ such that $w_{js} \neq 0$, is defined in (23). If every threshold $c \in C_{\mathcal{N}}$ is a $\beta_{\mathcal{N}}$-quasi-periodic number within $A_{\mathcal{N}}$, then the language $L = \mathcal{L}(\mathcal{N})$ recognized by $\mathcal{N}$, is regular.

Example 1 (continuing from p. 20) We illustrate the statement of Theorem 6 on the running example of the 1ANN language acceptor $\mathcal{N}$ which is depicted in Figure 1. By plugging the corresponding weights of $\mathcal{N}$ into (117) and (118) we obtain $\beta_{\mathcal{N}} = 1/w_{55} = 3$, $A_{\mathcal{N}} = \{0,2,3\}$, and $C_{\mathcal{N}} = \left\{0, \frac{1}{2}, 1\right\}$, cf. (35) and (36), respectively. We know $\beta_{\mathcal{N}} = 3 \in \mathbb{Z}$ is a Pisot number since every integer greater than 1 is Pisot. In addition, the digit alphabet is contained in the field extension[3] $\mathbb{Q}(3)$ due to $A_{\mathcal{N}} = \{0,2,3\} \subset \mathbb{Z} \subset \mathbb{Q} = \mathbb{Q}(3)$, and similarly, the set of thresholds meets $C_{\mathcal{N}} = \left\{0, \frac{1}{2}, 1\right\} \subset \mathbb{Q} = \mathbb{Q}(3)$. Hence, every $c \in C_{\mathcal{N}}$ is a $\beta_{\mathcal{N}}$-quasi-periodic number within $A_{\mathcal{N}}$, according to Theorem 1. Thus, the assumption of Theorem 6 is satisfied, which implies that the language $L = \mathcal{L}(\mathcal{N})$ recognized by $\mathcal{N}$ is regular. Indeed, the 1ANN $\mathcal{N}$ recognizes the language

$$L = \mathcal{L}(\mathcal{N}) = \{x_1 \ldots x_n \in \{0,1\}^* \mid n \leq 1 \text{ or } x_{n-1} = 0\} \tag{119}$$

over the binary alphabet $\Sigma = \{0,1\}$ that (apart from $\varepsilon, 0, 1 \in L$) contains all the words whose next-to-last symbol is 0, which is regular. It is because the output of the analog unit meets

$$y_5^{(2\tau+2)} = \sigma\left(\tfrac{2}{3}\, y_2^{(2\tau)} + \tfrac{1}{3}\, y_5^{(2\tau)}\right) \left\{ \begin{array}{ll} \geq \frac{2}{3} & \text{if } y_2^{(2\tau)} = 1 \\ \leq \frac{1}{3} & \text{if } y_2^{(2\tau)} = 0 \end{array} \right. \tag{120}$$

for every $\tau \geq 0$, due to $y_5^{(2\tau)} \in \mathbb{I}$, which means $y_2^{(2\tau)} = 0$ iff $y_5^{(2\tau+2)} < \frac{1}{2}$ iff $y_4^{(2\tau+3)} = 0$ iff $y_3^{(2\tau+4)} = 1$.

Proof (Theorem 6). According to Theorem 4, we can write the language $L = \mathcal{L}(\mathcal{N})$ recognized by 1ANN $\mathcal{N}$ in the form (30). Let $\Gamma_0 = \{\gamma_0, \gamma_1\}$ be composed of two new letters not contained in Γ. We define the homomorphism $g : \Gamma^* \longrightarrow (\Gamma' \cup \Gamma_0)^*$ so that for every $z \in \Gamma = \Gamma' \cup \Gamma''$,

$$g(z) = \left\{ \begin{array}{ll} z & \text{if } z \in \Gamma' \\ z_1\, z_2\, g'(z_2) & \text{if } z = (z_1, z_2) \in \Gamma'' = \Gamma_\lambda \times \Gamma_\sigma \end{array} \right. \tag{121}$$

where $g' : \Gamma_\sigma \longrightarrow \Gamma_0$ such that

$$g'(z_2) = \left\{ \begin{array}{ll} \gamma_0 & \text{if } z_2 \in \Gamma_1 \\ \gamma_1 & \text{if } z_2 \in \Gamma_p\,. \end{array} \right. \tag{122}$$

Observe that for any $c \in C_{\mathcal{N}}$, the homomorphism g generates the cut language

$$L'_{<c} = g\left(L_{<c}\right) = \left\{ z'_1 \ldots z'_n \in \Gamma_\lambda^* \cdot \Gamma_\lambda \cdot \Gamma_\sigma \cdot \Gamma_0 \,\left|\, \sum_{k=1}^{n} \alpha'\left(z'_k\right) \beta_{\mathcal{N}}^{-k} < c \right.\right\} \tag{123}$$

over the alphabet $\Gamma' \cup \Gamma_0$ (similarly $L'_{>c}$) from the cut language (33), where $\alpha' : \Gamma' \cup \Gamma_0 \longrightarrow A_{\mathcal{N}}$ extends (34) as

$$\alpha'(z) = \left\{ \begin{array}{ll} \alpha(z) & \text{if } z \in \Gamma' \\ 0 & \text{if } z = \gamma_0 \in \Gamma_0 \\ \beta_{\mathcal{N}} & \text{if } z = \gamma_1 \in \Gamma_0\,, \end{array} \right. \tag{124}$$

such that for any $\mathbf{z} \in \Gamma_\lambda^* \cdot \Gamma''$ it holds $\mathbf{z} \in L_{<c}$ iff $g(\mathbf{z}) \in L'_{<c}$. In particular, for any $\mathbf{z} = z_1 \ldots z_{n-1}(z_{n1}, z_{n2}) \in \Gamma_\lambda^* \cdot \Gamma''$, we have $\mathbf{z} \in L_{<c}$ iff $\sum_{k=1}^{n-1} \alpha(z_k)\beta_{\mathcal{N}}^{-k} + \alpha\big((z_{n1}, z_{n2})\big)\beta_{\mathcal{N}}^{-n} < c$ iff

$$\sum_{k=1}^{n-1} \alpha'(z_k)\beta_{\mathcal{N}}^{-k} + \alpha'(z_{n1})\beta_{\mathcal{N}}^{-n} + \alpha'(z_{n2})\beta_{\mathcal{N}}^{-n-1} + \alpha'\left(g'(z_{n2})\right)\beta_{\mathcal{N}}^{-n-2} < c \quad (125)$$

by (34), (122), and (124) iff $g(\mathbf{z}) \in L'_{<c}$ due to (121).

Assume that every $c \in C_{\mathcal{N}}$ is $\beta_{\mathcal{N}}$-quasi-periodic number within $A_{\mathcal{N}}$. According to Theorem 2, the cut language $L'_{<c}$ over alphabet $\Gamma' \cup \Gamma_0$ such that $\alpha'(\Gamma' \cup \Gamma_0) \subseteq A_{\mathcal{N}}$, is regular for any $c \in C_{\mathcal{N}}$, and hence, $L_{<c} = g^{-1}\left(L'_{<c}\right)$ is regular by (123) since regular languages are closed under inverse homomorphism. This ensures that the languages $L_1, \ldots, L_p$ defined in (32) are also regular because regular languages are closed under complement and intersection. Furthermore, regular languages are known to be closed under reversal, concatenation, union, Kleene star, and homomorphism. In addition, if S is regular, then its largest prefix-closed subset S^{Pref} is also regular as a corresponding finite automaton $\mathcal{A}_1$ recognizing $S = \mathcal{L}(\mathcal{A}_1)$ can be reduced to $\mathcal{A}_2$ such that $S^{Pref} = \mathcal{L}(\mathcal{A}_2)$, by eliminating all the non-final states in $\mathcal{A}_1$. It follows that the language $L = \mathcal{L}(\mathcal{N})$ in (30) defined using (33), is regular. □

As a consequence of Theorems 1 and 6, we obtain that any 1ANN whose inverse of the self-loop weight of the analog unit is a Pisot number β, while all its weights are in the smallest field extension over rational numbers including β, is equivalent to a finite automaton.

Corollary 1 *Let $\mathcal{N}$ be a 1ANN such that $0 < |w_{ss}| < 1$. If $\beta_{\mathcal{N}} = 1/w_{ss}$ is a Pisot number and $w_{ji} \in \mathbb{Q}(\beta_{\mathcal{N}})$ for every $j = 1, \ldots, s$ and $i = 0, \ldots, s$, then $\mathcal{L}(\mathcal{N})$ is regular.*

PROOF. It follows from the assumption that $c_j(\tilde{\mathbf{y}}) \in \mathbb{Q}(\beta_{\mathcal{N}})$ for every $\tilde{\mathbf{y}} \in \{0,1\}^{s-1}$ and $j \in V \setminus X$ such that $w_{js} \neq 0$, due to (23). Hence, $A_{\mathcal{N}} \subset \mathbb{Q}(\beta_{\mathcal{N}})$ and $C_{\mathcal{N}} \subset \mathbb{Q}(\beta_{\mathcal{N}})$ by (117) and (118), respectively. According to Theorem 1, we thus have that every threshold $c \in C_{\mathcal{N}}$ is a $\beta_{\mathcal{N}}$-quasi-periodic number within $A_{\mathcal{N}}$ since $\beta_{\mathcal{N}}$ is a Pisot number, which implies that $\mathcal{L}(\mathcal{N})$ is regular by Theorem 6. □

Example 1 (continuing from p. 35) In order to illustrate Corollary 1, we use it for another verification that the 1ANN language acceptor $\mathcal{N}$ from the

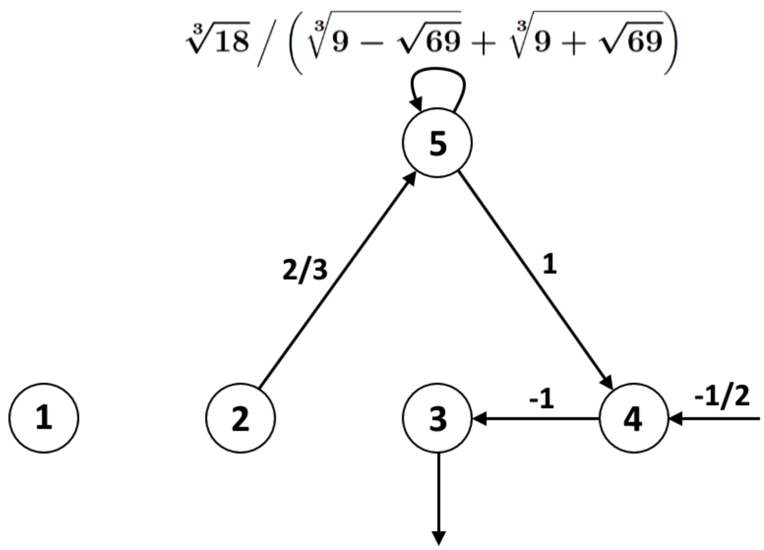

Figure 3: Example of a 1ANN that recognizes a regular language.

running example (Figure 1) recognizes a regular language. Clearly, the 1ANN $\mathcal{N}$ has only rational weights including the self-loop weight $w_{55} = \frac{1}{3}$ whose inverse is an integer $\beta_{\mathcal{N}} = 3 \in \mathbb{Z}$. Thus, $\beta_{\mathcal{N}} = 3$ is a Pisot number and the weights of $\mathcal{N}$ are in the field extension[3] $\mathbb{Q}(3) = \mathbb{Q}$. Hence, Corollary 1 confirms that $\mathcal{L}(\mathcal{N})$ is regular, which we know from (119).

In general, if a 1ANN $\mathcal{N}$ has rational weights $w_{ji} \in \mathbb{Q}$ for every $j = 1, \ldots, s$ and $i = 0, \ldots, s$, including the self-loop weight of the analog unit, $w_{ss} = 1/\beta_{\mathcal{N}}$ that is the inverse of some integer $\beta_{\mathcal{N}} > 1$ (a Pisot number), then $\mathcal{N}$ recognizes a regular language. This remains valid even if w_{ss} is the inverse of a non-integer Pisot which is not rational. For instance, the plastic constant $\rho = \left(\sqrt[3]{9-\sqrt{69}}+\sqrt[3]{9+\sqrt{69}}\right)/\sqrt[3]{18} \approx 1.324718$ (cf. Example 3) can be employed. Namely, if the rational weight $w_{55} = \frac{1}{3}$ in Figure 1 is replaced by $w_{55} = 1/\rho \approx 0.754878$ as depicted in Figure 3, then this modified 1ANN acceptor still recognizes only a regular language by Corollary 1, which, at the first sight, is not as clear as it was for the original $\mathcal{N}$ in (119).

On the other hand, 1ANNs with rational weights appear to be computationally more powerful than finite automata. We prove that they can even recognize languages that are not context-free. For this purpose, we first show in the following lemma how 1ANNs recognize single cut languages.

Lemma 1 *Let $\Gamma \neq \emptyset$ be a finite alphabet representing the digits in A through the mapping $\alpha : \Gamma \longrightarrow A$, and let $c \in \mathbb{R}$. In addition, assume that*

$$\mu = \inf_{z_1 \ldots z_n \in \Gamma^*} \sum_{k=1}^{n} \alpha(z_k) \beta^{-k} \geq 0\,. \tag{126}$$

Then the language $L = L_{<c}^R \cdot \Gamma$ where $L_{<c}$ is a cut language over alphabet Γ, can be recognized by a 1ANN.

Proof. Denote

$$\nu = \sup_{z_1 \ldots z_n \in \Gamma^*} \sum_{k=1}^{n} \alpha(z_k) \beta^{-k} \tag{127}$$

which is finite due to $|\beta| > 1$. Further assume $0 \leq \mu < c \leq \nu$ since otherwise $L_{<c} = \emptyset$ or $L_{<c} = \Gamma^*$ which can trivially be recognized by a 1ANN.

We will construct a 1ANN $\mathcal{N}$ that recognizes the language $\mathcal{L}(\mathcal{N}) = L_{<c}^R \cdot \Gamma$ within the time overhead $d = 2$ for processing one input symbol. The set of neurons V_t that are updated at time instant $t \geq 1$, satisfy $V_t = V_{t+2}$ for every $t \geq 1$. Apart from the input neurons $X \subseteq V$ and the output neuron out $\in V_2$ which implement the input/output protocol (5) and (6), respectively, we have the analog unit $s \in V_2$ and one binary neuron cut $\in V_1$. We define the corresponding weights as follows:

$$\begin{aligned}
w(s, \mathrm{cut}) &= 1 && (128)\\
w(0, \mathrm{cut}) &= -\frac{c}{\nu} && (129)\\
w(i, s) &= \frac{\alpha(\lambda_i)}{\beta \nu} \quad \text{for } i \in X \text{ s.t. } \lambda_i \in \Gamma && (130)\\
w(s, s) &= \beta^{-1} && (131)\\
w(0, s) &= 0 && (132)\\
w(\mathrm{cut}, \mathrm{out}) &= -1 && (133)\\
w(0, \mathrm{out}) &= 0 && (134)
\end{aligned}$$

The network $\mathcal{N}$ is schematically depicted in Figure 4.

Let $z_1 \ldots z_n \in \Gamma^*$ be an input to $\mathcal{N}$. We show by induction on $\tau \geq 0$ that

$$y_s^{(2\tau)} = \frac{1}{\nu} \sum_{k=1}^{\tau} \alpha(z_{\tau-k+1}) \beta^{-k}\,. \tag{135}$$

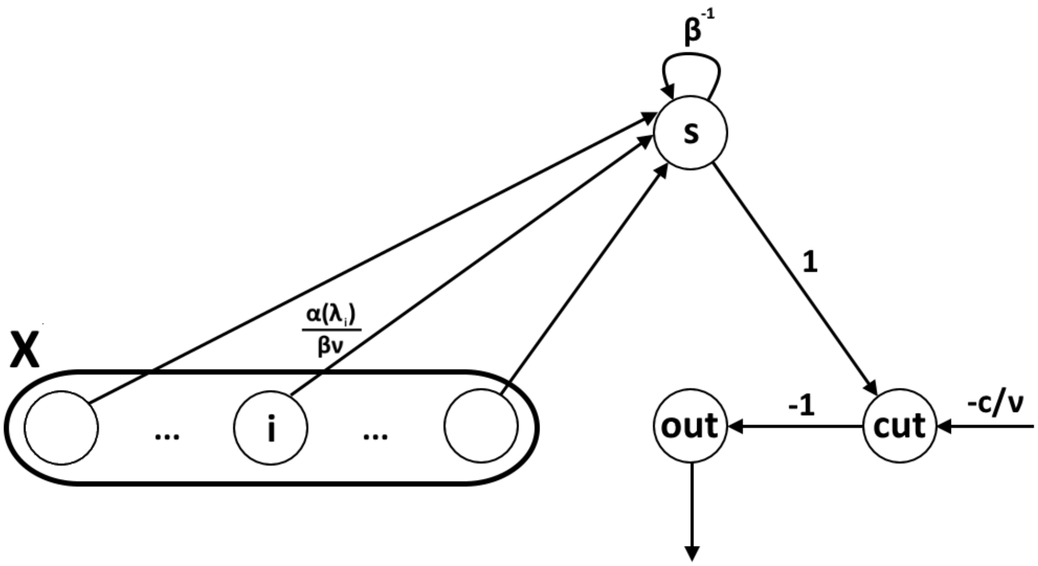

Figure 4: A schema of the neural network $\mathcal{N}$ that accepts the language $L^R_{<c} \cdot \Gamma$.

For $\tau = 0$, we assume the initial states $y_s^{(0)} = 0$, $y_{\mathrm{cut}}^{(0)} = 0$, and $y_{\mathrm{out}}^{(0)} = 1$. According to the definition of weights (130)–(132), for $\tau > 0$, we have

$$\xi_s^{(2(\tau-1))} = w(0,s) + \sum_{i \in X \,:\, \lambda_i \in \Gamma} w(i,s) y_i^{(2(\tau-1))} + w(s,s) y_s^{(2(\tau-1))}$$

$$= \frac{\alpha(z_\tau)}{\beta\nu} + \beta^{-1} \cdot \frac{1}{\nu} \sum_{k=1}^{\tau-1} \alpha(z_{\tau-k}) \beta^{-k} = \frac{1}{\nu} \sum_{k=1}^{\tau} \alpha(z_{\tau-k+1}) \beta^{-k} \qquad (136)$$

by the induction hypothesis. It follows from (126), (127), and (136) that $0 \le \xi_s^{(2(\tau-1))} \le 1$, and hence, $y_s^{(2\tau)} = \sigma\left(\xi_s^{(2(\tau-1))}\right) = \xi_s^{(2(\tau-1))}$ by (4), which completes the proof of (135).

In particular, for $\tau = n-1$, we obtain

$$y_s^{(2(n-1))} = \frac{1}{\nu} \sum_{k=1}^{n-1} \alpha(z_{n-k}) \beta^{-k} \qquad (137)$$

from (135). According to (128) and (129), we have

$$\xi_{\mathrm{cut}}^{(2(n-1))} = w(0,\mathrm{cut}) + w(s,\mathrm{cut})y_s^{(2(n-1))} = \frac{1}{\nu}\left(\sum_{k=1}^{n-1}\alpha(z_{n-k})\beta^{-k} - c\right) \quad (138)$$

for the binary neuron cut $\in V_1$, which implies

$$y_{\mathrm{cut}}^{(2(n-1)+1)} = 1 \quad \text{iff} \quad \sum_{k=1}^{n-1}\alpha(z_{n-k})\beta^{-k} \geq c \quad (139)$$

due to (3), (138), and $\nu > 0$. Moreover, the output neuron out $\in V_2$ computes the logical negation of output from cut $\in V_1$ by (133) and (134), and hence,

$$y_{\mathrm{out}}^{(2n)} = 1 \quad \text{iff} \quad \sum_{k=1}^{n-1}\alpha(z_{n-k})\beta^{-k} < c \quad (140)$$

from (139). We can conclude that $z_1 \ldots z_n \in L_{<c}^R \cdot \Gamma$ iff $z_{n-1} \ldots z_1 \in L_{<c}$ iff $y_{\mathrm{out}}^{(2n)} = 1$ due to (140), which means $\mathcal{L}(\mathcal{N}) = L_{<c}^R \cdot \Gamma$. □

Theorem 7 *There is a language $L' = \mathcal{L}(\mathcal{N}')$ accepted by a 1ANN $\mathcal{N}'$ with rational weights, which is not context-free.*

Proof. Example 4 presents instances of numbers that are not quasi-periodic, e.g. $c = 1$ is not β-quasi-periodic within $A = \{0,1\}$ for $\beta = \frac{3}{2}$. Let $\Gamma = A$ and $\alpha : \Gamma \longrightarrow A$ be the identity. According to Theorem 2, we thus know that the corresponding cut language $L_{<c} \subseteq \Gamma^*$ over alphabet Γ is not context-free. It follows that the same holds for $L' = L_{<c}^R \cdot \Gamma$ since $L_{<c}^R = \{\mathbf{y} \in \Gamma^* \mid \mathbf{y}0 \in L'\}$ and the context-free languages are closed under GSM (generalized sequential machine) mapping and reversal. On the other hand, L' can be recognized by a 1ANN $\mathcal{N}'$ according to Lemma 1, because (126) follows from $\beta > 0$ and $\alpha(z) = z \geq 0$ for $z \in \Gamma = \{0,1\}$. Moreover, $\mathcal{N}'$ has rational weights (128)–(134) since $c = 1$, $\beta = \frac{3}{2}$ and $\nu = \sum_{k=1}^{\infty}\left(\frac{3}{2}\right)^{-k} = 2$ by (127). □

Example 1 (continuing from p. 35) In fact, Lemma 1 is a generalization of the running example of the 1ANN language acceptor $\mathcal{N}$ that is depicted in Figure 1 which is an instance of Figure 4 for $X = \{1,2\}$ with

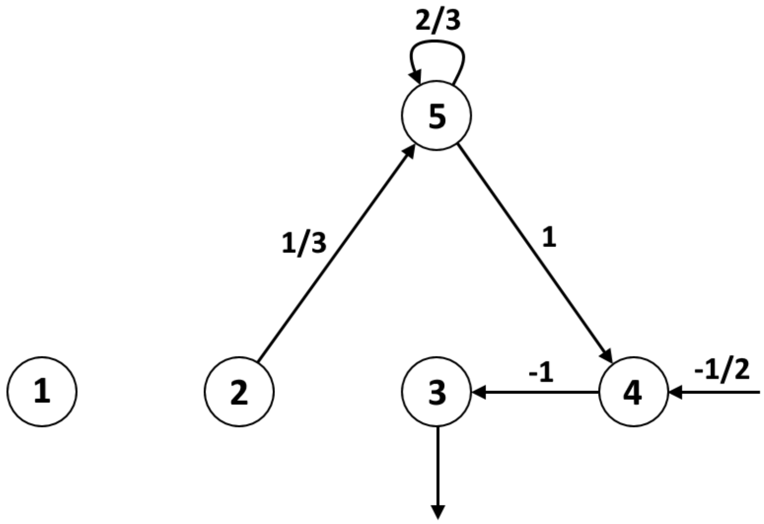

Figure 5: Example of a 1ANN $\mathcal{N}'$ recognizing a language that is not context-free.

$\alpha(\lambda_1) = \alpha(0) = 0$ and $\alpha(\lambda_2) = \alpha(1) = 1$, out $= 3$, cut $= 4$, $s = 5$, $c = \frac{1}{4}$, $\beta = 3$, and $\nu = \sum_{k=1}^{\infty} 3^{-k} = \frac{1}{2}$. It follows that

$$L = \mathcal{L}(\mathcal{N}) = L^R_{<\frac{1}{4}} \cdot \{0,1\} = \left\{ x_1 \ldots x_n \in \{0,1\}^* \;\middle|\; \sum_{k=1}^{n-1} x_{n-k}\, 3^{-k} < \frac{1}{4} \right\} \quad (141)$$

which coincides with the language (119) since in fact only the next-to-last symbol x_{n-1} in (141) decides whether $x_1 \ldots x_n \in L$. Namely, if $x_{n-1} = 1$, then $\sum_{k=1}^{n-1} x_{n-k}\, 3^{-k} \geq \frac{1}{3}$, while $\sum_{k=1}^{n-1} x_{n-k}\, 3^{-k} \leq \sum_{k=2}^{\infty} 3^{-k} = \frac{1}{6}$ for $x_{n-1} = 0$.

In contrast, the non-context-free language in Theorem 7 (see its proof),

$$L' = \mathcal{L}(\mathcal{N}') = L^R_{<1} \cdot \{0,1\} = \left\{ x_1 \ldots x_n \in \{0,1\}^* \;\middle|\; \sum_{k=1}^{n-1} x_{n-k} \left(\tfrac{3}{2}\right)^{-k} < 1 \right\} \quad (142)$$

is accepted by $\mathcal{N}'$ which is an instance of the 1ANN in Figure 4 (Lemma 1) for $c = 1$, $\beta = \frac{3}{2}$ and $\nu = 2$. In particular, $\mathcal{N}'$ differs from $\mathcal{N}$ in Figure 1 only in the weights $w_{52} = \frac{1}{3}$, and $w_{55} = \frac{2}{3}$, as depicted in Figure 5.

Theorem 7 provides a lower bound on the computational power of 1ANNs with rational weights. In the following theorem we prove a corresponding upper bound.

Theorem 8 *Any language $L = \mathcal{L}(\mathcal{N})$ accepted by a 1ANN $\mathcal{N}$ with rational weights and $0 < |w_{ss}| < 1$, is context-sensitive.*

Proof. According to Theorem 4, the language $L = \mathcal{L}(\mathcal{N})$ accepted by a 1ANN $\mathcal{N}$ can be written in the form (30) where $\beta \in \mathbb{Q}$, $A \subset \mathbb{Q}$, and $c_1, \ldots, c_{p+1} \in \mathbb{Q}$ are rationals by the assumption on the weights of $\mathcal{N}$. Since the context-sensitive languages are closed under complementation and intersection, Theorem 3 ensures that L_r used in (31) is context-sensitive for every $r = 1, \ldots, p$.

Furthermore, the context-sensitive languages are known to be closed under reversal, concatenation, union, Kleene star, and ε-free homomorphism. In addition, if S is context-sensitive, then its largest prefix-closed subset S^{Pref} is also context-sensitive as a nondeterministic linear bounded automaton (LBA) $\mathcal{M}^{Pref}$ that recognizes $S^{Pref} = L(\mathcal{M}^{Pref})$ runs successively LBA $\mathcal{M}$ for $S = \mathcal{L}(\mathcal{M})$ on every prefix of an input which can be stored within linear space, and S^{Pref} accepts if all these runs of $\mathcal{M}$ are accepting computations. Thus, it follows from (30) and (31) that the language L is context-sensitive. □

7. Conclusion

In this paper we have characterized the class of languages that are accepted online by binary-state neural networks with an extra analog unit, which is a subrecursive (sub-Turing) intermediate computational model between binary-state NNs, corresponding to finite automata, and analog-state NNs which are Turing universal. By using this characterization we have shown that the computational power of such networks with rational weights is between context-free and context-sensitive languages. In addition, we have formulated a sufficient condition when these networks accept only regular languages in terms of quasi-periodicity of their real weight parameters. The question of whether this condition is also necessary remains open.

Another challenge for further research is to generalize the representation Theorem 4 to offline 2ANNs employing two extra analog units and to find out whether 2ANNs are Turing universal (cf. Šíma, 2018). The ultimate goal is to prove a proper "natural" hierarchy of NNs between integer and rational weights similarly as it is known between rational and real weights (Balcázar et al., 1997) and possibly, map it to known hierarchies of regular/context-free languages. This problem is related to a more general issue of finding suitable complexity measures of subrecursive NNs establishing the complexity hierarchies, which could be employed in practical neurocomputing, e.g. the precision of weight parameters, energy complexity (Šíma, 2014a), temporal coding etc.

Acknowledgment

The presentation of this paper benefited significantly from valuable suggestions of an anonymous reviewer. The research was done with institutional support RVO: 67985807 and partially supported by the grant of the Czech Science Foundation No. 19-05704S.

References

Alon, N., Dewdney, A. K., Ott, T. J., 1991. Efficient simulation of finite automata by neural nets. Journal of the ACM 38 (2), 495–514.

Baker, S., 2014. Generalized golden ratios over integer alphabet. Integers 14, A15.

Baker, S., Masáková, Z., Pelantová, E., Vávra, T., 2017. On periodic representations in non-Pisot bases. Monatshefte für Mathematik 184 (1), 1–19.

Balcázar, J. L., Gavaldà, R., Siegelmann, H. T., 1997. Computational power of neural networks: A characterization in terms of Kolmogorov complexity. IEEE Transactions on Information Theory 43 (4), 1175–1183.

de Vries, M., Komornik, V., 2009. Unique expansions of real numbers. Advances in Mathematics 221 (2), 390–427.

Dombek, D., Masáková, Z., Pelantová, E., 2011. Number representation using generalized $(-\beta)$-transformation. Theoretical Computer Science 412 (48), 6653–6665.

Erdös, P., Joó, I., Komornik, V., 1990. Characterization of the unique expansions $1 = \sum_{i=1}^{\infty} q^{-n_i}$ and related problems. Bulletin de la Société Mathématique de France 118 (3), 377–390.

Frougny, C., Heller, P., Pelantová, E., Svobodová, M., 2014. k-Block parallel addition versus 1-block parallel addition in non-standard numeration systems. Theoretical Computer Science 543, 52–67.

Frougny, C., Lai, A. C., 2011. Negative bases and automata. Discrete Mathematics & Theoretical Computer Science 13 (1), 75–94.

Glendinning, P., Sidorov, N., 2001. Unique representations of real numbers in non-integer bases. Mathematical Research Letters 8 (4), 535–543.

Hare, K. G., 2007. Beta-expansions of Pisot and Salem numbers. In: Proceedings of the Waterloo Workshop in Computer Algebra 2006: Latest Advances in Symbolic Algorithms. World Scientific, pp. 67–84.

Horne, B. G., Hush, D. R., 1996. Bounds on the complexity of recurrent neural network implementations of finite state machines. Neural Networks 9 (2), 243–252.

Indyk, P., 1995. Optimal simulation of automata by neural nets. In: Proceedings of the STACS 1995 Twelfth Annual Symposium on Theoretical Aspects of Computer Science. Vol. 900 of LNCS. pp. 337–348.

Kilian, J., Siegelmann, H. T., 1996. The dynamic universality of sigmoidal neural networks. Information and Computation 128 (1), 48–56.

Koiran, P., 1996. A family of universal recurrent networks. Theoretical Computer Science 168 (2), 473–480.

Komornik, V., Lai, A. C., Pedicini, M., 2011. Generalized golden ratios of ternary alphabets. Journal of the European Mathematical Society 13 (4), 1113–1146.

Komornik, V., Pedicini, M., 2017. Critical bases for ternary alphabets. Acta Mathematica Hungarica 152 (1), 25–57.

Kong, D., Li, W., Dekking, F. M., 2010. Intersections of homogeneous cantor sets and beta-expansions. Nonlinearity 23 (11), 2815–2834.

Lupanov, O. B., 1973. On the synthesis of threshold circuits. Problemy Kibernetiki 26, 109–140.

Minsky, M., 1967. Computations: Finite and Infinite Machines. Prentice-Hall, Englewood Cliffs.

Orponen, P., 1997. Computing with truly asynchronous threshold logic networks. Theoretical Computer Science 174 (1-2), 123–136.

Parry, W., 1960. On the β-expansions of real numbers. Acta Mathematica Hungarica 11 (3), 401–416.

Rényi, A., 1957. Representations for real numbers and their ergodic properties. Acta Mathematica Academiae Scientiarum Hungaricae 8 (3-4), 477–493.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117.

Schmidt, K., 1980. On periodic expansions of Pisot numbers and Salem numbers. Bulletin of the London Mathematical Society 12 (4), 269–278.

Sidorov, N., 2003. Almost every number has a continuum of β-expansions. The American Mathematical Monthly 110 (9), 838–842.

Siegelmann, H. T., 1996. Recurrent neural networks and finite automata. Journal of Computational Intelligence 12 (4), 567–574.

Siegelmann, H. T., 1999. Neural Networks and Analog Computation: Beyond the Turing Limit. Birkhäuser, Boston.

Siegelmann, H. T., Sontag, E. D., 1994. Analog computation via neural networks. Theoretical Computer Science 131 (2), 331–360.

Siegelmann, H. T., Sontag, E. D., 1995. On the computational power of neural nets. Journal of Computer System Science 50 (1), 132–150.

Šíma, J., 1997. Analog stable simulation of discrete neural networks. Neural Network World 7 (6), 679–686.

Šíma, J., 2014a. Energy complexity of recurrent neural networks. Neural Computation 26 (5), 953–973.

Šíma, J., 2014b. The power of extra analog neuron. In: Proceedings of the TPNC 2014 Third International Conference on Theory and Practice of Natural Computing. Vol. 8890 of LNCS. pp. 243–254.

Šíma, J., 2017. Neural networks between integer and rational weights. In: Proceedings of the IJCNN 2017 Thirties International Joint Conference on Neural Networks. IEEE, pp. 154–161.

Šíma, J., 2018. Three analog units are Turing universal. In: Proceedings of the TPNC 2018 Seventh International Conference on Theory and Practice of Natural Computing. Vol. 11324 of LNCS. pp. 460–472.

Šíma, J., Orponen, P., 2003. General-purpose computation with neural networks: A survey of complexity theoretic results. Neural Computation 15 (12), 2727–2778.

Šíma, J., Savický, P., 2018. Quasi-periodic β-expansions and cut languages. Theoretical Computer Science 720, 1–23.

Šíma, J., Wiedermann, J., 1998. Theory of neuromata. Journal of the ACM 45 (1), 155–178.

Šorel, M., Šíma, J., 2004. Robust RBF finite automata. Neurocomputing 62, 93–110.