



Invariants for neural automata

Jone Uria-Albizuri¹ · Giovanni Sirio Carmantini² · Peter beim Graben³ · Serafim Rodrigues⁴

Received: 10 March 2023 / Revised: 15 April 2023 / Accepted: 1 May 2023 / Published online: 31 May 2023
© The Author(s) 2023

Abstract

Computational modeling of neurodynamical systems often deploys neural networks and symbolic dynamics. One particular way for combining these approaches within a framework called *vector symbolic architectures* leads to neural automata. Specifically, neural automata result from the assignment of symbols and symbol strings to numbers, known as Gödel encoding. Under this assignment, symbolic computation becomes represented by trajectories of state vectors in a real phase space, that allows for statistical correlation analyses with real-world measurements and experimental data. However, these assignments are usually completely arbitrary. Hence, it makes sense to address the problem which aspects of the dynamics observed under a Gödel representation is intrinsic to the dynamics and which are not. In this study, we develop a formally rigorous mathematical framework for the investigation of symmetries and invariants of neural automata under different encodings. As a central concept we define *patterns of equality* for such systems. We consider different macroscopic observables, such as the mean activation level of the neural network, and ask for their invariance properties. Our main result shows that only step functions that are defined over those patterns of equality are invariant under symbolic recodings, while the mean activation, e.g., is not. Our work could be of substantial importance for related regression studies of real-world measurements with neurosymbolic processors for avoiding confounding results that are dependant on a particular encoding and not intrinsic to the dynamics.

Keywords Computational cognitive neurodynamics · Symbolic dynamics · Neural automata · Observables · Invariants · Language processing

Introduction

Computational cognitive neurodynamics deals to a large extent with statistical modeling and regression analyses between behavioral and neurophysiological observables on the one hand and neurocomputational models of cognitive processes on the other hand (Gazzaniga et al. 2002; Rabinovich et al. 2012). Examples for experimentally measurable observables are response times (RT), eye-movements (EM), event-related brain potentials (ERP) in the domain of electroencephalography (EEG), event-related magnetic fields (ERF) in the domain of magnetoencephalography (MEG), or the blood-oxygen-level-dependent signal (BOLD) in functional magnetic resonance imaging (fMRI).

Computational models for cognitive processes often involve drift-diffusion approaches (Ratcliff 1978; Ratcliff and McKoon 2007), cognitive architectures such as ACT-R (Anderson et al. 2004), automata theory (Hopcroft and Ullman 1979), dynamical systems (van Gelder 1998; Kelso

✉ Jone Uria-Albizuri
jone.uria@ehu.eus

✉ Serafim Rodrigues
srodrigues@bcamath.org

Giovanni Sirio Carmantini
giovanni@carmantini.com

Peter beim Graben
Peter.beimGraben@b-tu.de

¹ Department of Mathematics, University of the Basque Country, Leioa, Spain

² foldAI, Munich, Germany

³ Bernstein Center for Computational Neuroscience, Berlin, Germany

⁴ Basque Center for Applied Mathematics, Bilbao, Spain

1995; Rabinovich and Varona 2018), and notably neural networks (e.g. Hertz et al. (1991); Arbib (1995)) that became increasingly popular after the induction of deep learning techniques in recent time (LeCun et al. 2015; Schmidhuber 2015).

For carrying out statistical correlation analyses between experimental data and computational models one has to devise *observation models*, relating the microscopic states within a computer simulation (e.g. the spiking of a simulated neuron) with the above-mentioned macroscopically observable measurements. In decision making, e.g., a suitable observation model is first passage time in a drift-diffusion model (Ratcliff 1978; Ratcliff and McKoon 2007). In the domain of neuroelectrophysiology, local field potentials (LFP) and EEG can be described through macroscopic mean-fields, based either on neural compartment models (Mazzoni et al. 2008; beim Graben and Rodrigues 2013; Martínez-Cañada et al. 2021), or neural field theory (Jirsa et al. 2002; beim Graben and Rodrigues 2014). For MRI and BOLD signals, particular hemodynamic observation models have been proposed (Friston et al. 2000; Stephan et al. 2004).

In the fields of computational psycholinguistics and computational neurolinguistics (Arbib and Caplan 1979; Crocker 1996; beim Graben and Drenhaus 2012; Lewis 2003) a number of studies employed statistical regression analysis between measured and simulated data. To name only a few of them, Davidson and Martin (2013) modeled speed-accuracy data from a translation-recall experiment among Spanish and Basque subjects through a drift-diffusion approach (Ratcliff 1978; Ratcliff and McKoon 2007). Lewis and Vasishth (2006) correlated self-paced reading times for English sentences of different linguistic complexity with the predictions of an ACT-R model (Anderson et al. 2004). Huyck (2009) devised a Hebbian cell assembly network of spiking point neurons for a related task. Using an automaton model for formal language (Hopcroft and Ullman 1979), Stabler (2011) argued how reading times could be related to the automaton's working memory load. Similarly, Boston et al. (2008) compared eye-movement data with the predictions of an automaton model for probabilistic dependency grammars (Nivre 2008).

Correlating human language processing with event-related brain dynamics became an important subject of computational neurolinguistics in recent years. Beginning with the seminal studies of beim Graben et al. (2000, 2004), similar work has been conducted by numerous research groups (for an overview cf. Hale et al. (2022)). Also to name only a few of them, Hale et al. (2015) correlated different formal language models with the BOLD response of participants listening to speech. Similarly, Frank et al. (2015) used different ERP components in the EEG, such as the N400 (a deflection of

negative polarity appearing about 400 ms after stimulus onset as a marker of lexical-semantic access) for such statistical modeling. beim Graben and Drenhaus (2012) correlated the temporally integrated ERP during the understanding of negative polarity items (Krifka 1995) with the harmony observable of a recurrent neural network (Smolensky 2006), thereby implementing a formal language processor as a *vector symbolic architecture* (Gayler 2006; Schlegel et al. 2021). Another neural network model of the N400 ERP-component is due to Rabovsky and McRae (2014), and to Rabovsky et al. (2018) who related this marker with neural prediction error and semantic updating as observation models. Similar ideas have been suggested by Brouwer et al. (2017); Brouwer and Crocker (2017), and Brouwer et al. (2021) who considered a deep neural network of layered simple recurrent networks (Cleeremans et al. 1989; Elman 1990), where the basal layer implements lexical retrieval, thus accounting for the N400 ERP-component, while the upper layer serves for contextual integration. Processing failures at this level are indicated by another ERP-component, the P600 (a positively charged deflection occurring around 600 ms after stimulus onset). Their neurocomputational model thereby implemented a previously suggested retrieval-integration account (Brouwer et al. 2012; Brouwer and Hoeks 2013).

In the studies of beim Graben et al. (2000, 2004, 2008), a dynamical systems approach was deployed — later dubbed *cognitive dynamical modeling* by beim Graben and Potthast (2009). This denotes a three-tier approach starting firstly with symbolic data structures and algorithms as models for cognitive representations and processes. These symbolic descriptions are secondly mapped onto a vectorial representation within the framework of vector symbolic architectures (Gayler 2006; Schlegel et al. 2021) through filler-role bindings and subsequent tensor product representations (Smolensky 1990, 2006; Mizraji 1989, 2020). In a third step, these linear structures are used as training data for neural network learning. More specifically, symbol strings and formal language processors can be mapped through Gödel encodings to *dynamical automata* (beim Graben et al. 2000, 2004, 2008; beim Graben and Potthast 2014).

In the seminal work by Siegelmann and Sontag (1995), recurrent neural networks are shown to support universal computation. Specifically, the authors construct a neural network architecture able to simulate a Universal Turing Machine in real-time. Their approach is based on the fractal encoding of the machine tape and state (beim Graben and Potthast 2009), and the application of affine linear transformations that appropriately map the encoded tape and state to a new tape and state, as dictated by the machine table, at each computation step. Recently, Car-mantini et al. (2017) have shown that recurrent neural

networks can also simulate dynamical automata in real-time, within an architecture named by the authors as *neural automata* (NA). Similarly to the approach of Siegelmann and Sontag (1995), NA make use of linear units in the network to apply affine linear transformations onto the fractal encoding of symbol strings. However, by basing their construction on dynamical automata, Carmantini et al. (2017) were able to define simpler, more parsimonious networks, with a direct correspondence between network architecture and the structure of the dynamical automata, they simulate.¹

Carmantini et al. (2017) also showed how neural automata can be used for neurolinguistic correlation studies. They implemented a diagnosis-repair parser (Lewis 1998; Lewis and Vasishth 2006) for the processing of initially ambiguous subject relative and object relative sentences (Frisch et al. 2004; Lewis and Vasishth 2006) through an interactive automata network. As an appropriate observation model they exploited the mean activation of the resulting neural network (Amari 1974) as *synthetic ERP* (beim Graben et al. 2008; Barrès et al. 2013) and obtained a model for the P600 component in their attempt.

For all these neurocomputational models symbolic content must be encoded as neural activation patterns. In vector symbolic architectures, this procedure involves a mapping of symbols onto filler vectors and of their possible binding sites in a data structure onto role vectors (beim Graben and Potthast 2009). Obviously, such an encoding is completely arbitrary and could be replaced at least by any possible permutation of a chosen code. Therefore, the question arises to what extent neural observation models remain *invariant* under permutations of an arbitrarily chosen code. Even more crucially, one has to face the problem whether a statistical correlation analysis depends on only one particularly chosen encoding, or not. Only if statistical models are also invariant under recoding, they could be regarded as reliable methods of scientific investigation.

It is the aim of the present study to provide a rigorous mathematical treatment of invariant observation models for the particular case of dynamical and neural automata and their underlying shift spaces. The article is structured as follows. In “[Invariants in dynamical systems](#)” section we introduce the general concepts and basic definitions about invariants in dynamical systems, focusing later in “[Neurodynamics](#)” section on the special case of neurodynamical ones. In “[Symbolic dynamics](#)” section we focus our attention on symbolic dynamics. After introducing the

basic notation we discuss the tools and facts that are needed in “[Rooted trees](#)” section about rooted trees and about Gödel encodings in “[Gödel encodings](#)” section. In “[Cylinder sets](#)” section we relate these concepts to cylinder sets in order to finally describe the invariant partitions for different Gödelizations of strings in “[Invariants](#)” section. Then, in “[Neural automata](#)” section we describe the architecture for neural automata and how to pass from single strings to dotted sequences. Finally, in “[Invariant observables](#)” section we describe a symmetry group defined by Gödel recoding of alphabets for neural automata, and we define a macroscopic observable that is invariant under this symmetry, based on the invariants described in “[Invariants](#)” section before. In the end, in “[Neurolinguistic application](#)” section, we apply our results to a concrete example with a neural automaton constructed to emulate parser for a context-free grammar. We demonstrate that the given macroscopic observable is invariant under Gödel recodings, whereas Amari’s mean network activity is not. “[Discussion](#)” section provides a concluding discussion. All the mathematical proofs about the facts claimed throughout the paper are collected in an “[appendix](#)”.

Invariants in dynamical systems

We consider a classical time-discrete and deterministic dynamical system in its most generic form as an ordered pair $\Sigma = (X, \Phi)$ where $X \subset \mathbb{R}^n$ is a compact Hausdorff space as its phase space of dimension $n \in \mathbb{N}$ and $\Phi : X \rightarrow X$ is an invertible (generally nonlinear) map (Atmanspacher and beim Graben 2007). The flow of the system is generated by the time iterates Φ^t , $t \in \mathbb{Z}$, i.e., $t \mapsto \Phi^t$ is a one-parameter group for the dynamics with time $t \in \mathbb{Z}$, obeying $\Phi^t \circ \Phi^s = \Phi^{t+s}$ for $t, s \in \mathbb{Z}$. The elements of the phase space $\mathbf{x} \in X$ refer to the microscopic description of the system Σ and are therefore called *microstates*. After preparation of an *initial condition* $\mathbf{x}_0 \in X$ the system evolves deterministically along a *trajectory* $T = \{\mathbf{x}(t) = \Phi^t(\mathbf{x}_0) \mid t \in \mathbb{Z}\}$.

A bounded function $f : X \rightarrow \mathbb{R}$ is called an *observable* with $f(\mathbf{x}) \in \mathbb{R}$ as measurement result in microstate \mathbf{x} . The function space $B(X) = \{f : X \rightarrow \mathbb{R} \mid \|f\| < \infty\}$, endowed with point wise function addition $(f + g)(x) = f(x) + g(x)$, function multiplication $(fg)(x) = f(x)g(x)$, and scalar multiplication $(\lambda f)(x) = \lambda f(x)$ (for all $f, g \in B(X)$, $\lambda \in \mathbb{R}$) is called the observable algebra of the system Σ with norm $\|\cdot\| : B(X) \rightarrow \mathbb{R}_0^+$. Restricting the function space $B(X)$ to the bounded continuous functions $C_0(X)$, yields the algebra of *microscopic observables* which describe ideal

¹ Note that neural automata are parsimonious implementations of universal computers, especially of Turing machines. These are not to be confused with neural Turing machines appearing in the framework of deep learning approaches (Graves et al. 2014).

measurements for uniquely distinguishing among different microstates within certain regions of phase space.

By contrast, complex real-world dynamical systems only allow the measurement of macroscopic properties. The corresponding *macroscopic observables* belong to the larger algebra of bounded functions² $B(X)$ and are usually defined as large-scale limits of so-called mean-fields (Hepp 1972; Sewell 2002). Examples for macroscopic mean-field observables in computational neuroscience are discussed below.

The algebra of macroscopic observables $B(X)$ contains step functions and particularly the indicator functions χ_A for proper subsets $A \subset X$ which are not continuous over whole X . Because $\chi_A(\mathbf{x}) = \chi_A(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in A$, the microstates \mathbf{x} and \mathbf{y} are not distinguishable by means of the macroscopic measurement of χ_A . Thus, Jauch (1964) and Emch (1964) called them *macroscopically equivalent*.³ The class of macroscopically equivalent microstates forms a *macrostate* in the given mathematical framework (Jauch 1964; Emch 1964; Sewell 2002). Hence, a macroscopic observable induces a partition of the phase space of a dynamical system Σ into macrostates.

The algebras of microscopic observables, $C_0(X)$, and of macroscopic observables, $B(X)$, respectively, are linear spaces with their additional algebraic products. As vector spaces, they allow the construction of linear homomorphisms $\varphi : B(X) \rightarrow B(X)$ which are vector spaces as well. An important subspace of the space of linear homomorphisms is provided by the space of linear automorphisms, $\text{Aut}(B(X))$, which contains the invertible linear homomorphisms. The space $\text{Aut}(B(X))$ is additionally a group with respect to function composition, $(\varphi \circ \eta)(f)$, called the *automorphism group* of the algebra $B(X)$.

Next, let G be a group possessing a faithful representation α in the automorphism group $\text{Aut}(B(X))$ of the dynamical system Σ ; that is, $\alpha : G \rightarrow \text{Aut}(B(X))$ is an injective group homomorphism. Then, for $a \in G$, $\alpha_a \in \text{Aut}(B(X))$ maps an observable $f \in B(X)$ onto its transformed $\alpha_a(f) \in B(X)$, such that for two $a, b \in G$ it holds $\alpha_{a*b}(f) = (\alpha_a \circ \alpha_b)(f)$ where ‘*’ denotes the group product in G . The group G is called a *symmetry* of the dynamical system Σ (Sewell 2002). Moreover, if the representation of G commutes with the dynamics of Σ ,

$$(\alpha_a(f \circ \Phi^t))(\mathbf{x}) = f(\Phi^t(\alpha_a^*(\mathbf{x}))) \quad (1)$$

for all $\mathbf{x} \in X$, the group G is called *dynamical symmetry*

(Sewell 2002). In Eq. (1), the map $\alpha_a^* : X \rightarrow X$ is the lifting result from the observables to phase space through

$$f \circ \alpha_a^* = \alpha_a(f). \quad (2)$$

As an example consider the macroscopic observable χ_A , i.e. the indicator function for a proper subset $A \subset X$ again. Choosing α_a^* in such a way that $\alpha_a^*(\mathbf{x}) \in A$ for all $\mathbf{x} \in A$, leaves χ_A invariant: $\chi_A(\alpha_a^*(\mathbf{x})) = \chi_A(\mathbf{x})$.

More generally, we say that an observable $f \in B(X)$ is *invariant* under the symmetry G if

$$f(\alpha_a^*(\mathbf{x})) = f(\mathbf{x}) \quad (3)$$

for all $a \in G$. It is the aim of the present study to investigate such invariants for particular neurodynamical systems, namely dynamical and neural automata (beim Graben et al. 2000, 2004, 2008; Carmantini et al. 2017).

Neurodynamics

Neurodynamical systems are essentially recurrent neural networks consisting of a large number, $n \in \mathbb{N}$, of model neurons (or units) that are connected in a complex graph (Hertz et al. 1991; Arbib 1995; LeCun et al. 2015; Schmidhuber 2015). Under a suitable normalization, the activity of a unit, e.g. its spike rate can be represented by a real number in the unit interval $[0, 1] \subset \mathbb{R}$. Then, the microstate of the entire network becomes a vector in the n -dimensional hypercube, $\mathbf{x} \in X = [0, 1]^n \subset \mathbb{R}^n$. The microscopic observables are projectors on the individual coordinate axes,

$$f_i(\mathbf{x}) = x_i$$

for $1 \leq i \leq n$. For discrete time, the network dynamics is generally given as a nonlinear difference equation

$$\mathbf{x}(t+1) = \Phi_{\mathbf{W}}(\mathbf{x}(t)). \quad (4)$$

Here $\mathbf{x}(t) \in X$ is the activation vector (the microstate) of the network at time t and $\Phi_{\mathbf{W}}$ is a nonlinear map, parameterized by the synaptic weight matrix $\mathbf{W} \in \mathbb{R}^{n^2}$. Often, the map $\Phi_{\mathbf{W}}$ is assumed to be of the form

$$\Phi_{\mathbf{W}}(\mathbf{x}) = \mathbf{F}(\mathbf{W} \cdot \mathbf{x}), \quad (5)$$

with a nonlinear squashing function $\mathbf{F} = (F_i)_{1 \leq i \leq n} : X \rightarrow X$ as the *activation function* of the network. For $F_i = \Theta$ (where Θ denotes the Heaviside jump function), equations (4, 5) describe a network of McCulloch-Pitts neurons (McCulloch and Pitts 1943). Another popular choice for the activation function is the logistic function

² In fact one needs the algebra of essentially bounded functions with respect to a given probability measure here. For a proper treatment of these concepts, *algebraic quantum theory* is required (Sewell 2002).

³ Cf. the related concept of epistemic equivalence used by beim Graben and Atmanspacher (2006, 2009).

$$F_i(x) = \frac{1}{1 + e^{-x_i}},$$

describing firing rate models (cf., e.g., beim Graben (2008)). Replacing Eq. (5) by the map

$$\Phi_{\mathbf{W}}(\mathbf{x}) = (1 - \Delta t)\mathbf{x} + \Delta t \mathbf{F}(\mathbf{W} \cdot \mathbf{x}) \tag{6}$$

yields a time-discrete leaky integrator network (Wilson and Cowan 1972; beim Graben et al. 2009; beim Graben and Rodrigues 2013). For numerical simulations using the Euler method, $\Delta t < 1$ is chosen for the time step.

For correlation analyses of neural network simulations with experimental data from neurophysiological experiments one needs a mapping from the high-dimensional neural activation space $X \subset \mathbb{R}^n$ into a much lower-dimensional *observation space* that is spanned by $p \in \mathbb{N}$ macroscopic observables $f_k : X \rightarrow \mathbb{R}$ ($1 \leq k \leq p$). A standard method for such a projection is principal component analysis (PCA) (Elman 1991). If PCA is restricted to the first principal axis, the resulting scalar variable could be conceived as a measure of the overall activity in the neural network. In the realm of computational neurolinguistics PCA projections were exploited by beim Graben et al. (2008).

Another important scalar observable, e.g. used by beim Graben and Drenhaus (2012) as a neuronal observation model, is Smolensky’s harmony (Smolensky 1986)

$$H(t) = \mathbf{x}(t)^+ \cdot \mathbf{W} \cdot \mathbf{x}(t) \tag{7}$$

with \mathbf{x}^+ as transposed activation state vector, and the synaptic weight matrix \mathbf{W} , above.

Brouwer et al. (2017) suggested the “dissimilarity” between the actual microstate and its dynamical precursor, i.e.

$$D(t) = 1 - \frac{\mathbf{x}(t) \cdot \mathbf{x}(t-1)}{\|\mathbf{x}(t)\| \|\mathbf{x}(t-1)\|} \tag{8}$$

as a suitable neuronal observation model.

In this study, however, we use Amari’s mean network activity (Amari 1974)

$$A(t) = \frac{1}{n} \sum_i x_i(t) \tag{9}$$

as time-dependent “synthetic ERP” (Barrès et al. 2013; Carmantini et al. 2017) of a neural network.

Symbolic dynamics

A symbolic dynamics arises from a time-discrete but space continuous dynamical system Σ through a partition of its phase space X into a finite family of m disjoint subsets totally covering the space X (Lind and Marcus 1995). Hence

$$\mathcal{P} = \{A_k \subset X | A_k \cap A_j = \emptyset \text{ for } k \neq j, \bigcup_{k=1}^m A_k = X\}.$$

Such a partition could be induced by a macroscopic observable with finite range. By assigning the index k of a partition set A_k as a distinguished *symbol* s_t to a state $\mathbf{x}(t)$ when $\mathbf{x}(t) \in A_k$, a trajectory of the system is mapped onto a two-sided infinite symbolic sequence. Correspondingly, the flow map of the dynamics Φ becomes represented by the left shift σ through $\sigma(s_t) = s_{t+1}$.

Following beim Graben et al. (2004, 2008), and Carmantini et al. (2017), a symbol is meant to be a distinguished element from a finite set \mathbf{A} , which we call an *alphabet*. A sequence of symbols $w \in \mathbf{A}^l$ is called a word of length l , denoted $l = |w|$. The set of words of all possible lengths w of finite length $|w| \geq 0$, also called the *vocabulary* over \mathbf{A} , is denoted \mathbf{A}^* (for $|w| = 0$, $w = \epsilon$ denotes the “empty word”).

Rooted trees

One can visualize the set of all words over the alphabet \mathbf{A} as a regular rooted tree, T , where each vertex is labeled by and corresponds to each word formed by using this alphabet. Let us assume that \mathbf{A} has m letters for some $m \in \mathbb{N}$. That is $\mathbf{A} = \{a_1, \dots, a_m\}$. Then, the tree T is inductively constructed as follows:

- (i) The root of the tree is a vertex labeled by the empty word ϵ .
- (ii) Assume we have constructed the vertices of step n , then we construct the vertices of step $n + 1$ as follows. Suppose that we have k vertices at step n that are labeled by the words w_1, \dots, w_k . Then
 - For each $i = 1, \dots, k$ and each $a_j \in \mathbf{A}$ we add a new vertex decorated by $w_i a_j$.
 - For each $i = 1, \dots, k$ and $j = 1, \dots, m$ we add an edge from w_i to $w_i a_j$.

This construction generates a regular rooted tree. Following the aforementioned construction, typically in the first step the root is placed at the top vertex. Subsequently the root is joined by edges, where each edge is associated to every word of length 1, that is, to every symbol of \mathbf{A} . Then iteratively, each of these edges labeled by a letter of \mathbf{A} is joined to any word of length two starting by that letter, and so on. Assuming that $\mathbf{A} = \{a_1, \dots, a_m\}$, this construction yields an infinite tree as in Fig. 1.

Each vertex of the tree corresponds to a word over the alphabet \mathbf{A} . That is, the set of vertices of the tree is \mathbf{A}^* . On the other hand, each infinite ray starting from the root, corresponds to an infinite sequence of symbols over \mathbf{A} , and

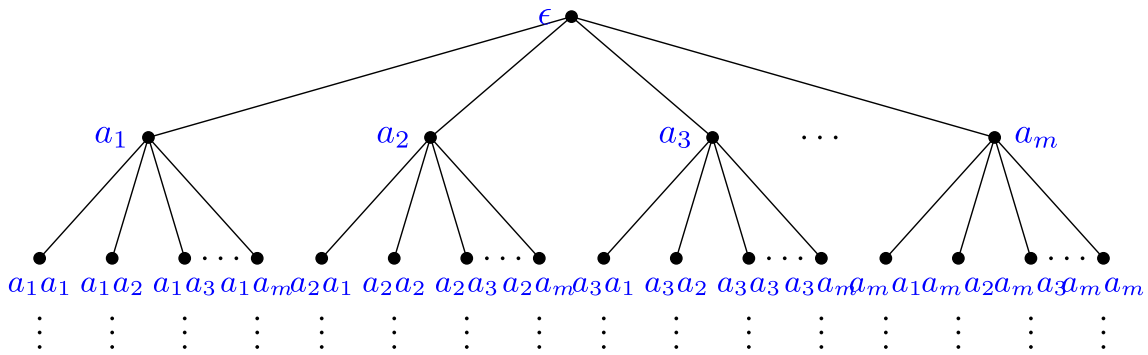


Fig. 1 The vocabulary \mathbf{A}^* as a rooted tree

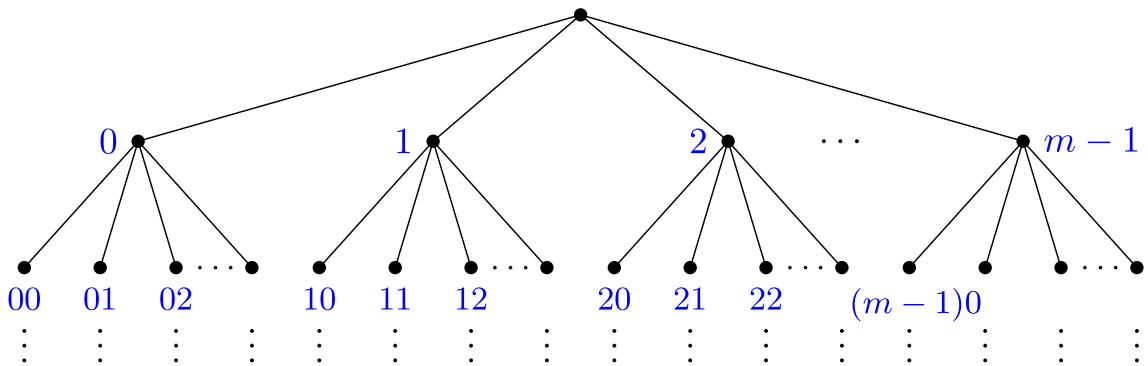


Fig. 2 The regular rooted tree T over the alphabet $\{0, 1, \dots, m - 1\}$

it belongs to the boundary of the tree. We denote this boundary by ∂T and as mentioned, viewed as a set is equal to $\mathbf{A}^{\mathbb{N}}$.

The construction of the tree is unique up to the particular ordering of the symbols in \mathbf{A} we chose. Thus, in principle, if $\gamma : \mathbf{A} \rightarrow \{0, \dots, m - 1\}$ is a particular ordering (i.e. a bijection) of the alphabet where an element a is denoted as a_i if $\gamma(a) = i - 1$, then the tree should be denoted by T_γ as it depends on that particular ordering of the alphabet.

Let us denote by T the regular rooted tree over the alphabet $\{0, 1, \dots, m - 1\}$ with the natural order induced by \mathbb{N} (see Fig. 2).

Henceforth will denote by \mathbf{M} the alphabet $\{0, \dots, m - 1\}$ and as before, by T the tree corresponding to the alphabet \mathbf{M} under the usual ordering on \mathbb{N} .

When we say that the construction is unique up to reordering of symbols, we mean that both trees are isomorphic as graphs, where an isomorphism of graphs is a bijection between vertices preserving incidence. Indeed, for any bijection $\gamma : \mathbf{A} \rightarrow \mathbf{M}$, the tree T_γ is isomorphic to T as a graph.

Lemma 1 *Let $\gamma : \mathbf{A} \rightarrow \mathbf{M}$ be an ordering of the alphabet \mathbf{A} . Then T_γ and T are isomorphic.*

Since being isomorphic is transitive, this lemma shows that for any two alphabets \mathbf{A}_1 and \mathbf{A}_2 of the same

cardinality and any two orderings of those alphabets γ_1 and γ_2 , their corresponding trees T_{γ_1} and T_{γ_2} will be isomorphic as graphs.

Gödel encodings

Having $\mathbf{A}^{\mathbb{N}}$, the space of one-sided infinite sequences over an alphabet \mathbf{A} containing $|\mathbf{A}| = m$ symbols and $s = a_1 a_2 \dots$ a sequence in this space, with a_k being the k -th symbol in s and an ordering $\gamma : \mathbf{A} \rightarrow \{0, \dots, m - 1\}$, then a Gödelization is a mapping from $\mathbf{A}^{\mathbb{N}}$ to $[0, 1] \subset \mathbb{R}$ defined as follows:

$$\psi_\gamma(s) := \sum_{k=1}^{\infty} \gamma(a_k) m^{-k}. \tag{10}$$

By the Lemma 1 we know that for each Gödelization of \mathbf{A} induced by γ , there is an isomorphism of graphs between T_γ and T . Since the choice for the ordering of the alphabet (in other words, the choice of γ) is arbitrary and leads to different Gödel encodings, we are interested in finding invariants for different such encodings.

One can define a metric on the boundary of the tree in the following way: given any two infinite rays of the tree $p = a_1 a_2 a_3 \dots$ and $q = b_1 b_2 b_3 \dots$ we define

$$d(p, q) = \begin{cases} 0 & \text{if } a_1 \neq b_1 \\ m^{-n} & \text{if } a_i = b_i \text{ for } i = 1, \dots, n \text{ and } a_{n+1} \neq b_{n+1} \\ 1 & \text{if } p = q \end{cases}$$

This defines an ultrametric on the boundary, that is, a metric that satisfies a stronger version of the triangular inequality, namely:

$$d(p, q) \leq \max\{d(p, r), d(r, q)\}.$$

When we encode the infinite strings under the Gödel encoding, we are sending rays that are close to each other under this ultrametric to points that are close in the $[0, 1]$ interval under the usual metric.

Lemma 2 *Let $p = a_1a_2a_3\dots$ and $q = b_1b_2b_3\dots$ be two infinite strings over \mathbf{A} . Then for any Gödel encoding ψ we have that*

$$d(p, q) \leq \frac{1}{m^n} \iff \exists k \in \{0, \dots, m^n - 1\}, \\ \psi(p), \psi(q) \in \left[\frac{k}{m^n}, \frac{k+1}{m^n} \right).$$

Recall that the lemma does not mean that points that are close (with respect to the usual metric) on the $[0, 1]$ interval come from rays that were close on the tree. For example, if the alphabet has 3 letters, the points $1/3 - \epsilon$ and $1/3$ are as close as we want for any $\epsilon > 0$ but are at distance 0 from each other on the tree. In fact, it gives a partition of the interval for each $n \in \mathbb{N}$ in a way that, if two points representing an infinite string are in the same interval according to the partition of the corresponding n , then they come from two rays that share at least a common prefix of length n .

Cylinder sets

In symbolic dynamics, a cylinder set (McMillan 1953) is a subset of the space $\mathbf{A}^{\mathbb{N}}$ of infinite sequences from an alphabet \mathbf{A} that agree in a particular building block of length $l \in \mathbb{N}$. Thus, let $w = \mathbf{A}^*$ be a finite word $a_1a_2\dots a_l$ of length l , we define the cylinder set

$$[w] = [a_1a_2\dots a_l] = \{s \in \mathbf{A}^{\mathbb{N}} \mid s_k = a_k, \quad k = 1, \dots, l\}. \tag{11}$$

We can also see the cylinder sets on the tree depicted in Fig. 3. In fact, for each level on the tree (where level refers to vertices corresponding to words of certain fixed length) we get a partition of the interval $[0, 1]$. The vertices hanging from each vertex on that level land on their corresponding interval of the partition. Thus, from a rooted tree view point, a cylinder set corresponds to a whole tree

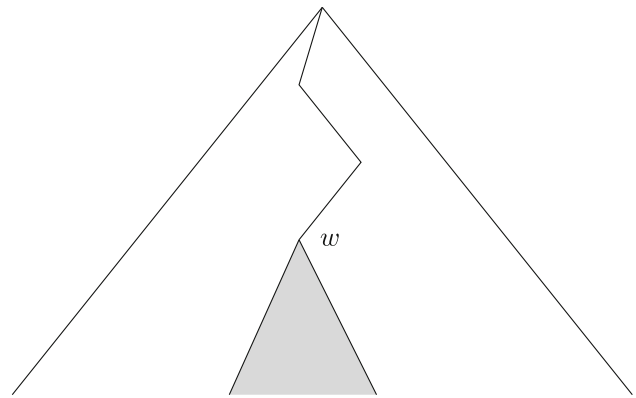


Fig. 3 Cylinder set corresponding to w seen on the tree

hanging from that vertex. Concretely, the cylinder set $[w]$ for the word $w \in \mathbf{A}^*$ is the subtree hanging from the vertex decorated by w .

Two different Gödel codes ψ, φ can only differ with respect to their assignments $\gamma_1, \gamma_2 : \mathbf{A} \rightarrow \{0, \dots, m - 1\}$. Thus, we call a permutation $\pi \in S_m$ (with S_m as the symmetric group) a *Gödel recoding*, if there exist two assignments γ_1 and γ_2 such that

$$\pi \circ \gamma_1 = \gamma_2.$$

Invariants

The ultimate goal of our study is to find invariants under Gödel recodings. Observe that under the notation of Lemma 1, $g_{\gamma_1} : T_{\gamma_1} \rightarrow T$ and $g_{\gamma_2} : T_{\gamma_2} \rightarrow T$ are two graph isomorphisms. In fact, they induce a graph automorphism of T , $g_{\pi} = g_{\gamma_2} \circ g_{\gamma_1}^{-1} : T \rightarrow T$. And this automorphism sends the vertices encoded by γ_1 to the ones encoded by γ_2 .

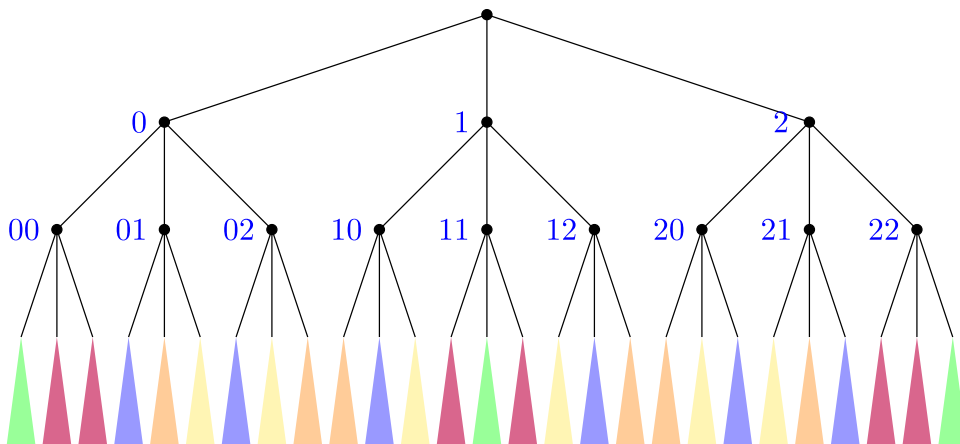
As Lemma 2 shows, a Gödel recoding preserves the size of cylinder sets after permuting vertices. However, the way of ordering the alphabet and how this permutes the rays of the tree is even more restrictive than just preserving the size of the cylinder sets. In fact, under the action of a reordering each vertex can only be mapped to certain vertices and it is forbidden to be sent to others. This is captured by the following most central definition.

Definition 1 Let $w = a_{i_1}a_{i_2}\dots a_{i_l} \in \mathbf{M}^l$ be a string of length l after an ordering γ . We define a partition of the set of integers $\{1, 2, \dots, l\}$,

$$\mathcal{P}_w = \{\{j_1, \dots, j_k\} \subset \mathbb{N} \mid a_{i_{j_1}} = \dots = a_{i_{j_k}}\}. \tag{12}$$

For any word $w \in \mathbf{M}^*$ we call \mathcal{P}_w the *pattern of equality* of w .

Fig. 4 Invariant partition of the cylinder sets according to their patterns of equality



Equipped with aforementioned formalisms we are now in a position to formulate the first main finding of our study as follows.

Theorem 1 For any other vertex $u \in T$ there exists a Gödel recoding π such that $g_\pi(w) = u$ if and only if

$$u \in \mathbf{M}^l \tag{13}$$

$$P_w = P_u. \tag{14}$$

Theorem 1 states that each vertex can be mapped to any vertex having the same pattern of equality and nowhere else.

Example 1 If $\mathbf{A} = \{a, b, c\}$ and we consider $w = aabcbabc \in \mathbf{A}^8$. Then we have $P_w = \{\{1, 2, 3, 6\}, \{4, 7\}, \{5, 8\}\}$, which gives us all the possible words where w can be mapped to. That would be the list of all the possibilities:

*bbbacbac cccbacba aaacbabc
bbbcabca cccabcab aaabcabc*

So we have only 6 possible vertices out of $3^8 = 6561$. And of course, this proportion reduces as we go deeper on the tree.

In terms of Gödelization into the $[0, 1]$ interval, we illustrate the implications by an example. Let us assume that $m = 3$ and $l = 3$, for example. Then, in Fig. 4 the cylinder sets of certain color can only be mapped through a recoding to a cylinder set of the same color and nowhere else.

Figure 5 shows the corresponding partition of the interval $[0, 1]$ where the intervals in each color may be mapped to another of the same color by a different assignment map and nowhere else.



Fig. 5 Invariant partition of the interval $[0, 1]$ after Gödelization

Neural automata

Following beim Graben et al. (2004, 2008), and Carmantini et al. (2017), a *dotted sequence* $s \in \mathbf{A}^{\mathbb{Z}}$ on an alphabet \mathbf{A} is a two-sided infinite sequence of symbols “ $s = \dots a_{-2} a_{-1} \cdot a_0 a_1 a_2 \dots$ ” where $a_i \in \mathbf{A}$, for all indices $i \in \mathbb{Z}$. Here, the dot “ \cdot ” is simply used as a mnemonic sign, indicating that the index 0 is to its right.

A shift space $M_S = (\mathbf{A}^{\mathbb{Z}}, \sigma)$ consists of the set $\mathbf{A}^{\mathbb{Z}}$ together with a map⁴ $\sigma : \mathbf{A}^{\mathbb{Z}} \rightarrow \mathbf{A}^{\mathbb{Z}}$ such that $\sigma(a_i) = a_{i+1}$ for $i \in \mathbb{Z}$ (Lind and Marcus 1995). Additionally, Moore (1990, 1991) have shown that the shift space $\mathbf{A}^{\mathbb{Z}}$ endowed with

$$F : \mathbf{A}^{\mathbb{Z}} \rightarrow \mathbb{Z}$$

$$\oplus : \mathbf{A}^{\mathbb{Z}} \times (\mathbf{A} \cup \{\Phi\})^{\mathbb{Z}} \rightarrow \mathbf{A}^{\mathbb{Z}} \tag{15}$$

$$G : \mathbf{A}^{\mathbb{Z}} \rightarrow (\mathbf{A} \cup \{\Phi\})^{\mathbb{Z}},$$

and their composition $\Omega(s) = \sigma^{F(s)}(s \oplus G(s))$, can simulate any Turing machine. The space $M_{GS} = (\mathbf{A}^{\mathbb{Z}}, \Omega)$ is called a *Generalized Shift (GS)* if there exists a *domain of dependance (DoD)*, i.e. an interval (k_l, k_r) , with $k_l \leq 0 \leq k_r$, such that the definition of the maps F and G only depend on the content of the string $s \in \mathbf{A}^{\mathbb{Z}}$ on that interval. The function G maps each symbol in the DoD of s to a symbol in \mathbf{A} , whereas all symbols outside of the DoD are mapped to an auxiliary symbol Φ . The \oplus operation then carries out a substitution operation where all symbols mapped to Φ by $G(s)$ are left untouched, whereas symbols in the DoD of

⁴ In the same way, the shift map’s inverse, σ^{-1} , can be defined as the operation that shifts every character in the string s one position to the right (or, equivalently, moves the dot one position to the left).

s are overwritten by their image under $G(s)$. Finally, the map F determines the number of shifts to be applied to the string resulting by the substitution operation.

Carmantini et al. (2017) introduced a more general shift space, called *versatile shift* (VS). The VS is equipped with a more versatile rewriting operation, where dotted words in the DoD are replaced by dotted words of arbitrary length, as opposed to replacing each symbol in the DoD with exactly one symbol, as in a GS. For that purpose the dot is interpreted as a meta-symbol which can be concatenated with two words $v_1, v_2 \in \mathbf{A}^*$ through $v = v_1.v_2$. Let $\hat{\mathbf{A}}^*$ denote the set of these dotted words. Moreover, let $\mathbb{Z}^- = \{i \mid i < 0, i \in \mathbb{Z}\}$ and $\mathbb{Z}^+ = \{i \mid i \geq 0, i \in \mathbb{Z}\}$ the sets of negative and non-negative indices. We can then reintroduce the notion of a dotted sequence as follows: a *dotted sequence* $s \in \mathbf{A}^{\mathbb{Z}}$ is a bi-infinite sequence of symbols such that $s = w_\alpha v w_\beta$ with $v \in \hat{\mathbf{A}}^*$ as a dotted word $v = v_1.v_2$ and $w_\alpha v_1 \in \mathbf{A}^{\mathbb{Z}^-}$ and $v_2 w_\beta \in \mathbf{A}^{\mathbb{Z}^+}$. Through this definition, the indices of s are inherited from the dotted word v and are thus not explicitly prescribed.

A VS is then defined as a pair $M_{VS} = (\mathbf{A}^{\mathbb{Z}}, \Omega)$, with $\mathbf{A}^{\mathbb{Z}}$ being the space of dotted sequences, and $\Omega : \mathbf{A}^{\mathbb{Z}} \rightarrow \mathbf{A}^{\mathbb{Z}}$ defined by

$$\Omega(s) = \sigma^{F(s)}(s \oplus G(s)) \tag{16}$$

with

$$\begin{aligned} F : \mathbf{A}^{\mathbb{Z}} &\rightarrow \mathbb{Z} \\ \oplus : \mathbf{A}^{\mathbb{Z}} \times \mathbf{A}^{\mathbb{Z}} &\rightarrow \mathbf{A}^{\mathbb{Z}} \\ G : \mathbf{A}^{\mathbb{Z}} &\rightarrow \mathbf{A}^{\mathbb{Z}}, \end{aligned} \tag{17}$$

where the operator “ \oplus ” substitutes the dotted word $v_1.v_2 \in \hat{\mathbf{A}}^*$ in s with a new dotted word $\hat{v}_1.\hat{v}_2 \in \hat{\mathbf{A}}^*$ specified by G , while $F(s) = F|_{\hat{\mathbf{A}}^*}(v_1.v_2)$ determines the number of shift steps as for Moore’s generalized shifts (Carmantini et al. 2017). For a more detailed explanation about VS see Section 2.1.1 in Carmantini et al. (2017).

A nonlinear dynamical automaton (NDA) is a triple $M_{NDA} = (Y, \mathcal{P}, \Phi)$, where \mathcal{P} is a rectangular partition of the unit square $Y = [0, 1]^2 \subset \mathbb{R}^2$, that is

$$\mathcal{P} = \{D^{(i,j)} \subset Y \mid 1 \leq i \leq m, 1 \leq j \leq n, m, n \in \mathbb{N}\}, \tag{18}$$

so that each cell is defined as $D^{(i,j)} = I_i \times J_j$, with $I_i, J_j \subset [0, 1]$ being real intervals for each bi-index (i, j) , with $D^{(i,j)} \cap D^{(k,l)} = \emptyset$ if $(i, j) \neq (k, l)$, and $\bigcup_{i,j} D^{(i,j)} = Y$. The couple (Y, Φ) is a time-discrete dynamical system with phase space Y and the flow $\Phi : Y \rightarrow Y$ is a piecewise affine-linear map such that $\Phi|_{D^{(i,j)}} := \Phi^{(i,j)}$, with $\Phi^{(i,j)}$ having the following form:

$$\Phi^{(i,j)}(\mathbf{y}) = \begin{pmatrix} a_1^{(i,j)} \\ a_2^{(i,j)} \end{pmatrix} + \begin{pmatrix} \lambda_1^{(i,j)} & 0 \\ 0 & \lambda_2^{(i,j)} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \tag{19}$$

with state vector $\mathbf{y} = (y_1, y_2)$. Carmantini et al. (2017) have shown that using Gödelization any versatile shift can be mapped to a nonlinear dynamical automaton. Therefore, one can reproduce the activity of a versatile shift on the unit square Y . In order to do so, the partition (18) is given by the so called domain of dependance (DoD). The domain of dependance is a pair $(l, r) \in \mathbb{N} \times \mathbb{N}$ which defines the length of the strings on the left and right hand side of the dot in a dotted sequence that is relevant for the versatile shift to act on the phase space. The dynamics of the versatile shift is completely determined by how the string looks like in each iteration on the domain of dependance. Then, if the domain is (l, r) and if the alphabet \mathbf{A} has size m , the partition of the unit square is given by m^r intervals on the y_1 axis and m^l intervals on the y_2 axis, corresponding to cells where the NDA is defined according to the versatile shift. Finally, a *neural automaton* (NA) is an implementation of an NDA by means of a modular recurrent neural network. The full construction can be followed in “Invariants in dynamical systems” section of Carmantini et al. (2017).

The neural automaton comprises a phase space $X = [0, 1]^n$ where the two-dimensional subspace $Y = [0, 1]^2$ of the underlying NDA is spanned by only two neurons that belong to the machine configuration layer (MCL). The remainder $X \setminus Y$ is spanned by the neurons of the branch selection layer (BSL) and the linear transformation layer (LTL), both mediating the piecewise affine mapping (19). Having an NDA defined from a versatile shift, each rectangle on the partition is given by the DoD, and the action of the NDA on each rectangle depends on the particular Gödel encoding of the alphabet \mathbf{A} that has been chosen. We are interested in invariant macroscopic observables of such automata under different Gödel encodings of the alphabet.

Since we are now interested on dotted sequences over an alphabet \mathbf{A} , instead of having an invariant partition of the interval $[0, 1]$ as in Fig. 5, we will have an invariant partition of the unit square $Y = [0, 1]^2$. That is, we will have a partition in rectangles where the machine might be at certain step of the dynamics or not. Each color in that partition gives all the possible places where a particular dotted sequence of certain right and left lengths could be under a different Gödel encoding.

For example, assuming that our alphabet has $m = 3$ letters in both sides of the dotted sequence and that we are looking at words of length $l = 2$ on the left hand side of the dot, and length $r = 3$ on the right hand side of the dot, the partition would be like in Fig. 6.

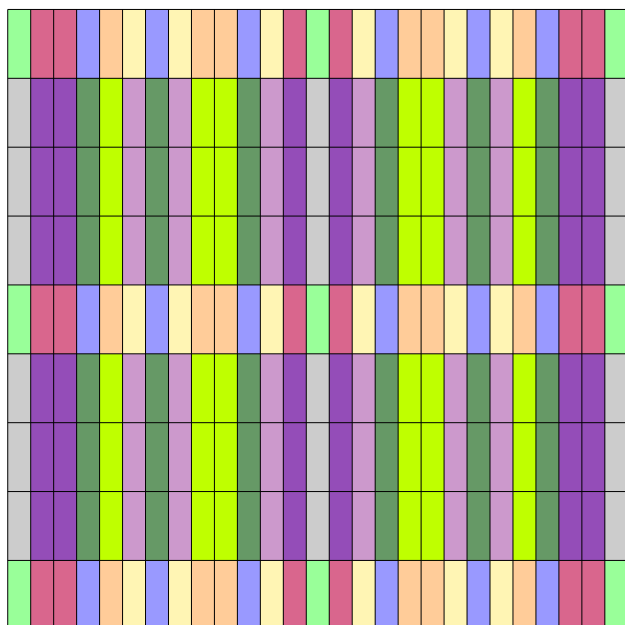


Fig. 6 Each small square corresponds to a square on the partition given by the dotted sequences of length (2, 3). The squares colored by the same color are those having the same pattern of equality, and thus, are those which can be mapped to each other under different Gödel encodings of the alphabet

Let us assume that we are considering the invariant partition for dotted sequences of length (l, r) , meaning that the left hand side has length l and the right hand side r . Then we know that the partition of the square Y is given by $E^{(i,j)} = [\frac{i}{m^l}, \frac{i+1}{m^l}] \times [\frac{j}{m^r}, \frac{j+1}{m^r}]$. Each left corner of the rectangle corresponds to the position of the Gödelization of a dotted sequence of size (l, r) . Each point $(y_1, y_2) = (\frac{i}{m^l}, \frac{j}{m^r})$ has a unique expansion on base m for its coordinates. That is, there are some c_1, \dots, c_l with $0 \leq c_i \leq m-1$ such that

$$y_1 = \frac{i}{m^l} = \frac{c_1}{m} + \frac{c_2}{m^2} + \dots + \frac{c_l}{m^l}. \tag{20}$$

These $\{c_1, \dots, c_l\}$ also define a partition of $\{1, \dots, l\}$ in the same way as given in Definition 1. Therefore $\{d_1, \dots, d_k\} \in \mathcal{P}_x \iff c_{j_1} = \dots = c_{j_k}$. This procedure similarly applies to the y_2 coordinate. Hence, the corners defining an invariant piece of the partition will be those sharing the same partition of $\{1, \dots, l\} \times \{1, \dots, r\}$. In other words, we can obtain the corners related to a given y by expanding y_1 and y_2 on base m and permuting the appearance of $0, \dots, m-1$ on the expansion.

For example, if $m = 3$ and $(l, r) = (2, 3)$, we have $3^2 \cdot 3^3 = 3^5$ rectangles. Now let us take, for instance the rectangle $[\frac{6}{3^2}, \frac{7}{3^2}] \times [\frac{10}{3^3}, \frac{11}{3^3}]$ and let us find its invariant partition. First we decompose

$$y_1 = \frac{6}{3^2} = \frac{2}{3} + \frac{0}{9} \text{ and } y_2 = \frac{10}{3^3} = \frac{1}{3} + \frac{0}{9} + \frac{1}{27}.$$

Hence a rectangle in the same invariant partition must be of the form $E^{(i,j)} = [\frac{i}{m^l}, \frac{i+1}{m^l}] \times [\frac{j}{m^r}, \frac{j+1}{m^r}]$ with $\frac{i}{3^2} = \frac{a}{3} + \frac{b}{9}$ and $\frac{j}{3^3} = \frac{c}{3} + \frac{b}{9} + \frac{c}{27}$ with $a, b, c \in \{0, 1, 2\}$ and different.⁵ This gives the following rectangles

$$\left(\frac{1}{9}, \frac{2}{9}\right) \times \left(\frac{23}{27}, \frac{24}{27}\right)$$

$$\left(\frac{2}{9}, \frac{3}{9}\right) \times \left(\frac{16}{27}, \frac{17}{27}\right)$$

$$\left(\frac{7}{9}, \frac{8}{9}\right) \times \left(\frac{3}{27}, \frac{4}{27}\right)$$

In this way we can construct the partition of the unit square given by the patterns of equality.

Invariant observables

Our aim now is to define an observable $f \in B(X)$, in the sense of “Invariants in dynamical systems” section for neural automata. That is $f : X \rightarrow \mathbb{R}$ should obey Eq. (3) where the map α_π^* corresponds to a symmetry induced by a Gödel recoding of the alphabets. Here π denotes the permutation of the alphabet needed to pass from one Gödel encoding to the other, as explained later.

Notice that in the previous discussion we were assuming that we knew the length of the strings that were encoded. However, this is not the case in practice, and may cause problems, as the length of the strings vary at each iteration. For instance, if for the alphabet $\{a, b\}$ the symbol a is mapped to 0 under certain Gödel encoding γ and the symbol b to 1, then the number $x = 1/2 \in [0, 1]$ would correspond to the word $w_r = ba^r \cdot 1a$ once we assume that the string is of length r for $r \in \mathbb{N} \cup \{0\}$. However, if we do not know the length of the encoded string, each w_r will have a different Gödel number under the Gödel encoding that sends b to 0 and a to 1, namely $\sum_{k=2}^{r-1} 1/2^k$. Thus, encoding symbols with the number 0 makes some strings indistinguishable under Gödel recoding, because having no symbols is interpreted as having the symbol encoded by 0 as many times as we want. This issue can be easily avoided by adding one symbol \sqcup to the alphabet, which will be

⁵ Here we are assuming that both the alphabet A and the Gödel Encoding is the same in both sides of the dot, otherwise we would have more freedom and obtain more intervals, but the procedure works anyway.

interpreted as a blank symbol, and will always be forced to be encoded as 0 by any Gödel encoding.

Suppose that we have an NDA defined from a versatile shift under the condition that the blank symbol \sqcup has been added to the alphabet \mathbf{A} representing the blank symbol and that is mapped to 0 under any Gödel encoding.⁶ We will assume that \mathbf{A} has m symbols after adding the blank symbol (that is, we had $m - 1$ symbols before). Then for any pair $(r, l) \in \mathbb{N} \times \mathbb{N}$, we can divide the unit square Y into the rectangle partition given by

$$\mathcal{R} = \{E^{(i,j)} \subset Y \mid 1 \leq i \leq m^r, 1 \leq j \leq m^l\}. \tag{21}$$

Next, we extend this partition of the phase space of the NDA, that equals the subspace of the machine configuration layer of the larger NA, to the entire phase space of the neural automaton. This is straightforwardly achieved by defining another partition

$$\mathcal{Q} = \{E^{(i,j)} \times [0, 1]^{n-2} \subset X \mid 1 \leq i \leq m^r, 1 \leq j \leq m^l\}. \tag{22}$$

Now, for each left corner $(x_1^{(i,j)}, x_2^{(i,j)}) \in E^{(i,j)}$ we find their pattern of equality \mathcal{P}_{ij} , assuming that the permutation is taking place just on the symbols $\{2, \dots, m\}$ (as the first symbol has to be mapped to 0 under any encoding).

Let us suppose that $\{\mathcal{P}_{ij_1}, \dots, \mathcal{P}_{ij_s}\}$ are all the different appearing patterns of equality and we define the indicator functions $\chi_k : X \rightarrow \{0, 1\}$ as

$$\chi_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in E^{(i,j)} \times [0, 1]^{n-2} \text{ and } \mathcal{P}_{ij} = \mathcal{P}_{ij_k} \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

for $\mathbf{x} \in X$. Then, we can choose $c_1, \dots, c_s \in \mathbb{R}$ to be s different real numbers and define a macroscopic observable $f : X \rightarrow \mathbb{R}$ as a step function

$$f(\mathbf{x}) = \sum_{k=1}^s c_k \chi_k(\mathbf{x}). \tag{24}$$

Clearly, we have $f \in B(X)$.

Our aim is to show that this observable is invariant under the symmetry group $S_{m-1} \times S_{m-1}$ of the dynamical system (X, Φ) given by the neural automaton in Eq. (19), where S_{m-1} denotes the symmetric group on $m - 1$ elements. First of all, we must show that $S_{m-1} \times S_{m-1}$ is a symmetry of the neural automaton.

Before doing this, we will define an auxiliary map. Let $\pi = (\pi_1, \pi_2) \in S_m \times S_m$ be any element of the product that fixes 1 (on the set $\{1, 2, \dots, m\}$ where S_m acts). Notice that the elements of S_m fixing the first element form a subgroup

⁶ In order to make things simpler we will assume that we have the same alphabet on the stack and the input symbols. This can always be assumed considering the union of both alphabets if needed.

of S_m that is isomorphic to S_{m-1} . Let now \mathbf{x} be any point in X . Let us consider $\mathbf{y} = (y_1, y_2)$ the first two coordinates of \mathbf{x} given by the activations of the machine configuration layer of the NA. Then, we can check in which of the intervals of the partition \mathcal{R} is, say $(y_1, y_2) \in E^{(i,j)} = \left[\frac{i}{m^r}, \frac{i+1}{m^r}\right) \times \left[\frac{j}{m^l}, \frac{j+1}{m^l}\right)$. We can therefore compute the expansion on base m of each corner and take the coefficients we get as words over the alphabet $\mathbf{M} = \{0, 1, \dots, m - 1\}$, say $c_1 \dots c_l \in \mathbf{M}^l$ and $d_1, \dots, d_r \in \mathbf{M}^r$. Then, we compute $g_{\pi_1}(c_1 \dots c_l)$ and $g_{\pi_2}(d_1 \dots d_r)$ and we encode these words by the canonical Gödel encoding (that is, the one given by the identity map on \mathbf{M}). Thus, we obtain a new corner of some rectangle in our partition of the phase space, say $E^{(i',j')} = \left[\frac{i'}{m^r}, \frac{i'+1}{m^r}\right) \times \left[\frac{j'}{m^l}, \frac{j'+1}{m^l}\right)$. We now define a map $\rho_\pi : Y \rightarrow Y$ by $\rho_\pi(y_1, y_2) = \left(y_1 + \frac{i'-i}{m^r}, y_2 + \frac{j'-j}{m^l}\right)$. This map can obviously be extended to a map from X to X being the identity on the rest of the coordinates. Abusing notation we also refer to ρ_π as to this map. Informally speaking, the map ρ_π rigidly permutes the squares on the partition \mathcal{R} according to the action of g_{π_1} and g_{π_2} on the words representing the corners.

Now, we can define $\alpha_\pi : B(X) \rightarrow B(X)$ as follows. For any $f \in B(X)$, we define

$$\alpha_\pi(f)(\mathbf{x}) = f(\rho_\pi(\mathbf{x})). \tag{25}$$

It is not difficult to check that if $\pi, \gamma \in S_{m-1} \times S_{m-1}$ are two group elements, then $\alpha_{\gamma\pi}(f) = (\alpha_\gamma \circ \alpha_\pi)(f)$ so that $S_{m-1} \times S_{m-1}$ is a symmetry of the system.

Thus, we obtain finally our main result.

Theorem 2 *Let $f \in B(X)$ be a macroscopic observable on the space space of a neural automaton as defined in (24). Then f is invariant under the symmetric group $S_{m-1} \times S_{m-1}$ of Gödel recodings of the automaton’s symbolic alphabet.*

It is worth mentioning that this procedure gives infinitely many different invariant observables. In fact, any choice of $(r, l) \in \mathbb{N} \times \mathbb{N}$ gives a thinner invariant partition, and respectively, a sharper observable.

Neurolinguistic application

As an instructive example we consider a toy model of syntactic language processing as often employed in computational psycholinguistics and computational neurolinguistics (Arbib and Caplan 1979; Crocker 1996; beim Graben and Drenhaus 2012; Hale et al. 2022; Lewis 2003).

In order to process the sentence given by beim Graben and Pothast (2014) in example 2, linguists often derive a

Table 1 Sequence of state transitions of the versatile shift processing the well-formed string from example 2, i.e. NP V NP. The operations are indicated as follows: “predict (X)” means prediction according to rule (X) of the context-free grammar; attach means cancellation of successfully predicted terminals both from stack and input; and “accept” means acceptance of the string as being well-formed

| Time | State | Operation |
|------|---------------------|--------------|
| 0 | S.NP V NP | Predict (26) |
| 1 | VP NP .NP V NP | Attach |
| 2 | VP .V NP | Predict (27) |
| 3 | NP V .V NP | Attach |
| 4 | NP .NP | Attach |
| 5 | $\epsilon.\epsilon$ | Accept |

context-free grammar (CFG) from a phrase structure tree (Hopcroft and Ullman 1979).

Example 2 the dog chased the cat

In our case, the CFG consists of *rewriting rules*

$$S \rightarrow NPVP \tag{26}$$

$$VP \rightarrow VNP \tag{27}$$

$$NP \rightarrow \text{thedog} \tag{28}$$

$$V \rightarrow \text{chased} \tag{29}$$

$$NP \rightarrow \text{thecat} \tag{30}$$

where the left-hand side always presents a nonterminal symbol to be expanded into a string of nonterminal and terminal symbols at the right-hand side. Omitting the lexical rules (28 – 30), we regard the symbols NP, V, denoting ‘noun phrase’ and ‘verb’, respectively, as terminals and the symbols S (‘sentence’) and VP (‘verbal phrase’) as nonterminals.

Then, a versatile shift processing this grammar through a simple top down recognizer (Hopcroft and Ullman 1979) is defined by

$$\begin{aligned} S.aVP &\mapsto NP.a \\ VP.aNP &\mapsto V.a \\ a.a &\mapsto \epsilon.\epsilon \end{aligned} \tag{31}$$

where the left-hand side of the tape is now called ‘stack’ and the right-hand side ‘input’. In (31) *a* stands for an arbitrary input symbol. Note the reversed order for the stack left of the dot. The first two operations in (31) are *predictions* according to a rule of the CFG while the last one is an *attachment* of subsequent input with already predicted material.

This machine then *parses* the well formed sentence NP V NP as shown in Table 1 from beim Graben and Potthast (2014). We reproduce this table here as Table 1.

Once we obtained the versatile shift, an NA simulating it can be generated. When we do so, we chose a particular Gödel encoding of the symbols. Suppose we chose the following two Gödelizations $\gamma = (\gamma_1, \gamma_2)$ and $\delta = (\delta_1, \delta_2)$ that are given by

$$\begin{aligned} \gamma_1 : \{\sqcup, NP, V\} &\rightarrow \{0, 1, 2\} \\ \sqcup &\mapsto 0 \\ NP &\mapsto 1 \\ V &\mapsto 2 \end{aligned}$$

$$\begin{aligned} \gamma_2 : \{\sqcup, NP, V, VP, S\} &\rightarrow \{0, 1, 2, 3, 4\} \\ \sqcup &\mapsto 0 \\ NP &\mapsto 1 \\ V &\mapsto 2 \\ VP &\mapsto 3 \\ S &\mapsto 4 \end{aligned}$$

on the one hand, and by

$$\begin{aligned} \delta_1 : \{\sqcup, NP, V\} &\rightarrow \{0, 1, 2\} \\ \sqcup &\mapsto 0 \\ NP &\mapsto 2 \\ V &\mapsto 1 \end{aligned}$$

$$\begin{aligned} \delta_2 : \{\sqcup, NP, V, VP, S\} &\rightarrow \{0, 1, 2, 3, 4\} \\ \sqcup &\mapsto 0 \\ NP &\mapsto 4 \\ V &\mapsto 3 \\ VP &\mapsto 1 \\ S &\mapsto 2 \end{aligned}$$

on the other hand. Defining the step function $f : X \rightarrow \mathbb{R}$ as in (24) after choosing $(l, r) = (2, 3)$ and the c_i -s randomly. The neural automaton consists of $n = 72$ neurons, i.e. the phase space is given by the hypercube $X = [0, 1]^{72}$. Running the neural network with both encodings and computing the step function f on each iteration $i = 1, \dots, 6$, we see in Fig. 7 that f is indeed invariant under Gödel recoding.

The step function clearly distinguishes among different states (where here by “different” we mean with different patterns of equality), but returns the same value for the states corresponding to the same pattern of equality, that is, states that differ on the Gödel encodings, as desired.

In contrast, if we use Amari’s observable Eq. (9) for the same simulation, we obtain a very different picture, showing that this observable is not invariant under Gödel recoding, as shown in Fig. 8. Obviously, this observable

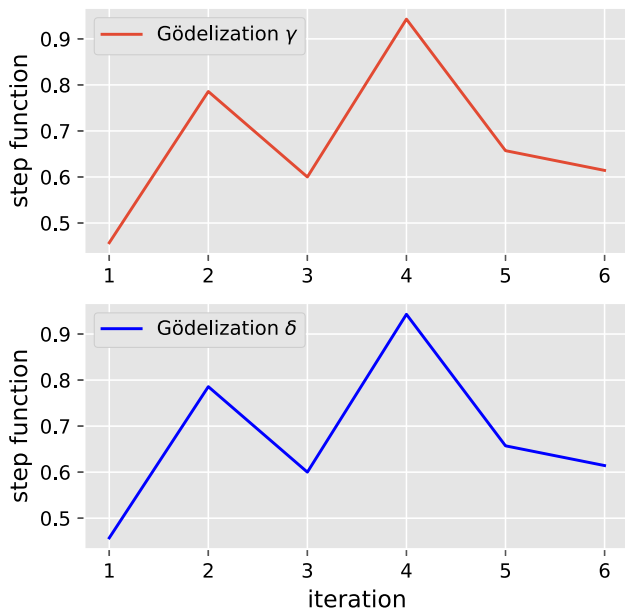


Fig. 7 The macroscopic observable f , given by the step function (24) is invariant under Gödel recoding. The figure shows the result of ‘measuring’ f to a neural automaton encoded by γ on top and to the same machine encoded by δ below

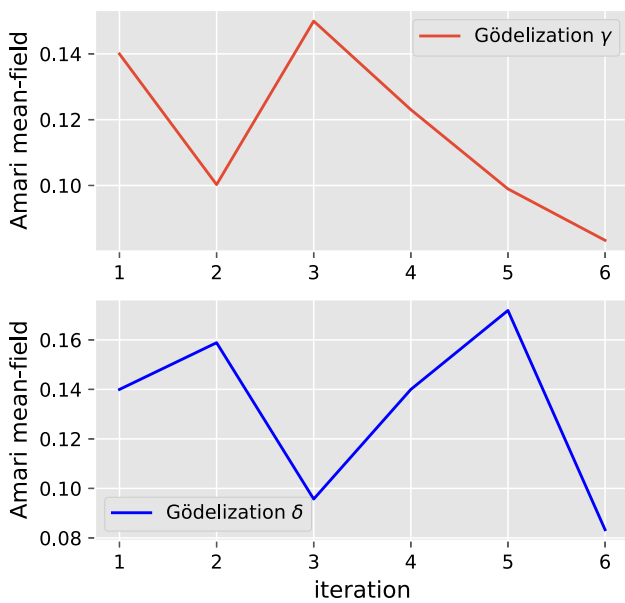


Fig. 8 Amari’s mean-field observable Eq. (9) of the neural automaton under two different Gödel encodings γ and δ

strongly depends on the particular Gödel encoding we have chosen.

Discussion

In this study we have presented a way of finding particular macroscopic observables for nonlinear dynamical systems that are generated by Gödel encodings of symbolic dynamical systems, such as nonlinear dynamical automata (NDA: beim Graben et al. (2000, 2004, 2008); beim Graben and Potthast (2014)) and their respective neural network implementation, namely, neural automata (NA: Carmantini et al. (2017)). Specifically, we have investigated under which circumstances such observables could be invariant under any particular choice for the Gödel encoding.

When mapping symbolic dynamics to a real phase space, the numbering of the symbols is usually arbitrary. Therefore, it makes sense to ask which information of the dynamics is preserved or can be recovered from what we see in phase space under the different possible options. In this direction, we have provided a complete characterisation of the strings that are and are not distinguishable after certain Gödel encoding in terms of *patterns of equality*. We have proven a partition theorem for such invariants.

In the concrete case of NA constructed as in Carmantini et al. (2017), which can emulate any Turing Machine, we have a dynamical system for a neural automaton. This system completely depends on the choice of the Gödel numbering for the symbols on the alphabet of the NA. Based on the invariant partition mentioned before, we were able to define a macroscopic observable that is invariant under any Gödel recoding. In fact, by the way we define this observable, the definition is based on an invariant partition according to the length of the strings on the left and right hand side of a dotted sequence comprising the machine tape of the NA. This means that each choice of the length of those strings provides a sharper invariant, making strings with different patterns of equality completely distinguishable. It is also important to mention that macroscopic observables in general are not invariant under Gödel recoding. As a particular example, we computed the mean neural network activation originally suggested by Amari (1974) and later employed by Carmantini et al. (2017) as a modeled “synthetic ERP” (Barrès et al. 2013) in neurocomputing.

In fact, any observable that is invariant under Gödel recoding must be equally defined for points on the phase space corresponding to Gödelizations of strings sharing the same patterns of equality. This could probably provide an important constraint in the finding of other invariant macroscopic observables.

Theoretically, one could run neural automaton under all (or many) possible Gödel encodings and check which observables are preserved by the dynamics and which are

not. This could provide important information about the performance of the neural network architecture that is intrinsic of the dynamical system, and not dependant on the choice of the numbering for the codification of the symbols. In practice, the computation of all the permutations of the alphabet grows with the factorial of the alphabet’s cardinality, and the computation of invariant partitions even with powers of that number for longer strings. This, of course, would present some practical constraints for large alphabets and sharp invariant observables.

Our results could be of substantial importance for any kind of related approaches in the field of computational cognitive neurodynamics. All models that rely upon the representation of symbolic mental content by means of high-dimensional activation vectors as training patterns for (deep) neural networks (Arbib 1995; LeCun et al. 2015; Hertz et al. 1991; Schmidhuber 2015), such as vector symbolic architectures (Gayler 2006; Schlegel et al. 2021; Smolensky 1990, 2006; Mizraji 1989, 2020) in particular, are facing the problems of arbitrary symbolic encodings. As long as one is only interested in building inference machines for artificial intelligence, this does not really matter. However, when activation states of neural network simulations have to be correlated with real-word data from experiments in the domains of human or animal cognitive neuroscience and psychology, the given encoding may play a role. Thus, the investigation of invariant observables in regression analyses and statistical modeling becomes mandatory for avoiding possible confounds that could result from a particularly chosen encoding.

These results also have implications in Mathematical and Computational Neuroscience, where the aim is to explain by means of mathematical theories and computational modelling neurophysiological processes as observed in in-vitro and in-vivo experiments via instrumentation devices. Our results forces us to consider the possibility as to what extent (if any) that observations, which motivate the development of models in the literature (e.g Spiking models), are epiphenomenon? To conclude, we express the hope that our study paves the way towards a more a comprehensive research in computational cognitive neurodynamics, mathematical and computational neuroscience where the study of macroscopic observations and its invariant formulation can lead to interesting new insights.

Reproducibility

All numerical simulations that have been presented in “Neurolinguistic application” section may be reproduced using the code available at the GitHub repository https://github.com/TuringMachinegun/Turing_Neural_Networks. The repository contains the code to build the architecture of a neural automaton as introduced in (Carmantini et al.

(2017)) together with particular examples. The code that computes the invariant partitions given by equality patterns can also be found in the repository. The code allows the user to implement various observables (e.g step function, Amari’s observable) in order to test further cases, exploit and further develop our framework.

Appendix: Proofs of lemmata and theorems

Proof of Lemma 1 The ordering γ itself induces the isomorphism between both graphs. Namely let $g_\gamma : T_\gamma \rightarrow T$ be such that if $s = a_{i_1} a_{i_2} a_{i_3} \dots \in T_\gamma$ then

$$g_\gamma(s) = \gamma(a_{i_1})\gamma(a_{i_2})\gamma(a_{i_3}) \dots$$

which clearly belongs to T . We must show that it defines a bijection between vertices and that preserves incidence. It is easy to prove that it is a bijection. Namely if $w = a_{i_1} a_{i_2} \dots a_{i_n}$ and $u = b_{i_1} \dots b_{i_n}$ are any two vertices of the tree T_γ then $g_\gamma(w) = g_\gamma(u)$ implies that both strings must have the same length, hence $n = k$. And since $\gamma(a_{i_j}) = \gamma(b_{i_j})$ and γ is a bijection, we must have $a_{i_j} = b_{i_j}$ so that $w = u$. Moreover, for any $v = l_1 \dots l_n \in T$ there is $z = a_{\gamma^{-1}(l_1)} \dots a_{\gamma^{-1}(l_n)}$ which is clearly mapped to v through g_γ . The only thing that is left to show is that g_γ preserves incidence. That is, that given $w \in \mathbf{A}^*$ and $a \in \mathbf{A}$, then $g_\gamma(wa) = g_\gamma(w)a$ for some $l \in \{0, \dots, m - 1\}$. But this is also clear from the definition of g_γ . \square

Proof of Lemma 2 Let us suppose that $d(p, q) \leq \frac{1}{m^n}$. This means that at least the first n symbols in both strings are equal. Then, if ψ is a Gödel encoding defined by the assignment $\gamma : \mathbf{A} \rightarrow \mathbf{M}$ we have that

$$\psi(p) = \sum_{i=1}^{\infty} \gamma(a_i) \frac{1}{m^i} = \sum_{i=1}^n \gamma(a_i) \frac{1}{m^i} + \sum_{i=n+1}^{\infty} \gamma(a_i) \frac{1}{m^i}.$$

Let us put $r = \sum_{i=1}^n \gamma(a_i) \frac{1}{m^i}$. Now, since $\gamma(a_i) \leq m - 1$

$$\begin{aligned} r &= \frac{a_1 m^{n-1} + a_2 m^{n-2} + \dots + a_{n-1} m + a_n}{m^n} \leq \\ &\leq \frac{m^n - m^{n-1} + m^{n-1} - \dots - d + d - 1}{m^n} = \frac{m^{n-1}}{m^n}. \end{aligned}$$

So $r = \frac{k}{m^n}$ for some $k = 0, \dots, m^n - 1$. Since $\sum_{i=n+1}^{\infty} \gamma(a_i) \frac{1}{m^i} < \frac{1}{m^{n+1}}$, we get that $\psi(p) \in [\frac{k}{m^n}, \frac{k+1}{m^n})$. Since q is

equal to p on at least the first n strings we will also have $\psi(q) = r + \sum_{i=n+1}^{\infty} \gamma(b_i) \frac{1}{m^i}$ and by the same reason it will be

on the same interval. For the other implication, if we have two real numbers $\psi(p)$ and $\psi(q)$ after encoding some infinite strings p and q , we want to show that if they are on some interval of the type $[\frac{k}{m^n}, \frac{k+1}{m^n})$, then they have the same prefix of at least length n . We can always write those numbers as $\psi(p) = \frac{k}{m^n} + r_1$ and $\psi(q) = \frac{k}{m^n} + r_2$ with $r_1, r_2 < \frac{1}{m^{n+1}}$. If we write the number k in its m -adic expansion, it

will be uniquely determined by $l_1, \dots, l_n \in \{0, \dots, m - 1\}$, and each number r_i can be written as a series by $r_i = \sum_{j=n+1}^m l_j \frac{1}{m^j}$, for $i = 1, 2$.

Taking the inverse images of each l_i , that is $\gamma^{-1}(l_i) = a_i$ we will obtain that $p = a_1 \dots a_n a_{n+1} \dots$ and $q = a_1 \dots a_n b_{n+1} \dots$. That is, they are at least at distance $\frac{1}{m^n}$. \square

Proof of Theorem 1 Let us first assume that given u there exists some π such that its induced automorphisms maps w to u . The condition Eq. (13) is clear, by Lemma 2. Then, we can write $u = c_1 \dots c_n$. Now, we know that $g_\pi(w) = a_{\pi(1)} a_{\pi(2)} \dots a_{\pi(n)} = c_1 \dots c_n = u$. Then

$$\begin{aligned} \{j_1, \dots, j_k\} \in P_w &\iff a_{i_{j_1}} = \dots = a_{i_{j_k}} \\ &\iff a_{\pi(i_{j_1})} = \dots = a_{\pi(i_{j_k})} \\ &\iff c_{i_{j_1}} = \dots = c_{i_{j_k}} \\ &\iff \{j_1, \dots, j_k\} \in P_u. \end{aligned}$$

To show the other direction, it suffices to define π so that $g_\pi(w) = u$. That is, if $u = c_1 \dots c_n$, let us define $\pi(a_i) = c_i$ and let us send the $a_j \in \mathbf{M}$ not appearing in w to the c_j -s not appearing in u in a bijective way. This can be done, it is well defined by condition Eq. (14), and it defines a bijection on \mathbf{M} by construction. \square

Proof of Theorem 2 We have to show that equation (3) is satisfied. Namely, we have to show that $f(\alpha_\pi^*(\mathbf{x})) = f(\mathbf{x})$ for each $\pi \in S_{m-1} \times S_{m-1}$ and $\mathbf{x} \in X$. Note that by definition, $\alpha_\pi^* = \rho_\pi$ in our case. Therefore, we must show that $f(\rho_\pi(\mathbf{x})) = f(\mathbf{x})$. Let $\pi = (\pi_1, \pi_2) \in S_{m-1} \times S_{m-1}$ and $(y_1, y_2) \in Y$. Then, there is some $E^{(ij)}$ for which $(x, y) \in E^{(ij)}$. Let us denote by $\pi' = (\pi'_1, \pi'_2) \in S_m \times S_m$ the permutation fixing 1 and sending $\pi'(i) = \pi(i) + 1$ for $i = 2, \dots, m - 1$ (namely, the permutation fixing the first letter and permuting the rest as π_i permutes the $m - 1$ letters for $i = 1, 2$ respectively). Note that since we have enlarged our alphabet with the \square symbol, and since both our original encoding and the one permuted by π send this symbol to 0, if (y_1, y_2) is decoded as (c_1, \dots, c_l) and (d_1, \dots, d_r) , the possible 0s appearing at the end of each encoding (indicating that the string has smaller length than l and/or r) will remain being 0-s, and therefore the point will not be mapped to a point encoding longer strings. After applying ρ_π , we will obtain $\rho_\pi((y_1, y_2)) \in E^{(i'j')}$ for some i', j' . However, since ρ_π is defined through g_{π_1} and g_{π_2} both $E^{(ij)}$ and $E^{(i'j')}$ belong to the same P_{ij} . That is, they have the same pattern of equality. Hence, by the definition of f we obtain that $f(\mathbf{x}) = f(\rho_\pi(\mathbf{x}))$, and we are done. \square

Acknowledgements SR acknowledges support from Ikerbasque (The Basque Foundation for Science), the Basque Government through the BERC 2022-2025 program and by the Ministry of Science and Innovation: BCAM Severo Ochoa accreditation CEX2021-001142-S / MICIN / AEI / 10.13039/501100011033 and through project RTI2018-093860-B-C21 funded by (AEI/FEDER, UE) and acronym MathNEURO. JUA acknowledges support from the Spanish Government, grants PID2020-117281GB-I00 and PID2019-107444GA-I00, partly with European Regional Development Fund (ERDF), and the Basque Government, grant IT1483-22.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

Amari SI (1974) A method of statistical neurodynamics. *Kybernetik* 14:201–215

Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004) An integrated theory of the mind. *Psychol Rev* 111(4):1036–1060

Arbib MA (ed) (1995) *The handbook of brain theory and neural networks*. MIT Press, Cambridge

Arbib MA, Caplan D (1979) Neurolinguistics must be computational. *Behav Brain Sci* 2(03):449–460

Atmanspacher H, Beim Graben P (2007) Contextual emergence of mental states from neurodynamics. *Chaos Complexity Lett* 2(2/3):151–168

Barrès V, Arbib ASM (2013) Synthetic event-related potentials: A computational bridge between neurolinguistic models and experiments. *Neural Netw* 37:66–92

Beim Graben P (2008) Foundations of neurophysics. In: Zhou C, Thiel M, Kurths J, Graben PB (eds) *Lectures in supercomputational neuroscience dynamics in complex brain networks* springer complexity series. Springer, Berlin

Beim Graben P, Atmanspacher H (2006) Complementarity in classical dynamical systems. *Found Phys* 36(2):291–306

Beim Graben P, Atmanspacher H (2009) Extending the philosophical significance of the idea of complementarity. In: Atmanspacher H, Primas H (eds) *Recasting reality Wolfgang Pauli’s philosophical ideas and contemporary science*. Springer, Berlin

Beim Graben P, Drenhaus H (2012) *Computationelle neurolinguistik*. *Z. Germanistische Linguistik* 40(1):97–125

Beim Graben P, Potthast R (2009) Inverse problems in dynamic cognitive modeling. *Chaos* 19(1):015103

Beim Graben P, Potthast R (2014) Universal neural field computation. In: Potthast R, Wright JJ, Coombes S, Beim Graben P (eds) *Neural fields theory and applications*. Springer, Berlin

Beim Graben P, Rodrigues S (2013) A biophysical observation model for field potentials of networks of leaky integrate-and-fire neurons. *Front Comput Neurosci* 6(100):10042

Beim Graben P, Rodrigues S (2014) On the electrodynamics of neural networks. In: Potthast R, Wright JJ, Coombes S, Beim Graben P (eds) *Neural fields theory and applications*. Springer, Berlin

Beim Graben P, Liebscher T, Saddy JD (2000) Parsing ambiguous context-free languages by dynamical systems: disambiguation and phase transitions in neural networks with evidence from event-related brain potentials (ERP). In: Jokinen K, Heylen D, Nijholt A (eds) *Learning to Behave Universiteit Twente Enschede, TWLT 18. Internalising Knowledge, Enschede*

Beim Graben P, Jurish B, Saddy D, Frisch S (2004) Language processing by dynamical systems. *Int J Bifurcat Chaos* 14(2):599–621

- Beim Graben P, Gerth S, Vasishth S (2008) Towards dynamical system models of language-related brain potentials. *Cogn Neurodyn* 2(3):229–255
- Beim Graben P, Barrett A, Atmanspacher H (2009) Stability criteria for the contextual emergence of macrostates in neural networks. *Netw Comput Neural Syst* 20(3):178–196
- Boston MF, Hale JT, Patil U, Kliegl R, Vasishth S (2008) Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam sentence corpus. *J Eye Mov Res* 2(1):1–12
- Brouwer H, Crocker MW (2017) On the proper treatment of the P400 and P600 in language comprehension. *Front Psychol* 8:1327
- Brouwer H, Hoeks JCJ (2013) A time and place for language comprehension: mapping the N400 and the P600 to a minimal cortical network. *Front Human Neurosci* 7(758):4572
- Brouwer H, Fitz H, Hoeks J (2012) Getting real about Semantic Illusions: rethinking the functional role of the P600 in language comprehension. *Brain Res* 1446:127–143
- Brouwer H, Crocker MW, Venhuizen NJ, Hoeks JCJ (2017) A neurocomputational model of the N400 and the P600 in language processing. *Cogn Sci* 41(S6):1318–1352
- Brouwer H, Delogu F, Venhuizen NJ, Crocker MW (2021) Neurobehavioral correlates of surprisal in language comprehension: a neurocomputational model. *Front Psychol* 12:110
- Carmantini GS, Beim Graben P, Desroches M, Rodrigues S (2017) A modular architecture for transparent computation in recurrent neural networks. *Neural Netw* 85:85–105
- Cleeremans A, Servan-Schreiber D, McClelland JL (1989) Finite state automata and simple recurrent networks. *Neural Comput* 1(3):372–381
- Crocker MW (1996) Computational psycholinguistics studies in computational psycholinguistics. Kluwer, Dordrecht
- Davidson DJ, Martin AE (2013) Modeling accuracy as a function of response time with the generalized linear mixed effects model. *Acta Physiol (Oxf)* 144(1):83–96
- Elman JL (1990) Finding structure in time. *Cogn Sci* 14:179–211
- Elman JL (1991) Distributed representations, simple recurrent networks, and grammatical structure. *Mach Learn* 7:195–225
- Emch G (1964) Coarse-graining in Liouville space and master equation. *Helv Phys Acta* 37:532–544
- Frank SL, Otten LJ, Galli G, Vigliocco G (2015) The ERP response to the amount of information conveyed by words in sentences. *Brain Lang* 140:1–11
- Frisch S, Beim Graben P, Schlesewsky M (2004) Parallelizing grammatical functions: P600 and P345 reflect different cost of reanalysis. *Int J Bifurcat Chaos* 14(2):531–549
- Friston KJ, Mechelli A, Turner R, Price CJ (2000) Nonlinear responses in fMRI: the balloon model, Volterra kernels, and other hemodynamics. *Neuroimage* 12(4):466–477
- Gayler RW (2006) Vector symbolic architectures are a viable alternative for Jackendoff's challenges. *Behav Brain Sci* 29:78–79. <https://doi.org/10.1017/S0140525X06309028>
- Gazzaniga MS, Ivry RB, Mangun GR (eds) (2002) Cognitive neuroscience. The Biology of the Mind, New York
- Graves A, Wayne G, Danihelka I (2014) Neural Turing machines. [arxiv:1410.5401](https://arxiv.org/abs/1410.5401) [cs.ne], Google DeepMind
- Hale JT, Lutz DE, Luh WM, Brennan JR (2015) Modeling fMRI time courses with linguistic structure at various grain sizes. In: Proceedings of the 2015 workshop on cognitive modeling and computational linguistics, North American Association for Computational Linguistics, Denver
- Hale JT, Campanelli L, Li J, Bhattasali S, Pallier C, Brennan JR (2022) Neurocomputational models of language processing. *Ann Rev Linguist* 8(1):427–446
- Hepp K (1972) Quantum theory of measurement and macroscopic observables. *Helv Phys Acta* 45(2):237–248
- Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. In: Lecture notes of the Santa FE institute studies in the science of complexity. Perseus Books, Cambridge
- Hopcroft JE, Ullman JD (1979) Introduction to automata theory languages, and computation. Addison-Wesley, Menlo Park
- Huyck CR (2009) A psycholinguistic model of natural language parsing implemented in simulated neurons. *Cogn Neurodyn* 3(4):317–330
- Jauch JM (1964) The problem of measurement in quantum mechanics. *Helv Phys Acta* 37:293–316
- Jirsa VK, Jantzen KJ, Fuchs A, Kelso JAS (2002) Spatiotemporal forward solution of the EEG and MEG using network modeling. *IEEE Trans Med Imag* 21(5):493–504
- Kelso JAS (1995) Dynamic patterns. MIT Press, Cambridge
- Krifka M (1995) The semantics and pragmatics of polarity items. *Linguist Anal* 25:209–257
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
- Lewis RL (2003) Computational psycholinguistics. In: Encyclopedia of cognitive science. Macmillan Reference Ltd., London
- Lewis RL (1998) Reanalysis and limited repair parsing: leaping off the garden path. In: Ferreira F, Fodor JD (eds) Reanalysis in sentence processing. Kluwer, Amsterdam
- Lewis RL, Vasishth S (2006) An activation-based model of sentence processing as skilled memory retrieval. *Cogn Sci* 29:375–419
- Lind B, Marcus B (1995) An introduction to symbolic dynamics and coding. Cambridge University Press, Cambridge
- Martínez-Cañada P, Ness TV, Einevoll G, Fellin T, Panzeri S (2021) Computation of the electroencephalogram (EEG) from network models of point neurons. *PLoS Comput Biol* 17(4):1–41
- Mazzoni A, Panzeri S, Logothetis NK, Brunel N (2008) Encoding of naturalistic stimuli by local field potential spectra in networks of excitatory and inhibitory neurons. *PLoS Comput Biol* 4(12):e1000239
- McCulloch WS, Pitts W (1943) A logical calculus of ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133
- McMillan B (1953) The basic theorems of information theory. *Ann Math Stat* 24:196–219
- Mizraji E (1989) Context-dependent associations in linear distributed memories. *Bull Math Biol* 51(2):195–205
- Mizraji E (2020) Vector logic allows counterfactual virtualization by the square root of NOT. *Logic J IGPL* 25:463
- Moore C (1990) Unpredictability and undecidability in dynamical systems. *Phys Rev Lett* 64:2354
- Moore C (1991) Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity* 4:199
- Nivre J (2008) Algorithms for deterministic incremental dependency parsing. *Comput Linguist* 34(4):513–553
- Rabinovich MI, Varona P (2018) Discrete sequential information coding: heteroclinic cognitive dynamics. *Front Comput Neurosci* 12:73
- Rabinovich M, Friston K, Varona P (eds) (2012) Principles of brain dynamics: global state interactions. MIT Press, Cambridge
- Rabovsky M, McRae K (2014) Simulating the n400 ERP component as semantic network error: insights from a feature-based connectionist attractor model of word meaning. *Cognition* 132(1):68–89
- Rabovsky M, Hansen SS, McClelland JL (2018) Modelling the N400 brain potential as change in a probabilistic representation of meaning. *Nat Human Behav* 2:693
- Ratcliff R (1978) A theory of memory retrieval. *Psychol Rev* 85(2):59–108
- Ratcliff R, McKoon G (2007) The diffusion decision model: theory and data for two-choice decision tasks. *Neural Comput* 20(4):873–922

- Schlegel K, Neubert P, Protzel P (2021) A comparison of vector symbolic architectures. *Artif Intell Rev* 55:4523
- Schmidhuber J (2015) Deep learning in neural networks: An overview. *Neural Netw* 61:85–117
- Sewell GL (2002) Quantum mechanics and its emergent macrophysics. Princeton University Press, Princeton
- Siegelmann HT, Sontag ED (1995) On the computational power of neural nets. *J Comput Syst Sci* 50(1):132–150
- Smolensky P (1986) Information processing in dynamical systems: foundations of harmony theory. In: Rumelhart DE, McClelland JL (eds) *The PDP Research Group parallel distributed processing: explorations in the microstructure of cognition*. MIT Press, Cambridge
- Smolensky P (1990) Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif Intell* 46(1–2):159–216
- Smolensky P (2006) Harmony in linguistic cognition. *Cogn Sci* 30:779–801
- Stabler EP (2011) Top-down recognizers for MCFGs and MGs. In: *Proceedings of the 2nd workshop on cognitive modeling and computational linguistics*. Association for Computational Linguistics, Portland, pp 39 – 48
- Stephan KE, Harrison LM, Penny WD, Friston KJ (2004) Biophysical models of fMRI responses. *Curr Opin Neurobiol* 14:629–635
- van Gelder T (1998) The dynamical hypothesis in cognitive science. *Behav Brain Sci* 21(05):615–628
- Wilson HR, Cowan JD (1972) Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys J* 12(1):1–24

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.