

827 Appendices

828 The appendices provide proofs of the theorems stated in the main body, as well as more detailed
829 exposition of preliminary notions, and illustrative figures. It is structured as a supplemental body of work
830 which can be read from top to bottom, and which gives a detailed presentation of Metric Automata
831 Theory and its main results. While the main body gives a big picture overview of the key notions and
832 results, the appendices aim to serve as a foundational text, showcasing how Metric Automata Theory
833 can be used to develop new theories and draw novel insights about RNN architectures—in addition to
834 providing full proofs of all results stated in the main body.

835 **Appendix A** provides standard *preliminary notions*, required for later sections and in particular for
836 proving our results.

837 **Appendix B** presents the *foundations of Metric Automata Theory (MAT)*, which build on several
838 different fields—metric spaces, dynamical systems, algebraic and classic automata theory. Also
839 establishing novel and fundamental connections across such fields. We prove Theorem 1 in this
840 appendix.

841 **Appendix C** introduces the novel notion of ϵ -robust dynamics, which allows us to argue about
842 real-world floating point implementations of models. It also describes numerical and parametrisation
843 stability properties of systems, thus going beyond the phenomena which can be described by discrete
844 systems. We provide proofs of Theorem 2 and Theorem 5.

845 **Appendix D** employs Metric Automata Theory and its connection to Algebraic Automata Theory to
846 show a collection of expressivity results in the η -finite setting, including Theorems 3, 4, 6 and 7.

847 **Appendix E** explores the setting of *Geometrically-Constrained Systems (GCS)*, in connection to
848 the empirical length-generalisation capabilities of Mamba, which go beyond its finite-precision
849 expressivity. We give proofs of Theorem 8 and Theorem 9.

850 **Appendix F** gives further details on the visualisation experiments we conducted to showcase the
851 state-space collapse suffered by Mamba SSMs.

852 **Appendix G** contains technical proofs and constructions deferred from other sections, which are not
853 necessary to fully comprehend the overall argument they are used in.

854 A Additional Preliminaries

855 In this Appendix, we introduce the preliminary notions for the remainder of this work.

856 Section A.1 covers basic mathematical notions and notation used throughout.

857 Section A.2 introduces the necessary background in Metric Spaces and Topology, notably properties
858 of *compactness* and *path-connectedness*.

859 Section A.3 defines the language of Dynamical Systems, which we use to describe RNNs and to build
860 our theory.

861 Section A.4 shows the key Algebraic Automata Theory results and notions which we use in our work.

862 Finally, Section A.5 and Section A.6 cover MLPs and introduce relevant RNN architectures.

863 A.1 Basic Concepts and Notation

864 We introduce basic mathematical concepts and notation required in later sections.

865 A.1.1 Numeric Domains

866 We write $\mathbb{B} = \{0, 1\}$ for the Boolean domain, we write $\mathbb{N} = \{0, 1, \dots\}$ for the natural numbers, we
867 write $\mathbb{N}_{>0} = \{1, 2, \dots\}$ for the natural numbers excluding zero, we write \mathbb{R} for the real numbers, we
868 write \mathbb{R}_+ for the positive real numbers including zero, we write $\mathbb{R}_{>0}$ for the positive real numbers
869 excluding zero, i.e., $\mathbb{R}_{>0} = \mathbb{R}_+ \setminus \{0\}$, and we write $\mathbb{C} = \{\langle a, b \rangle \mid a, b \in \mathbb{R}\}$ for the complex
870 numbers—where every pair $\langle a, b \rangle$ is to be seen as the complex number $a + ib$.

871 For $i, j \in \mathbb{N}$ with $m \leq n$, we define the notation $[i..j] := \{i, i+1, \dots, j\}$.

872 In the rest of the section, let Z be a set.

873 A.1.2 Powersets

874 We write $\mathcal{P}(Z)$ for the *powerset* of Z , and we define $\mathcal{P}_+(Z) := \mathcal{P}(Z) \setminus \{\emptyset\}$.

875 A.1.3 Tuples and Matrices

876 For $n \in \mathbb{N}$, the set of Z -valued n -vectors, or n -tuples over Z , is $Z^n := \{\langle z_1, \dots, z_n \rangle \mid z_i \in Z\}$.
 877 We typically write an element of Z^n as $\mathbf{z} = \langle z_1, \dots, z_n \rangle$. For $m, n \in \mathbb{N}$, the set of Z -valued
 878 $(m \times n)$ -vectors, or $m \times n$ matrices over Z , is $Z^{m \times n} := \{\langle \mathbf{z}_1, \dots, \mathbf{z}_m \rangle \mid \mathbf{z}_i \in Z^n\}$. We typically
 879 write an element of $Z^{m \times n}$ as $\mathbf{Z} = \langle \mathbf{z}_1, \dots, \mathbf{z}_m \rangle$.

880 We use the compact notation $Z_{[i..j]}$ to specify the set $Z_i \times \dots \times Z_j$ resulting from the Cartesian
 881 product of the sets Z_i, \dots, Z_j , meaning that they are contextually introduced by the notation.

882 A.1.4 Sequences

883 A *sequence* over Z with indices $I \subseteq \mathbb{N}$ is a function $s : I \rightarrow Z$, which we commonly present as
 884 $(z_i)_{i \in I}$ where $z_i = s(i)$ for every $i \in I$. A sequence is *finite* if so is its index set, and it is *infinite*
 885 otherwise. When s is an infinite sequence with index set of the form $I = \{m, m+1, \dots\}$, we adopt a
 886 simplified notation and write the sequence as $(z_i)_{i \geq m}$, instead of $(z_i)_{i \in I}$. When s is a finite sequence,
 887 the cardinality of its index set is called the *length* of s . The *empty sequence*, denoted by ε , is the
 888 sequence having length zero, i.e., the sequence with indices $I = \emptyset$. Any finite sequence s with indices
 889 $I = [i..j]$ can be presented as the list z_i, \dots, z_j by letting $z_k = s(k)$ for every $k \in [i..j]$; in this case,
 890 the sequence can also be written in compact form as $z_{[i..j]}$. We write Z^ω for the set of all infinite
 891 sequences on Z , we write Z^* for the set of all finite sequences on Z , we write Z^+ for the set of
 892 all non-empty finite sequences on Z , and we write Z^ℓ for the set of all sequence of a given length
 893 $\ell \in \mathbb{N}$ —noting that this definition of Z^ℓ clearly corresponds to the definition given above of Z^ℓ as
 894 the set of all ℓ -tuples over Z .

895 We often say that a property holds *eventually* for a sequence $(z_i)_{i \geq m}$ if there exists $m' \geq m$ such
 896 that it holds for the sequence $(z_i)_{i \geq m'}$. That is, the property holds for some *tail* of the sequence.

897 A.1.5 Strings

898 A *string* over a finite set Σ is a concatenation (juxtaposition) of elements of Σ . Namely, a string is an
 899 expression $\sigma_1 \sigma_2 \dots \sigma_n$ with $\sigma_i \in \Sigma$, for every $i \in [1..n]$. In this context, we call Σ an *alphabet*, and
 900 we call each element σ_i a *letter* or *symbol* of the string s . We can equivalently see a string $\sigma_1 \sigma_2 \dots \sigma_n$
 901 as the finite sequence $\sigma_{[1..n]}$, following the definition of finite sequence given above, and hence apply
 902 all notions already introduced for finite sequences. In particular, we have that the length of a string
 903 $\sigma_1 \sigma_2 \dots \sigma_n$ is n , that ε is the empty string, that Σ^ℓ is the set of all strings of given length $\ell \in \mathbb{N}$ over
 904 alphabet Σ , that Σ^* is the set of all strings over alphabet Σ , and that Σ^+ is the set of all non-empty
 905 strings over alphabet Σ .

906 A.1.6 Functions and Transformations

907 The image of a function $f : X \rightarrow Y$ is $\text{Im } f := \{f(x) \mid x \in X\} \subseteq Y$. We say that f is an *identity* if
 908 $f(x) = x$ for every $x \in X$, and we say that f is a *permutation* if it is a bijection. A *transformation* of
 909 X is a function $f : X \rightarrow X$ where the codomain coincides with the domain. Note that every identity
 910 transformation is also a permutation, and hence it is sometimes important to distinguish permutations
 911 that are not identities by referring to them as *non-identity permutations*.

912 A.1.7 Equivalence

913 For \sim an equivalence relation on Z , the *equivalence class* of z w.r.t. \sim is the set $[z]_\sim := \{z' \in Z \mid$
 914 $z' \sim z\}$. We denote by Z/\sim the set of equivalence classes of Z w.r.t. \sim .

915 A.2 Metric Spaces and Topology

916 We follow [Willard, 2012] as a general reference for this section, revisiting the notation. Let X be a
917 set fixed for the rest of this section.

918 A.2.1 Metrics

919 A *metric*, or *distance function*, is a function $d : X \times X \rightarrow \mathbb{R}_{>0}$ that satisfies all the following
920 properties for every $x, y, z \in X$:

- a) $d(x, y) = 0 \iff x = y$
- b) $d(x, y) \geq 0$ (positivity)
- c) $d(x, y) = d(y, x)$ (symmetry)
- d) $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality)

921 Notable metrics, relevant to us, are the following ones.

- 922 • The *Euclidean distance*, or L^2 -*norm distance*, is defined as

$$L_X^2(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\| := \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}.$$

- 923 • The *discrete metric* is defined as

$$\mathcal{D}_X(\mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{x} \neq \mathbf{y}, \\ 0 & \text{if } \mathbf{x} = \mathbf{y}. \end{cases}$$

924 We will omit X from a metric when it is clear from the context. For instance, we will write L^2 and \mathcal{D}
925 for L_X^2 and \mathcal{D}_X , respectively.

926 A.2.2 Metric spaces

927 A *metric space* is a tuple $\mathbf{S} = \langle X, d \rangle$ where $d : X \times X \rightarrow \mathbb{R}$ is a metric. Given metric spaces $\mathbf{X} =$
928 $\langle X, d_X \rangle$ and $\mathbf{Y} = \langle Y, d_Y \rangle$, an *isometry* between \mathbf{X} and \mathbf{Y} (or *distance-preserving function*) is a bijective
929 function $f : X \rightarrow Y$ such that, for every $x_1, x_2 \in X$, we have $d_X(x_1, x_2) = d_Y(f(x_1), f(x_2))$.
930 When an isometry exists, the spaces \mathbf{X} and \mathbf{Y} are said to be *isometric*. Intuitively, two isometric
931 spaces are essentially the same metric space. Notable metric spaces, relevant to us, are the following
932 ones, for $n \in \mathbb{N}_{>0}$.

- 933 • The *Euclidean n -space* $\langle \mathbb{R}^n, L^2 \rangle$.
- The *complex n -space* $\langle \mathbb{C}^n, L^2 \rangle$, seen as isometric to $\langle \mathbb{R}^{2n}, L^2 \rangle$, by the following isometry:

$$f(a_1 + ib_1, \dots, a_n + ib_n) = \langle \langle a_1, b_1 \rangle, \dots, \langle a_n, b_n \rangle \rangle.$$

934 In particular, by the isometry above, all our results for Euclidean n -spaces transfer to complex
935 n -spaces seamlessly.

936 We omit the metric when referring to metric spaces, since in the following sections we only consider
937 Euclidean n -spaces $\langle \mathbb{R}^n, L^2 \rangle$ and complex n -spaces $\langle \mathbb{C}^n, L^2 \rangle$, that are always equipped with the L^2
938 as described above. Thus we simply refer to them as \mathbb{R}^n and \mathbb{C}^n , respectively.

939 A *subspace* $\langle Y, d_Y \rangle$ of $\langle X, d_X \rangle$ is a metric space with $Y \subseteq X$ and d_Y given by restriction of d_X to
940 $Y \times Y$.

941 We define the *open ball* $B_X(x, r)$ and *closed ball* $\overline{B}_X(x, r)$ at $x \in X$ of radius $r \geq 0$ in $\langle X, d \rangle$ as
942 the set of points in X with distance $\delta < r$ and $\delta \leq r$ from x , respectively:

$$B_X(x, r) := \{y \in X \mid d(x, y) < r\}, \quad \overline{B}_X(x, r) := \{y \in X \mid d(x, y) \leq r\}.$$

943 A subspace $\langle Y, d_Y \rangle$ of $\langle X, d_X \rangle$ is a metric space with $Y \subseteq X$ and d_Y given by restriction of d_X to
944 $Y \times Y$. We say that a subspace $S \subseteq X$ is *bounded*, if there is some $x \in X$ and $\infty > M \geq 0$ s.t.
945 $S \subseteq B_X(x, M)$. We call a subspace $S \subseteq X$ *open* in X if for all $s \in S$ there is some $\epsilon_s > 0$ s.t.
946 $B_X(s, \epsilon_s) \subseteq S$. S is *closed* in X if $X \setminus S$ is open in X .

947 *Example 1.* The open intervals (a, b) and (a, ∞) are open in \mathbb{R} (with the usual metric). The closed
948 interval $[a, b]$ is closed in \mathbb{R} . The subspace $\{0, 2^{-n} : n \in \mathbb{N}\}$ is closed in \mathbb{R} , while $\{2^{-n} : n \in \mathbb{N}\}$ is
949 neither closed nor open in \mathbb{R} . ■

950 A.2.3 Topology

951 The notion of open subspaces in terms of open balls defines a *topology* on any metric space, which
 952 determines what functions are *continuous*. Formally, a topological space is a tuple (S, \mathcal{T}) , with S
 953 being the underlying set, and $\mathcal{T} \subseteq \mathcal{P}(S)$ being the collection of open sets, such that S and \emptyset , the
 954 union of *any* collection of open sets is open, and the intersection of any *finite* collection of open sets
 955 is open. The open sets definition in terms of open balls for a metric space satisfies these properties.
 956 Many aspects of Metric Automata Theory could be easily restated in the language of Topology
 957 Theory, but we choose a more concrete setting, to make it more accessible.

958 Intuitively, the closed subspaces of X are precisely the ones which contain all their limit points, i.e.
 959 if $(x_n)_{n \geq 1} \subseteq S$ converges to some limit $l \in X$, then $l \in S$.

960 **Fact A.2.1.** For a metric space X , a subset $S \subseteq X$ is closed iff for all sequences $(x_n)_{n \geq 1} \subseteq S$
 961 converging to $l \in X$ we have that $l \in S$. (see §10, Cor. 10.5 of Willard [2012], as every metric space
 962 is first-countable)

963 Note that the notion of openness/closeness is not inherent to the subspace S : it also depends on the
 964 superspace X , since the definition involves balls in X . In fact, any subspace $S \subseteq X$ is by definition
 965 *both open and closed* as a subspace of itself, regardless of whether is open or closed in X . Any time
 966 we use openness or open balls, we need to exercise caution and be clear which space the openness is
 967 referring to.

968 *Example 2.* Consider $M = \mathbb{R}^2$ and $X = \mathbb{R} \times \{0\} = \{(x, 0) \in \mathbb{R}^2 : x \in \mathbb{R}\}$. $(-1, 1) \times \{0\} \subseteq X$ is
 969 an open ball at $(0, 0)$ of radius 2 in X , and thus an open set. However, it is not even an open set in
 970 M ! For any $\epsilon > 0$ we have $\|(0, 0) - (0, \epsilon)\| = \epsilon$, but $(0, \epsilon) \notin S$, and so no open X -ball centred at
 971 $(0, 0)$ is wholly contained in S . ■

972 In fact, any subspace $S \subseteq X$ is by definition *both open and closed* as a subspace of itself, regardless
 973 of whether is open or closed in X .

974 A *continuous function* $f : (M, d) \rightarrow (M', d')$ is the a set function $f : M \rightarrow M'$ such that for
 975 all sequences $(x_n)_{n \geq 1} \subseteq M$ converging to some $x \in M$, the mapped sequence $(f(x_n)) \subseteq M'$
 976 converges to $f(x) \in M'$. The $\epsilon - \delta$ definition of continuity, as well as the topological definition of
 977 continuity ($Y \subseteq M'$ open $\implies f^{-1}(Y) \subseteq M$ open) are equivalent in the metric space setting.

978 *Example 3.* Let S be a subspace of X . Then the inclusion map $\iota : S \rightarrow X$, given by set-theoretical
 979 inclusion $S \subseteq X$, is continuous. ■

980 The topological definition of continuity makes clear the following:

981 **Fact A.2.2.** All functions $f : (M, d) \rightarrow (M', d')$ are continuous for a discrete metric space (M, d) .

982 Next, we introduce two elementary notions in Topology and Metric Space Theory: *compactness* and
 983 *path-connectedness*.

984 A.2.4 Compactness

985 **Definition 9.** A space X is called *compact* if all coverings of X by open subsets of X admit a finite
 986 subcover. For metric spaces, equivalently X is (sequentially) compact, if all sequences in X have a
 987 subsequence converging to a limit in X (see 17G.3 of Willard [2012]). ■

988 The following is a characterization of compact subspaces of \mathbb{R}^d .

989 **Fact A.2.3.** (Heine-Borel) $X \subseteq \Omega$ is a compact subspace iff. X is a bounded, closed subset of \mathbb{R}^d
 990 (see 17.9 of Willard [2012]).

991 *Example 4.* Subspaces $[a, b]$, $\{a\}$, $\{0, 2^{-n} : n \in \mathbb{N}\}$ are compact in \mathbb{R} . (a, b) , $\{2^{-n} : n \in \mathbb{N}\}$ are not
 992 closed, and so they are not compact. \mathbb{R} is not bounded, and so it is not compact. ■

993 Turns out that compactness, unlike openness, is inherent to the subspace, as demonstrated by the
 994 following theorem:

995 **Fact A.2.4.** A continuous image of a compact space is compact (see 17.7 of Willard [2012])

996 Finally, *Tychonoff* Theorem tells us that compactness is a property which is preserved by cartesian
 997 products.

998 **Fact A.2.5.** (Tychonoff) The cartesian product of two compact spaces is compact (see 17.8 of Willard
999 [2012])

1000 A.2.5 Path-connectedness

1001 **Definition 10.** A *path* in X from a to b is a continuous function $\gamma : [0, 1] \rightarrow X$ such that $\gamma(0) = a$
1002 and $\gamma(1) = b$. A space X is called *path-connected* if for all $a, b \in X$ there is a path from a to b . ■

1003 Path-connectedness partitions the space into components, which we will later think of as atomic parts
1004 of the state-space for a dynamical system. - any continuous decoder assigning discrete symbols to the
1005 state-space must be *constant* on a path-connected component, see Lemma 22.

1006 See Section 27D of Willard [2012] for the following:

1007 **Fact A.2.6.** The relation \sim on X given by $a \sim b \iff$ there is a path from a to b in X is an
1008 equivalence. The equivalence classes of \sim are the maximal path-connected subspaces of X .

1009 *Example 5.* Any convex subspace of \mathbb{R}^d is path-connected, in particular open and closed \mathbb{R}^d -balls
1010 are path-connected. $(-1, 0) \cup (0, 1)$ has 2 path-connected components: $(-1, 0)$ and $(0, 1)$. ■

1011 Just like compactness, path-connectedness is an inherent property of the subspace, and is preserved
1012 by Cartesian products (see 27B of Willard [2012]):

1013 **Fact A.2.7.** A continuous image of a path-connected space is path-connected.

1014 **Fact A.2.8.** The cartesian product of two path-connected spaces is path-connected.

1015 A.3 Dynamical Systems

1016 Following Knorozova and Ronca [2024a], we adopt dynamical systems as an general formalism to
1017 describe all systems that operate by maintaining a state recurrently. This allows for treating such
1018 systems in a uniform way despite their differences. In this work specifically, we will use dynamical
1019 systems to formalise Finite Automata and several RNN architectures in Section A.6.

1020 **Definition 11.** A (*dynamical*) *system* is a tuple $S = \langle X, U, f, x_0, Y, h \rangle$, where X is the *state space*,
1021 U is the *input space*, $f : X \times U \rightarrow X$ is the *dynamics function*, $x_0 \in X$ is the *initial state*, Y is the
1022 *output space* and $h : X \times U \rightarrow Y$ is the *output function*. We have that X, U, Y are metric spaces,
1023 and f, h are *continuous*. In our analysis it will be useful to refer to the tuple $D = \langle X, U, f \rangle$ as the
1024 *dynamics* of S , allowing us to focus on just the state transitions.

1025 Given $x_0 \in X$, D defines a map from sequences of inputs $(u_n)_{n \geq 1} \subseteq U$ to sequences of states
1026 $(x_n)_{n \geq 0} \subseteq X$, given by

$$x_{n+1} = f(x_n, u_{n+1}) \quad \text{for } n \geq 0$$

1027 With this, we can define the *state-sequence function* $D : X \times U^* \rightarrow X$ as

$$D(x_0, \varepsilon) = x_0; \quad D(x, u_{1..n}) = x_n$$

1028 S defines a map from sequences of inputs $(u_n)_{n \geq 1} \subseteq U$ to sequences of states $(x_n)_{n \geq 1} \subseteq X$ and
1029 sequences of outputs $(y_n)_{n \geq 1} \subseteq Y$, given by

$$y_n = h(x_n, u_n) = h(D(x_0, u_{1..n}), u_n)$$

1030 Hence we say that S *defines* the function $U^+ \rightarrow Y$, with $S(u_{1..n}) = y_n$. In the special case that h
1031 is independent of U , we may define $S(\epsilon) = h(x_0)$, extending the definition to $S : U^* \rightarrow Y$. ■

1032 **Lemma 10** (State continuity). *Let $S = \langle X, U, f \rangle$ be a dynamics, and for input sequence $(u_n)_{n \geq 1}^N \subseteq$
1033 U and $x_0 \in X$ let $(x_n)_{n \geq 1}^N \subseteq X$ be the sequence of states*

$$x_n = f(x_{n-1}, u_n)$$

1034 *Then x_n is a continuous function of x_0, u_1, \dots, u_n for all $n \in 1..N$. Consequently $y_n = h(x_n, u_n)$
1035 is also a continuous function of x_0, u_1, \dots, u_n , for any continuous h .*

1036 *Proof.* By induction. Writing $x_n(u_1, \dots, u_n)$ we have that

$$x_{n+1} = f(x_n(x_0, u_1, \dots, u_n), u_{n+1})$$

1037 is also a continuous function of x_0, u_1, \dots, u_{n+1} . □

The formalism of cascades provides a flexible way to describe dynamical systems consisting of subsystems forming an acyclic network. Their flexibility will allow us, e.g., to consider not only feed-forward layers of SSMs as in Grazzi et al. [2025], Sarrof et al. [2024], but also more complex architectures with, e.g., blocks in parallel, and mixes of different types of neurons.

Definition 12. A *feed-forward cascade* C is a form of dynamics $\langle X, U, f \rangle$ with $X = X_1 \times \dots \times X_n$, and dynamics function of the form

$$f(\langle x_1, \dots, x_n \rangle, u) = \langle x'_1, \dots, x'_n \rangle$$

$$\text{where } x'_i = f(x_i, \langle u, x'_1, \dots, x'_{i-1} \rangle)$$

We may see C as consisting of dynamics D_1, \dots, D_n where

$$D_i = \langle X_i, U \times X_{[1, i-1]}, f_i \rangle$$

and write $C = D_1 \rightsquigarrow \dots \rightsquigarrow D_n$. ■

Thus, the cascade is evaluated in a feedforward fashion: on input u , first the state of D_1 is updated, then for all subsequent components D_i , the state of D_i is updated based on u and the *updated* states of D_1, \dots, D_{i-1} . This differs from some recurrent neural network literature, where D_i is updated based on u and the *initial* states of D_1, \dots, D_{i-1} , i.e. the update happens at the same time for all components. We refer to such cascades as *serial cascades*.

Definition 13. A *serial cascade* C is a form of dynamics $\langle X, U, f \rangle$ where states are of the form $X = X_1 \times \dots \times X_n$, and the dynamics function is of the form

$$f(\langle x_1, \dots, x_n \rangle, u) = \langle f_1(x_1, u_1), \dots, f_n(x_n, u_n) \rangle, \quad \text{with } u_i = \langle u, x_1, \dots, x_{i-1} \rangle.$$

We may see C as consisting of dynamics D_1, \dots, D_n where

$$D_i = \langle X_i, U \times X_{[1, i-1]}, f_i \rangle$$

and write $C = D_1 \bowtie \dots \bowtie D_n$. ■

Serial cascading can be achieved with feed-forward cascades, and the distinction between the two is irrelevant for our purposes. For details, see Appendix G.2.

In further sections, it will be useful to allow *connection* functions in a cascade, transforming the inputs between components. It will not alter the expressivity results, but it allows us to e.g. define one canonical FLIP-FLOP dynamics, rather than a family of FLIP-FLOP-like dynamics for every possible input and output set.

Definition 14. For dynamics D_1, D_2 with $D_i = \langle X_i, U_i, f_i \rangle$ for all $i \in [1..2]$, and for continuous $i : U \rightarrow U_1$ and $g : U \times X_1 \rightarrow U_2$, we define the *feed-forward cascade with input i and connection g* , written $\overset{i}{\rightsquigarrow} D_1 \overset{g}{\rightsquigarrow} D_2$, and the *serial cascade with input i and connection g* , written $\overset{i}{\bowtie} D_1 \overset{g}{\bowtie} D_2$ as the dynamics $\langle X_1 \times X_2, U, f \rangle, \langle X_1 \times X_2, U, f' \rangle$ respectively, where f and f' are given by

$$f(\langle x_1, x_2 \rangle, u) = \langle x'_1, x'_2 \rangle, \quad \text{where}$$

$$x'_1 = f_1(x_1, i(u))$$

$$x'_2 = f_2(x_2, g(u, x'_1)),$$

and $f'(\langle x_1, x_2 \rangle, u) = \langle f_1(x_1, i(u)), f_2(x_2, g(u, x_1)) \rangle$. Note that for $U_2 = U_1 \times X_2$ and $g = \text{id}$, we recover the usual notion of feed-forward cascade and serial cascade/. ■

For dynamics $D = \langle X, U, f \rangle$ and continuous function $g : Z \rightarrow U$, we define the *dynamics with input function* $D_g = \langle X, Z, (x, z) \mapsto f(x, g(z)) \rangle$. With the notation from the previous definition, note that $D_{1,i} \rightsquigarrow D_{2,g} \equiv \overset{i}{\rightsquigarrow} D_1 \overset{g}{\rightsquigarrow} D_2$, and $D_{1,i} \bowtie D_{2,g} \equiv \overset{i}{\bowtie} D_1 \overset{g}{\bowtie} D_2$. In our expressivity results we will not care about how the dynamics of a neuron interpret the input function, only about the induced transformations of the state-space. Thus, in further sections in proofs we will only consider feed-forward cascading without connection functions, without loss of generality, in order to simplify notation. Further discussion about serial cascades and connecting functions is deferred to Appendix B.5. The next lemma shows the intuitive fact, that it does not matter in which order we "connect" the components of the cascade. In the following propositions, it will be useful to view a cascade $D_1 \rightsquigarrow \dots \rightsquigarrow D_n$ as $(D_1 \rightsquigarrow \dots \rightsquigarrow D_{n-1}) \rightsquigarrow D_n$ for inductive proofs.

1077 **Definition 15.** For dynamics D_1, D_2 , where $D_i = \langle X_i, U_i, f_i \rangle$ for all $i \in [1..2]$, write $D_1 \equiv D_2$ if
 1078 $X_1 = X_2, U_1 = U_2$ and $f_1 = f_2$.

1079 **Lemma 11.** The cascading operation is associative, i.e. we have

$$D_1 \rightsquigarrow (D_2 \rightsquigarrow D_3) \equiv (D_1 \rightsquigarrow D_2) \rightsquigarrow D_3,$$

1080 where ‘ \equiv ’ is as introduced in Definition 15

1081 *Proof.* Say we have $D_i = \langle X_i, U \times X_{[1,i]}, f_i \rangle$ for $i \in 1..3$. Both the LHS and RHS dynamics have
 1082 state space $X_1 \times X_2 \times X_3$ and input space U . Consider a state $\langle x_1, x_2, x_3 \rangle \in X_1 \times X_2 \times X_3$ and
 1083 input $u \in U$.

1084 Write $x'_1 = f_1(x_1, u), x'_2 = f_2(x_2, \langle u, x'_1 \rangle), x'_3 = f_3(x_3, \langle u, x'_1, x'_2 \rangle)$. Also write f_{23} for the
 1085 dynamics function of $D_2 \rightsquigarrow D_3$ and f_{12} for the dynamics function of $D_1 \rightsquigarrow D_2$. Then the state
 1086 update of the LHS system is as follows:

$$\begin{aligned} f_{LHS}(\langle x_1, x_2, x_3 \rangle, u) &= \langle x'_1, f_{23}(\langle x_2, x_3 \rangle, \langle u, x'_1 \rangle) \rangle \\ &= \langle x'_1, \langle x'_2, f_3(x_3, \langle u, x'_1, x'_2 \rangle) \rangle \rangle \\ &= \langle x'_1, x'_2, x'_3 \rangle. \end{aligned}$$

1087 where the second line follows from the definition of cascade dynamics for $D_2 \rightsquigarrow D_3$, and the third
 1088 line follows from associativity of the cartesian product. Analogously,

$$f_{RHS}(\langle x_1, x_2, x_3 \rangle, u) = \langle x'_{12}, f_3(x_3, \langle u, x'_{12} \rangle) \rangle, \quad \text{where } x'_{12} = f_{12}(\langle x_1, x_2 \rangle, u).$$

1089 Now, we have $x'_{12} = f_{12}(\langle x_1, x_2 \rangle, u) = \langle x'_1, f_2(x_2, \langle u, x'_1 \rangle) \rangle = \langle x'_1, x'_2 \rangle$, and so

$$\begin{aligned} f_{RHS}(\langle x_1, x_2, x_3 \rangle, u) &= \langle x'_{12}, f_3(x_3, \langle u, x'_{12} \rangle) \rangle \\ &= \langle \langle x'_1, x'_2 \rangle, f_3(x_3, \langle u, x'_1, x'_2 \rangle) \rangle \\ &= \langle x'_1, x'_2, x'_3 \rangle. \end{aligned}$$

1090 Thus both ways of composing the dynamics D_1, D_2, D_3 results in the same dynamics function. \square

1091 A.4 Algebraic Automata Theory (AAT)

1092 We present an extended version of the background on Algebraic Automata Theory given in the
 1093 preliminaries of the main body.

1094 Algebraic Automata Theory (AAT) allows for studying finite automata through the lens of algebraic
 1095 notions such as semigroups and groups, c.f. [Hartmanis and Stearns, 1966, Ginzburg, 1968, Arbib,
 1096 1969, Dömösi and Nehaniv, 2005]. Its fundamental theorem is the seminal *Prime Decomposition*
 1097 *Theorem* by Krohn and Rhodes [1965], that shows how every semiautomaton can be decomposed
 1098 into a *cascade* of elementary *prime* semiautomata. One prime semiautomaton is the *flip-flop*, that
 1099 describes the elementary system with the ability to store and manipulate one bit of information.

Definition 16. The *flip-flop* is the two-state semiautomaton defined as

$$\text{FLIP-FLOP} := \langle \{\text{high}, \text{low}\}, \{\text{set}, \text{reset}, \text{id}\}, \delta \rangle$$

1100 where

$$\delta(q, \text{id}) = q, \quad \delta(q, \text{set}) = \text{high}, \quad \delta(q, \text{reset}) = \text{low}.$$

1101 AAT often focuses on *state transformations* rather than on the transition function δ of an automaton.
 1102 State transformations are the functions $\delta_\sigma(q) := \delta(q, \sigma)$ obtained by fixing an input σ . They allow us
 1103 to characterise semiautomata in terms of semigroups and groups. In particular, the transitive closure
 1104 of the state transformations of an automaton forms a semigroup, and a monoid or group in special

cases. From this algebraic point of view, the flip-flop is characterised by the *flip-flop* semigroup, which is in fact given by the set of state transformations of FLIP-FLOP. All the other primes are characterised by finite simple groups, and for this reason they are called *group-like*. Specifically, their state transformations form a finite simple group.

Automata whose semiautomaton can be decomposed purely into flip-flops are called *group-free*, and they play a central role in our theory and in general, due to the following theorem whose proof also involves the celebrated theorem by Schützenberger [1965]) on aperiodic semiautomata, cf. [Ginzburg, 1968].

Theorem 12. *The star-free languages is the class of languages recognised by groupfree automata.*

All other automata, that do not admit the above decomposition, are called *non-group-free*, since their prime decompositions always include group-like semiautomata. They admit the following characterisation in terms of state transformations, relevant to our results.

Theorem 13. (Lemma 9 of [Knorozova and Ronca, 2024a]¹) *If a semiautomaton $\langle Q, \Sigma, \delta \rangle$ is not group-free, then there exist $Q' \subseteq Q$ and $\sigma \in \Sigma$ such that the state transformation $\delta_\sigma : Q \rightarrow Q$ is a non-identity permutation on Q' .*

Our theory will extend the applicability of AAT to the study of general dynamical systems. And in particular to analyse the structure of such systems using algebraic means like group theory. A notion from AAT that is key to our results is the notion of *realisation* for Mealy machines (cf. Definitions 1.14 and 1.15 of [Hartmanis and Stearns, 1966]).

Realisation describes how a machine can imitate another machine after a renaming of inputs and outputs—noting that actual names of inputs and outputs are not important in order to characterise what functionalities a machine is fundamentally able to implement.

We recall that a *Mealy machine* is a tuple $\langle Q, \Sigma, \delta, \Gamma, \theta \rangle$ where $\langle Q, \Sigma, \delta \rangle$ is a semiautomaton, Γ is an output alphabet, and $\theta : Q \times \Sigma \rightarrow \Gamma$ is an output function.

A Mealy machine defines the mapping $Q \times \Sigma^+ \rightarrow \Gamma$ given by

$$M(q, w) = \theta(D_M(q, w), w_{-1}),$$

where D_M is the semiautomaton of M .

Given a (finite) automaton $A = \langle Q, \Sigma, \delta, q_0, \Gamma, \theta \rangle$, the *associate Mealy machine* $M_A = \langle Q, \Sigma, \delta, \Gamma, \theta \rangle$ is obtained by dropping the initial state from automaton A .

Given a semiautomaton $D_A = \langle Q, \Sigma, \delta \rangle$ we define its *canonical* Mealy machine as

$$\mathcal{M}(D) := \langle Q, \Sigma, \delta, \Gamma, \theta \rangle, \quad \text{where } \Gamma = Q \times \Sigma, \text{ and } \theta = \text{id}.$$

Definition 17 (Definitions 1.14 and 1.15 of [Hartmanis and Stearns, 1966]). *If $M = \langle Q, \Sigma, \delta, \Gamma, \theta \rangle$ and $M' = \langle Q', \Sigma', \delta', \Gamma', \theta' \rangle$ are Mealy machines, then the triple (α, ι, ζ) is called an *assignment* of M into M' when the functions*

$$\alpha : Q \rightarrow \mathcal{P}_+(Q'), \quad \iota : \Sigma \rightarrow \Sigma', \quad \zeta : \Gamma' \rightarrow \Gamma,$$

satisfy the two conditions below for every $q \in Q$, every $q' \in \alpha(q)$, and every $\sigma \in \Sigma$.

$$\text{I) } \delta'(q', \iota(\sigma)) \in \alpha(\delta(q, \sigma))$$

$$\text{II) } \zeta \circ \theta'(q', \iota(\sigma)) = \theta(q, \sigma)$$

If an assignment of M into M' exists, then M' is said to be a *realisation* of M . ■

The following results tells us how a machine M' that is a realisation of another machine M actually implements its behaviour. Any trajectory through M factors through M' , with ι and ζ acting as the encoder and decoder, respectively, and with α providing an initial state to start from.

Theorem 14. (Theorem 1.5 in §1.3 of [Hartmanis and Stearns, 1966]) *If $M' = \langle Q', \Sigma', \delta', \Gamma', \theta' \rangle$ is a realisation of $M = \langle Q, \Sigma, \delta, \Gamma, \theta \rangle$ through an assignment (α, ι, ζ) , then for all $x_0 \in Q$, $w \in \Sigma^+$, and $x'_0 \in \alpha(x_0)$*

$$\theta(D(x_0, w), w_{-1}) = \zeta \circ \theta'(D'(x'_0, \iota(w)), \iota(w_{-1}))$$

i.e., $M(x_0, w) = \zeta \circ M'(q'_0, \iota(w))$.

¹Lemma 9 of [Knorozova and Ronca, 2024a] can be found in the appendix of its extended version [Knorozova and Ronca, 2023].

We will use the following version of the Krohn-Rhodes decomposition theorem, presented in [Hartmanis and Stearns, 1966], which uses the notion of realisability.

Theorem 15. (Theorem 7.8, §8, Hartmanis and Stearns [1966]) *Let M be a Mealy machine, with group-free semiautomaton. Then M can be realised by a machine with serial cascade dynamics, consisting of FLIP-FLOP components.*

A.5 Multilayer Perceptrons

A Multilayer Perceptron (MLP) is a tuple

$$N = \langle d, \mathbf{n}, U, Y, \alpha, \beta, \mathbf{W}, \mathbf{b} \rangle,$$

where $d \in \mathbb{N}_{>0}$ is called the *depth* or *number of layers*, $\mathbf{n} = \langle n, n_2, n_3, \dots, n_d, m \rangle$ is called *architecture*, $U \subseteq \mathbb{R}^n$ is the input domain, $Y \subseteq \mathbb{R}^m$ is the output domain (or codomain), $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ is called *activation function*, $\beta : \mathbb{R} \rightarrow \mathbb{R}$ is called *activation function of the last layer*, $\mathbf{W} = \langle W_1, \dots, W_d \rangle$ with $W_i \in \mathbb{R}^{n_i \times n_{i+1}}$ called *weight matrices*, and $\mathbf{b} = \langle b_1, \dots, b_d \rangle$ with $b_i \in \mathbb{R}^{n_i}$ called *bias vectors*. Then, N defines the function $f : U \subseteq \mathbb{R}^n \rightarrow Y \subseteq \mathbb{R}^m$ given by the composition $f_1 \circ \dots \circ f_d$ of the functions $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_{i+1}}$ defined as

$$\begin{aligned} f_i(x) &= \alpha(W_i^\top x + b_i) \quad \forall i \in [1..d-1], \\ f_d(x) &= \beta(W_d^\top x + b_d). \end{aligned}$$

We often identify N with the function f , and hence see the network as a function $N : U \rightarrow Y$. The functions f_i are called *layers*, with the first layer f_1 called the *input layer*, the last layer f_d called the *output layer*, and the other layers called *hidden layers*. The (maximum) *width* of N is $\max\{n_2, \dots, n_d\}$. Typical choices for the activation function α are $\text{sigmoid}(x) := \frac{1}{1+\exp(-x)}$ and the *Rectified Linear Unit* $\text{ReLU}(x) := \max\{0, x\}$. The same choices are valid for the last-layer activation function β ; however, as it computes the output of the network, it is often specialised by choosing β to be: the identity function (e.g., for regression tasks), sigmoid (e.g., for binary classification), softmax (e.g., for modelling distributions).

MLPs are universal approximators as long as their activation function α is *non-polynomial*, as established by several well-known Universal Approximation Theorems for feedforward neural networks, cf. [Cybenko, 1992, Hornik et al., 1989].

Theorem 16 (Universal Approximation). *Let α be any non-polynomial activation function. Additionally, let $X \subseteq \mathbb{R}^n$ be compact, and let $f : X \subseteq \mathbb{R}^n \rightarrow Y \subseteq \mathbb{R}^m$ be continuous. For every $\epsilon > 0$, there exists a 2-layer MLP N with activation function α , and identity as its last-layer activation function, such that the following inequality holds:*

$$\sup_{x \in X} \|f(x) - N(x)\| < \epsilon.$$

Note that ReLU and sigmoid are non-polynomial activation functions.

In light of the above theorem, in the rest we will focus on MLPs having non-polynomial activation function α , as well as identity as their last-layer activation function β . This will be relevant in all expressivity results for RNNs whose architecture includes MLPs—as also discussed in Section A.6.

A.6 Recurrent Neural Network Architectures

We present the Recurrent Neural Network (RNN) architectures studied in the following sections.

Classical RNNs are networks of neurons with hidden state $h \in \mathbb{R}^{d_{\text{state}}}$ and update rule of the form

$$h_t = \phi(h_{t-1}, x_t) \quad \text{for } x \in \mathbb{R}^{d_{\text{input}}}$$

where ϕ is commonly a linear transformation composed with a non-linearity, like sigmoid or tanh. We model such neurons as dynamical systems, with hidden state taking values in X , and inputs taking values in U . The hidden state of the neuron at step t may be available to other neurons in the network as part of their input at time $t + 1$.

In modern Machine Learning applications, notably NLP, the networks are in the form of feed-forward connections, with learnable transformations between the neurons. Also some neurons may appear in

parallel, and some neurons might additionally include residual connections. Most generally, we can model such RNNs as acyclic networks, and for nodes N_1, \dots, N_L consider the connection functions $\psi_{i,j}$, describing the transformation which is applied to the value going from neuron N_i to neuron N_j . The network input also may be given to N_i , after going through some transformation ι_i . As the network is acyclic, we may assume that there are no connection functions $\psi_{i,j}$ for $i > j$. Finally, the inputs to N_i are accumulated by some α_i . Now, we may express the network as a feed-forward cascade $D_1 \rightsquigarrow \dots \rightsquigarrow D_L$, with $D_i = \langle X_i, U \times X_{[1..n]}, f_i \rangle$, where X_i is the state-space of neuron N_i , U is the input space of the network, and f_i is given by

$$f_i(x, x_{[1..n]}) = \phi\left(h, \alpha_i(\langle \iota_i(u), \psi_{1,i}(x_1), \dots, \psi_{i-1,i}(x_{i-1}) \rangle)\right)$$

This is how our framework allows to pull the details about the state-less transformations of the input or state-space into the dynamics function.

Classical (Vanilla) RNNs. Vanilla RNNs are networks where the state is updated through a linear combination of the previous state and current input, followed by the application of a non-linear activation function. A prominent example of a vanilla RNN architecture is the *Elman RNN*, which is given by dynamics $D = \langle X, U, f \rangle$ with state space $X \subseteq \mathbb{R}^{\text{state}}$, input space $U \subseteq \mathbb{R}^{\text{input}}$, and dynamics function

$$f(x, u) = \tanh(A_X \cdot x + A_U \cdot u + b),$$

where $A_X \in \mathbb{R}^{\text{state} \times \text{state}}$ is a matrix defining a linear transformation of the state, $A_U \in \mathbb{R}^{\text{state} \times \text{input}}$ is a matrix defining a linear transformation of the input, and $b \in \mathbb{R}^{\text{state}}$ is the bias vector.

State Space Models. *State Space Models (SSMs)* are a family of models based on linear recurrence with particular parametrisation. Notable ones are *Mamba* [Gu and Dao, 2023] and *S4* [Gu et al., 2020].

To model *linear recurrence* in general, we introduce *Linear Recurrent Dynamics*, defined as dynamics $D = \langle X, U, f \rangle$, with state space $X \subseteq \mathbb{K}^{d_{\text{state}}}$, input space $U = \mathbb{K}^{d_{\text{input}}}$, where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$, and with dynamics function

$$f(x, u) = A(u) \cdot x + B(u),$$

where $A(u) \in \mathbb{K}^{d_{\text{state}} \times d_{\text{state}}}$ is the *state-transition gate* and $B(u) \in \mathbb{K}^{d_{\text{state}}}$ is the *input gate*.

SSM architectures often combine linear recurrence blocks with linear projections, non-linearities, residual connections and convolutions. Our theory can easily model such setups with cascade compositions—introduced in Section 2. Consider the Mamba block:

$$\begin{aligned} z_{[1..n]} &= \text{SSM} \circ \sigma \circ \text{Conv} \circ \text{linear}_1(u_{[1..n]}) \\ y_{[1..n]} &= \sigma \circ \text{linear}_2(u_{[1..n]}) \\ o_{[1..n]} &= \text{linear}_3(z_{[1..n]} \times y_{[1..n]}) \end{aligned}$$

where the input sequence $u_{[1..n]} \in U^+$ and output sequence $o_{[1..n]} \in Y^+$ are processed sequentially, each linear_i is a linear projection, σ is a non-linearity, SSM is an SSM block, Conv is a *causal* convolution, and \times is element-wise multiplication. Only Conv and SSM are stateful transformations here. In Figure 2, we present it in the form of a system with cascade dynamics.

We introduce a general class of dynamics as an abstraction for convolution blocks.

Definition 18. *Finite Context Dynamics (FCDs)* with context length ℓ are dynamics $D = \langle X, U, f \rangle$ such that their state depends only on the most recent ℓ inputs. That is, in view of Lemma 10, there is a continuous function $C : U^\ell \rightarrow X$ such that

$$D(x, w) = C(w_{-u}, \dots, w_{-1})$$

for all $x \in X$ and $w \in U^*$ with $|w| \geq \ell$, where w_{-i} is the i -th-to-last element of w .

xLSTM. The recently introduced model xLSTM [Beck et al., 2024] is a successor of the LSTM architecture [Hochreiter and Schmidhuber, 1997], and it achieves performance competitive with transformer architectures. It makes use of both non-linear and linear recurrences. xLSTM introduces two types of blocks: sLSTM and mLSTM. In this work we will focus on the sLSTM block.

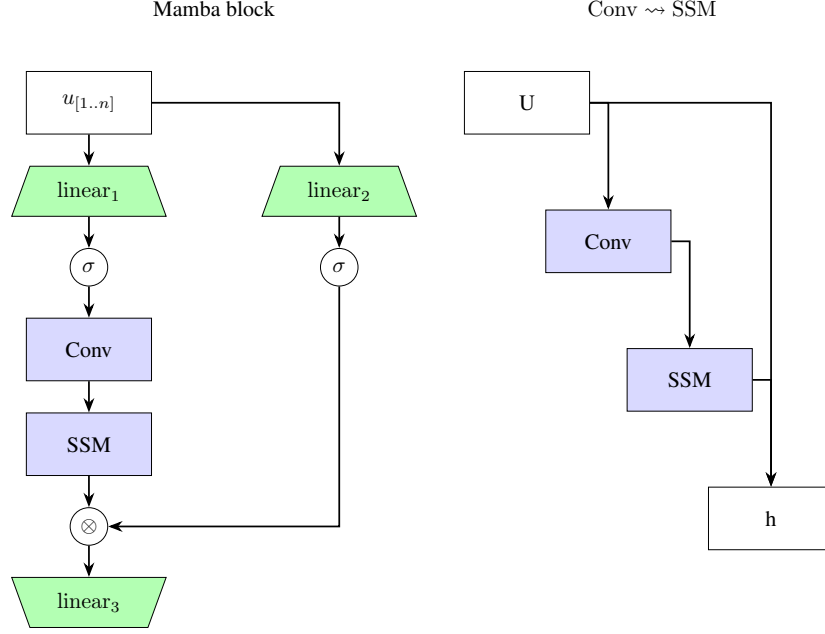


Figure 2: The feedforward cascade structure of a Mamba block. Only Conv and SSM are stateful, so the cascade has 2 components. Structure on the left as it is presented in [Gu and Dao, 2023].

1227 The state space of a sLSTM is \mathbb{R}^3 , and the input space is \mathbb{R}^d for some $d \geq 1$. The dynamics function
 1228 of the form $\langle \langle c, n, h \rangle, u \rangle \mapsto \langle f_c(\langle c, n, h \rangle, u), f_n(\langle c, n, h \rangle, u), f_h(\langle c, n, h \rangle, u) \rangle$, where

$$\begin{aligned} f_c(\langle c, n, h \rangle, u) &= \psi(l_f(h, u)) \cdot c + \exp(l_i(h, u)) \cdot \varphi(l_z(h, u)) \\ f_n(\langle c, n, h \rangle, u) &= \psi(l_f(h, u)) \cdot n + \exp(l_i(h, u)) \\ f_h(\langle c, n, h \rangle, u) &= \sigma(l_o(h, u)) \cdot \frac{f_c(\langle c, n, h \rangle, x)}{f_n(\langle c, n, h \rangle, x)} \end{aligned}$$

1229 where each $l_s : s \in o, i, z, f$ is a function of the form $w_s^t \cdot u + r_s \cdot h + b_s$, for $w_s \in \mathbb{R}^d, r_s, b_s \in \mathbb{R}$,
 1230 ψ is either exp or σ , and φ is tanh.

B Foundations of Metric Automata Theory

In this appendix, we develop the key notions of Metric Automata Theory within the η -finiteness framework.

In Sections B.1 and B.2 we introduce the basic properties of η -finite spaces and dynamics.

In Section B.3 we develop the correspondence between η -finite systems and finite automata, which is crucial to unlocking the powerful theorems of AAT. We provide the proof for Theorem 1.

In Sections B.4 and B.5 we import the notion *realizability* to continuous systems, via the correspondence with automata, and use it to translate algebraic decomposition theorems into the setting of η -finiteness.

B.1 The Notion of η -Finiteness

We begin by introducing η -finiteness, which is a central notion of Metric Automata Theory and our novel finite-precision framework.

Definition 19. Let $X \subseteq \Omega$ for some $d \geq 1$. Call X η -finite if it is a finite union of compact, path-connected sets.

Immediately from the definition we have that an η -finite space is necessarily *compact*—in the case of metric spaces, finite union of bounded, closed sets is bounded and closed. The next result resolves the technicality, that the defining sets in the union of a η -finite X need not be disjoint.

Lemma 17. Let X be η -finite. Then X has finitely many path connected components, say X_1, \dots, X_n , and each of X_i is compact. We shall refer to them as the η -components of X .

Proof. By def, $X = \bigcup_{i=1}^N Y_i$ for some compact and path-connected subsets. By induction on N : If $N = 1$, then the claim is immediate. Now, consider the inductive hypothesis for $N \geq 2$, that $X' = \bigcup_{i=1}^{N-1} Y_i$ has finitely many path connected components X_1, \dots, X_n , each compact. The path connected components of X are then unions of elements from $\{X_1, \dots, X_n, Y_N\}$. Each of these sets is compact, and so each such finite union is compact: clearly it is still bounded, and a finite union of closed sets is still closed. \square

Example 6. Any finite alphabet is η -finite, with each symbol in a separate η -component. The subspace $[-2, 1] \cup \{2\} \subseteq \mathbb{R}$ is η -finite. The subspace $(-2, 1) \cup \{2\}$ is not η -finite, since it is not compact. The subspace $\{0, 2^{-n} : n \in \mathbb{N}\}$ is compact but not η -finite, since it is not a finite union of path-connected sets. \blacksquare

Both compactness and path-connectedness are preserved by continuous mappings and (finite) Cartesian products, see Facts A.2.4, A.2.5, A.2.7, and A.2.8. This gives us the corresponding results for η -finite spaces.

Lemma 18. Continuous image of an η -finite space is η -finite.

Proof. Write $X = \bigcup_{i=1}^N X_i$ for path-connected, compact sets X_i . Let $f : X \rightarrow Y$ be continuous. We have:

$$f(X) = \bigcup_{i=1}^N f(X_i)$$

By Facts A.2.4 and A.2.7, each $f(X_i)$ is compact and path-connected. Thus by definition $f(X)$ is η -finite. \square

Lemma 19. The Cartesian product $X \times Y$ space of η -finite spaces is η -finite. The η -components of $X \times Y$ are the products of η -components of X and η -components of Y .

Proof. Let X_1, \dots, X_n and Y_1, \dots, Y_m be the C-components of X, Y respectively. We have $X = \bigcup_{i=1}^n X_i$, $Y = \bigcup_{j=1}^m Y_j$ and so

$$X \times Y = \left(\bigcup_{i=1}^n X_i \right) \times \left(\bigcup_{j=1}^m Y_j \right) = \bigcup_{i=1}^n \bigcup_{j=1}^m X_i \times Y_j$$

1272 By Facts A.2.8 and A.2.5 each $X_i \times Y_j$ is path-connected. Therefore by def. $X \times Y$ is η -finite.
 1273 Moreover, the η -components of $X \times Y$ are unions of the products $X_i \times Y_j$. Now, fix $i \in [1..n]$, $j \in$
 1274 $[1..j]$. Let Z be the η -component of $X \times Y$ containing $X_i \times Y_j$. consider the projection map
 1275 $\pi_X : X \times Y \rightarrow X$. As the projection is continuous, the image, $\pi_X(Z)$ is path-connected in X by
 1276 Fact A.2.7. Moreover, $X_i \in \pi_X(Z)$. Thus, as X_i is a maximal path-connected subspace of X , we
 1277 have $X_i = \pi_X(Z)$. Similarly, considering the projection $\pi_Y : X \times Y \rightarrow Y$, we have $Y_j = \pi_Y(Z)$.
 1278 Since $X_i \times Y_j \subseteq Z$, we therefore must have $X_i \times Y_j = Z$. Therefore $X \times Y$ has finitely many
 1279 η -components, and they are the products of η -components of X and η -components of Y . \square

1280 **Lemma 20.** *Let X be η -finite, with η -component X_1, \dots, X_n . For some $\delta > 0$ we have*

$$\inf_{x \in X_i, y \in X_j} \|x - y\| \geq \delta \quad \text{for all } i \neq j.$$

1281 *Proof.* It is sufficient to show this in the case that X has two η -components, say X_1, X_2 . Define
 1282 $f : X_1 \times X_2 \rightarrow \mathbb{R}_{\geq 0}$ by $f(x_1, x_2) = \|x_1 - x_2\|$. This is continuous, and so $\text{Im } f$ is compact, as
 1283 $X_1 \times X_2$ is compact. Since X_1, X_2 are disjoint, $0 \notin \text{Im } f$. Thus 0 is not a limit point of $\text{Im } f$, and
 1284 so for some $\delta > 0$ we have that $[0, \delta) \not\subseteq \text{Im } f$. \square

1285 **Corollary 21.** *Let $X \subseteq \Omega$ be η -finite and $(x_n)_{n \geq 1} \subseteq X$ converge in Ω . Then $(x_n)_{n \geq 1}$ is eventually*
 1286 *contained in a single η -component of X .*

1287 **Lemma 22.** *Let X be an η -finite space and Σ a finite alphabet. Then a function $f : X \rightarrow \Sigma$ is*
 1288 *continuous if and only if it is constant on the η -components of X*

1289 *Proof.* (\Leftarrow) Suppose $f : X \rightarrow \Sigma$ is constant on η -components of X . Let $(x_n)_{n \geq 1} \subseteq X$ converge to
 1290 $x \in X$. Then by Lemma 20, $(x_n)_{n \geq 1}$ is eventually contained in the same η -component as x . Thus
 1291 $f(x_n) = f(x)$ eventually, in particular $f(x_n) \rightarrow f(x)$ as $n \rightarrow \infty$. Hence f is continuous.

1292 (\Rightarrow) If f is continuous, then it maps η -component of X to path-connected subspaces of Σ . Therefore
 1293 f must be constant on η -components. \square

1294 B.2 Dynamical Systems and η -Finiteness

1295 **Definition 20.** We say that dynamics $\langle X, U, f \rangle$ are η -finite if both X and U are η -finite. A system S
 1296 is η -finite if its dynamics are η -finite.

1297 *Example 7.* Take $X = [-1, -1/2] \cup [1/2, 1]$ and $U = \{-1, 0, 1\}$. The both X and U are η -finite.
 1298 Define $f : X \times U \rightarrow X$ by:

$$f(x, u) = \begin{cases} x & \text{if } u = 0 \\ u & \text{if } u = 1, -1 \end{cases}$$

1299 Thus under input $u = 0$ the dynamics function performs the identity transformation on X , and under
 1300 inputs $u = 1, -1$, X is mapped to $1, -1$ respectively. The dynamics $D = \langle X, U, f \rangle$ is η -finite. \blacksquare

1301 Note, that by Lemma 19, a cascade of η -finite components is itself η -finite.

1302 **Lemma 23.** *Let $D = \langle X, U, f \rangle$ be a η -finite dynamics, and $h : X \times U \rightarrow Y$ be continuous. Then*
 1303 *the image of h , $\text{Im } h \subseteq Y$, is η -finite.*

1304 *Proof.* Immediately follows from Lemma 18. \square

1305 **Lemma 24** (Path-connected \Rightarrow same state). *Let $D = \langle X, U, f \rangle$ be a dynamics, and consider*
 1306 *$x_0, x'_0 \in X$, and input sequences $(u_n)_{n \geq 1}, (u'_n)_{n \geq 1} \subseteq U$, and the corresponding state sequences*
 1307 *$(x_n)_{n \geq 1}, (x'_n)_{n \geq 1} \subseteq X$. Suppose that for all $n \geq 1$, $u_n \sim_U u'_n$, and $x_0 \sim_X x'_0$. Then for all $n \geq 1$*
 1308 *we have that $x_n \sim_X x'_n$, i.e.,*

$$D(x_0, u_{[1..n]}) \sim_X D(x'_0, u'_{[1..n]})$$

1309 *Proof.* Let $n \geq 1$. By 10, we have that there is for each n a continuous function $x_n(x_0, u_1, \dots, u_n)$
 1310 determining the n -th state. Now, since each pair u_i, u'_i for $i \in 1..n$ is path-connected in U , we have

1311 that $\langle u_{1..n} \rangle$ and $\langle u'_{1..n} \rangle$ are path-connected in U^n - the path connecting them applies the corresponding
 1312 1-d paths pointwise. Thus by continuity of x_n ,

$$x_n = x_n(x_0, \langle u_{1..n} \rangle), x'_n = x_n(x_0, \langle u'_{1..n} \rangle)$$

1313 are path-connected in X . □

1314 **Corollary 25.** Let $S = \langle X, U, f, x_0, Y, h \rangle$ be a η -finite system, and let us consider input sequences
 1315 $(u_n)_{n \geq 1}, (u'_n)_{n \geq 1} \subseteq U$ such that for all n u_n and u'_n are in the same path-connected component.
 1316 Then the corresponding state sequences $(x_n)_{n \geq 1}, (x'_n)_{n \geq 1} \subseteq X$, and the corresponding output
 1317 sequences $(y_n)_{n \geq 1}, (y'_n)_{n \geq 1} \subseteq Y$ are such that for all n x_n and x'_n are in the same path-connected
 1318 component of X and y_n and y'_n are in the same path-connected component of $\text{Im } h$

1319 In light of the above results, we introduce the notion of *equivalent* sequences, for convenience in later
 1320 proofs.

1321 **Definition 21.** Let X be a η -finite space. Call sequences $(x_n)_{n \geq 1}, (x'_n)_{n \geq 1} \subseteq X$ *equivalent*, if for
 1322 each n we have that x_n and x'_n are in the same component of X . Call these sequences *eventually*
 1323 *equivalent*, if they have equivalent tail sequences.

1324 Overall, the notions of η -finiteness and η -component have *very favourable theoretical properties*.
 1325 Any continuous mapping $f : X \rightarrow Y$, with X and Y η -finite, is guaranteed to map every element of
 1326 an η -component of X into a single η -component of Y .

1327 In the case of η -finite systems, this means that the dynamics function acts on the η -components
 1328 of the state-space (referred to as η -states) in the same way for each input within an η -component
 1329 of the input-space (referred to as η -input). Moreover, every point within an η -component of the
 1330 output function image (which is always η -finite), must be decoded as the same alphabet symbol. We
 1331 formalize these properties in the following section.

1332 B.3 Representing η -Finite Systems as Automata and Proof of Theorem 1

1333 For set A and equivalence \sim on A , write A/\sim for the set of its equivalence classes. For $a \in A$ write
 1334 $[a]_A$ for the \sim -equivalence class containing a .

1335 For η -finite spaces A , we will write \bar{A} for the set A/\sim_A , with \sim_A being the path-connectedness
 1336 equivalence. For X, Y being η -finite spaces, we have by Lemma 19 that $\bar{X} \times \bar{Y} = \overline{X \times Y}$.

1337 **Definition 22.** Any η -finite dynamical system $S = \langle X, U, f, x_0, Y, h \rangle$ defines its *canonical automa-*
 1338 *ton*

$$A_S = \langle \bar{X}, \bar{U}, \bar{f}, [x_0]_{\sim_X}, \overline{\text{Im } h}, \bar{h} \rangle$$

1339 Similarly, any η -finite dynamics $D = \langle X, U, f \rangle$ defines its *canonical semiautomaton* $D_A =$
 1340 $\langle \bar{X}, \bar{U}, \bar{f} \rangle$. ■

1341 Note that by Lemma 23, $\text{Im } h$ is indeed η -finite. $\bar{f} : (\bar{X}) \times (\bar{U}) \rightarrow (\bar{X})$ is defined as $[x]_{\sim_X}, [u]_{\sim_U} \mapsto$
 1342 $[f(x, u)]_{\sim_X}$. $\bar{h} : \bar{X} \times \bar{U} \rightarrow \overline{\text{Im } h}$ is defined as $[x]_{\sim_X}, [u]_{\sim_U} \mapsto [h(x, u)]_{\sim_{\text{Im } h}}$. This is well defined
 1343 by Lemma 25.

1344 For a η -finite dynamical system $S = \langle X, U, f, x_0, Y, h \rangle$, define the *canonical regular function*
 1345 $F_S : (\bar{U})^+ \rightarrow \overline{\text{Im } h}$ to be the function defined by the FSA A_S . The following lemma shows that the
 1346 dynamics of the canonical automaton determine—up to path-connectedness—the dynamics of the
 1347 system.

1348 **Lemma 26.** Let $D = \langle X, U, f \rangle$ be a η -finite dynamics, and D_A be its canonical semiautomaton.
 1349 Then

$$D_A([x_0]_{\sim_X}, [w]_{\sim_U}) = [D(x_0, w)]_{\sim_X} \quad \forall w \in U^* \quad (1)$$

1350 where $[w]_{\sim_U} \in U^*$ denotes the word with each letter of w replaced by its equivalence class.

1351 *Proof.* By induction on the length of w . For the base case $w = \varepsilon$, we have $D_A([x_0]_{\sim_X}, [\varepsilon]_{\sim_U}) =$
 1352 $D_A([x_0]_{\sim_X}, \varepsilon) = [x_0]_{\sim_X}$ and by definition $D(x_0, \varepsilon) = x_0$, so that $[D(x_0, \varepsilon)]_{\sim_X} = [x_0]_{\sim_X}$.

1353 Now, suppose for $w \in U^*$ we have $D_A([x_0]_{\sim_X}, [w]_{\sim_U}) = [D(x_0, w)]_{\sim_X}$, and let $[u]_{\sim_U} \in \overline{U}$.

1354 Write $w[u]_{\sim_U}$ for the word obtained by appending $[u]_{\sim_U}$ at the end of w , we have

$$\begin{aligned} D_A([x_0]_{\sim_X}, [wu]_{\sim_U}) &= \overline{f}(D_A([x_0]_{\sim_X}, [w]_{\sim_U}), [u]_{\sim_U}) \\ &= \overline{f}([D(x_0, w)]_{\sim_X}, [u]_{\sim_U}) \\ \text{by def. of } \overline{f} &= [f(D(x_0, w), u)]_{\sim_X} \\ &= [D(x_0, wu)]_{\sim_X} \end{aligned}$$

1355 Thus by induction the statement holds for all $w \in U^*$. □

1356 **Lemma 27.** Let S be a η -finite system and F_S be its canonical regular function. Then, F_S is
 1357 implemented by S with encoder $\overline{\text{enc}} : \overline{U} \rightarrow U$ given by $[u]_{\sim_U} \mapsto u'$ with $u' \in [u]_{\sim_U}$ chosen
 1358 arbitrarily, and with decoder $\overline{\text{dec}} : \text{Im } h \rightarrow \overline{\text{Im } h}$, given by $y \mapsto [y]_{\sim_{\text{Im } h}}$.

1359 *Proof.* $\overline{\text{enc}}$ is continuous, since \overline{U} is a finite alphabet. $\overline{\text{dec}}$ is continuous by Lemma 22. Let D_A be
 1360 the dynamics of A_S , and let D_S be the dynamics of S . Then we have

$$F_S(w) = \overline{h}(D_A([x_0]_{\sim_X}, w), w_{-1}) \quad \forall w \in \overline{U}^+$$

1361 where w_{-1} denotes the last symbol in word w . Now consider $w \in (\overline{U}/\sim_U)^+$ and write $[u]_{\sim_U}$ for
 1362 w_{-1} . By Lemma 26, we have $D_A([x_0]_{\sim_X}, w) = [D_S(x_0, \overline{\text{enc}}(w))]_{\sim_X}$, so that

$$\begin{aligned} \overline{h}(D_A([x_0]_{\sim_X}, w), w_{-1}) &= \overline{h}\left([D_S(x_0, \overline{\text{enc}}(w))]_{\sim_X}, [u]_{\sim_U}\right) \\ \text{as } u' = \overline{\text{enc}}([u]_{\sim_U}) \in [u]_{\sim_U} &= \overline{h}\left([D_S(x_0, \overline{\text{enc}}(w))]_{\sim_X}, [u']_{\sim_U}\right) \\ \text{by def. of } \overline{h} &= \left[h\left(D_S(x_0, \overline{\text{enc}}(w)), u'\right)\right]_{\text{Im } h} \\ &= \left[h\left(D_S(x_0, \overline{\text{enc}}(w)), \overline{\text{enc}}(w_{-1})\right)\right]_{\text{Im } h} \\ &= \left[S(\overline{\text{enc}}(w))\right]_{\text{Im } h} = \overline{\text{dec}} \circ S(\overline{\text{enc}}(w)) \end{aligned}$$

1363 This concludes the proof. □

1364 **Lemma 28.** Let η -finite system $S = \langle X, U, f, x_0, Y, h \rangle$ implement function $F : \Sigma^+ \rightarrow \Gamma$ with
 1365 encoder $\text{enc} : \Sigma \rightarrow U$ and decoder $\text{dec} : \text{Im } h \rightarrow \Gamma$. Then there are (continuous) functions
 1366 $\text{enc}' : \Sigma \rightarrow \overline{U}$ and $\text{dec}' : \overline{\text{Im } h} \rightarrow \Gamma$ such that

$$F(w) = \text{dec}' \circ F_S(\text{enc}'(w)) \quad \forall w \in \Sigma^+$$

1367 where $F_S : (\overline{U})^+ \rightarrow (\overline{\text{Im } h})$ is the canonical function for S .

1368 *Proof.* Define enc' as $\sigma \mapsto [\text{enc}(\sigma)]_{\sim_U}$ for all $\sigma \in \Sigma$.

1369 As for dec' , define it as $[y]_{\sim_{\text{Im } h}} \mapsto \text{dec}(y)$. This is well-defined: Consider $y_1, y_2 \in \text{Im } h$ such that
 1370 $y_1, y_2 \in [y]_{\sim_{\text{Im } h}}$. Since y_1, y_2 are path-connected in $\text{Im } h$, by continuity of $\text{dec} : \text{Im } h \rightarrow \Gamma$ we have
 1371 that $h(y_1), h(y_2)$ are path-connected in Γ . Therefore necessarily $h(y_1) = h(y_2)$.

1372 Let A_S be the canonical FSA of S . Denote the dynamics of S as D_S and the dynamics of A_S as D_A .
 1373 By Lemma 26, we have

$$D_A([x_0]_{\sim_X}, \text{enc}'(w)) = [D_S(x_0, \text{enc}(w))]_{\sim_X} \quad \forall w \in \Sigma^+$$

1374 Thus we have for all $w \in \Sigma^+$

$$\begin{aligned} \text{dec}' \circ F_S(\text{enc}'(w)) &= \text{dec}' \circ \bar{h}(D_A([x_0]_{\sim_X}, \text{enc}'(w)), \text{enc}'(w_{-1})) \\ &= \text{dec}' \circ \bar{h}([D_S(x_0, \text{enc}(w))]_{\sim_X}, [\text{enc}(w_{-1})]_{\sim_U}) \\ &= \text{dec}' \left[h(D_S(x_0, \text{enc}(w)), \text{enc}(w_{-1})) \right]_{\sim_{\text{Im } h}} \\ &= \text{dec} \circ S(\text{enc}(w)) \end{aligned}$$

1375 Finally, enc' and dec' are continuous, since their domains are finite alphabets. \square

1376 **Theorem 1.** An η -finite system S can implement the same functions as its canonical automaton,
 1377 which are necessarily regular.

1378 *Proof.* Suppose $S = \langle X, U, f, x_0, Y, h \rangle$ implements a function $F : \Sigma \rightarrow \Gamma$, with encoder $\text{enc} : \Sigma \rightarrow U$
 1379 and decoder $\text{dec} : Y \rightarrow \Gamma$. By Lemma 28, we have that the canonical FSA of S , say
 1380 $A_S = \langle \bar{X}, \bar{U}, \bar{f}, [x_0]_{\sim_X}, \bar{\text{Im } h}, \bar{h} \rangle$, implements F with encoder enc' and decoder dec' .

1381 Moreover, consider the FSA $A' = \langle \bar{X}, \Sigma, \delta, [x_0]_{\sim_X}, \Gamma, \theta \rangle$, where $\delta : \bar{X} \times \Sigma \rightarrow \bar{X}$ is given by

$$\delta([x]_{\sim_X}, \sigma) = \bar{f}([x]_{\sim_X}, \text{enc}'(\sigma))$$

1382 and $\theta : \bar{X} \times \Sigma \rightarrow \Gamma$ is given by

$$\theta([x]_{\sim_X}, \sigma) = \text{dec}' \circ \bar{h}([x]_{\sim_X}, \text{enc}'(\sigma))$$

1383 Then we have that $F(w) = A'(w)$ for all $w \in \Sigma^+$. Thus F is necessarily regular.

1384 Now, suppose that A_S implements a function $F : \Sigma \rightarrow \Gamma$, with encoder $\text{enc} : \Sigma \rightarrow U$ and decoder
 1385 $\text{dec} : \text{Im } h \rightarrow \Gamma$. By Lemma 27, S implements F_S with encoder $\overline{\text{enc}}$ and decoder $\overline{\text{dec}}$. Thus we have
 1386 the following: for all $w \in \Sigma^+$

$$\begin{aligned} F(w) &= \text{dec} \circ A_S(\text{enc}(w)) \\ &= \text{dec} \circ F_S(\text{enc}(w)) \\ &= \text{dec} \circ \overline{\text{dec}} \circ (\overline{\text{enc}} \circ \text{enc}(w)) \end{aligned}$$

1387 so that S implements F with encoder $\overline{\text{enc}} \circ \text{enc}$ and decoder $\text{dec} \circ \overline{\text{dec}}$. \square

1388 B.4 Algebraic Theory of η -Finite Systems

1389 The connection between η -finite systems and canonical automata is extremely useful. It gives us a
 1390 way to employ the powerful characterisations and results of AAT to any η -finite system dynamics.
 1391 Namely, we can extend the notion of *realisability* to continuous η -finite systems, via the canonical
 1392 automaton.

1393 **Definition 23.** We say that η -finite dynamics D' are a realisation of η -finite dynamics D when
 1394 $\mathcal{M}(\mathcal{C}(D'))$ is a realisation of $\mathcal{M}(\mathcal{C}(D))$ of D .

1395 We say that automaton A' is a realisation of system A , if the associated machine $M_{A'}$ is a realisation of
 1396 of the associated machine M_A via an assignment (α, ι, ζ) , and the respective initial states x'_0, x_0 are
 1397 such that $x'_0 \in \alpha(x_0)$.

1398 Say that η -finite system S' is a realisation of system S , if $A_{S'}$ is a realisation of A_S , where $A_S, A_{S'}$
 1399 are the canonical automata. \blacksquare

1400 The notion of realisation for machines is transitive. See §1.3 of Hartmanis and Stearns [1966].

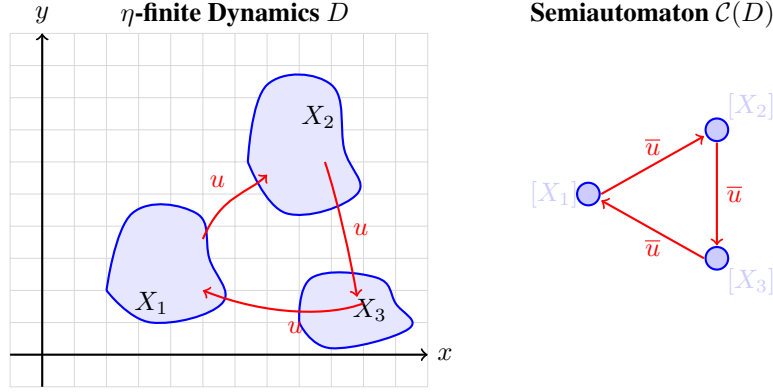


Figure 3: System dynamics and corresponding canonical semiautomaton.

Fact B.4.1. If M is a realisation of M' and M' is a realisation of M'' , then M realises M'' .

It is easy to see that the notion of realisation for dynamics and systems is also transitive.

Lemma 29. Suppose that semiautomaton D' is a realisation of semiautomaton D . Then

1) for any machine M with dynamics D , the canonical machine $\mathcal{M}(D')$ of D' is a realisation of M ,

2) for any automaton A with dynamics D , an initial state can be picked for $\mathcal{M}(D')$ such that the resulting automaton is a realisation of A .

Proof. Say $D = \langle Q, \Sigma, \delta \rangle$ and $D' = \langle Q', \Sigma', \delta' \rangle$. Suppose we have an assignment (α, ι, ζ) from D to D' . That is, $\alpha : Q \rightarrow \mathcal{P}_+(Q')$, $\iota : \Sigma \rightarrow \Sigma'$, $\zeta : Q' \times \Sigma' \rightarrow Q \times \Sigma$

Let $M = \langle Q, \Sigma, \delta, \Gamma, \theta \rangle$ be a Mealy machine with semiautomaton D . The canonical machine for D' is

$$\mathcal{M}(D') = \langle Q', \Sigma', \delta', \Gamma' = Q' \times \Sigma', \theta' = \text{id} \rangle$$

Define $\zeta' : (Q' \times \Sigma') \rightarrow \Gamma$ by $\zeta' = \theta \circ \zeta$. Want to show: (α, ι, ζ') give an assignment of M into $\mathcal{M}(D')$. We already have that the condition I) is satisfied.

Now, for any $q \in Q, \sigma \in \Sigma$ and $q' \in \alpha(q)$ we have that $\zeta \circ \theta'(q', \iota(\sigma)) = \text{id}(q, \sigma) = (q, \sigma)$, since (α, ι, ζ) give an assignment of D into D' . Thus

$$\zeta' \circ \theta'(q', \iota(\sigma)) = \theta \circ \zeta \circ \theta'(q, \sigma) = \theta(q, \sigma)$$

So (α, ι, ζ') also satisfy condition II). Thus the 1) part of the statement holds.

Now for the part 2): Let $A = \langle Q, \Sigma, \delta, q_0, \Gamma, \theta \rangle$ be a system with dynamics D . By part 1), the associated machine $M_A = \langle Q, \Sigma, \delta, q_0, \Gamma, \theta \rangle$ has some assignment (α, ι, ζ) into $M_{D'}$. $\alpha(x_0)$ is a non-empty set, and so we may arbitrarily pick $x'_0 \in \alpha(x_0)$. Then the automaton $A' = \langle Q', \Sigma', \delta', q'_0, Q' \times \Sigma', \text{id} \rangle$ obtained from setting initial state x'_0 for machine $M_{D'}$, by definition is a realisation of A . \square

We have the following proposition to connect our notion of dynamical systems with Algebraic Automata Theory.

Before proceeding, we remark that Definition 1 must be made fully precise by saying that a decoder is a function $\text{dec} : \text{Im } h \rightarrow \Gamma$ where h is the output function of system S , (rather than a function $\text{dec} : Y \rightarrow \Gamma$).

Theorem 30. Let S and S' be η -finite systems, and $A_S, A_{S'}$ their respective canonical automata. If $A_{S'}$ is a realisation of A_S , then S' can implement all the functions that S can implement.

Proof. Say we have $A_S = \langle \overline{X}, \overline{U}, \overline{f}, \overline{x_0}, \overline{\text{Im } h}, \overline{h} \rangle$ and $A_{S'} = \langle \overline{X'}, \overline{U'}, \overline{f'}, \overline{x'_0}, \overline{\text{Im } h'}, \overline{h'} \rangle$.

Say that an assignment of A_S into $A_{S'}$ is given by $\alpha : \overline{X} \rightarrow \mathcal{P}_+(\overline{X'}), \iota : \overline{U} \rightarrow \overline{U'}$ and $\zeta : \overline{\text{Im } h'} \rightarrow \overline{\text{Im } h}$. Let $F_S : (\overline{U})^+ \rightarrow \overline{\text{Im } h}$ be the canonical regular function for S . By Lemma 28, it suffices to show that $A_{S'}$ can implement F_S .

1431 Define the encoder $\text{enc} : \overline{U} \rightarrow \overline{U'}$ as $\text{enc} = \iota$ and decoder $\text{dec} : \overline{\text{Im } h'} \rightarrow \overline{\text{Im } h}$ as $\text{dec} = \zeta$. Let
 1432 D, D' be the dynamics of $A_S, A_{S'}$ resp. By Theorem 1.4 in §1.3 of [Hartmanis and Stearns, 1966],
 1433 we have for all $x' \in \alpha(\overline{x_0})$ and all $w \in (\overline{U})^+$, that

$$h(D(\overline{x_0}, w), w_{-1}) = \zeta \circ h'(D'(x', \iota(w)), \iota(w_{-1})).$$

1434 Thus, for all $w \in (\overline{\Sigma})^+$ we have

$$\begin{aligned} F_S(w) &= A_S(w) = \overline{h}(D(x_0, w), w_{-1}) \\ &= \zeta \circ h'(D'(x'_0, \iota(w)), \iota(w_{-1})) \\ &= \text{dec} \circ S'(\text{enc}(w)). \end{aligned}$$

1435 This concludes the proof. \square

1436 *Example 8.* The reverse implication to Theorem 30 does not hold in general. Consider $\Sigma = \Sigma' = \{\sigma\}$,
 1437 $Q = \{a, b\}$, $Q' = \{a\}$ and unary dynamics functions $\delta : Q \times \Sigma \rightarrow Q$ defined as $\delta(q, \sigma) = q$ for
 1438 every $q \in Q$, and depicted next.



1439

1440 And similarly $\delta' : Q' \times \Sigma \rightarrow Q'$ defined as $\delta'(q, \sigma) = q$ for every $q \in Q'$, and depicted next.



1441

1442 Define system $S = \langle Q, \Sigma, \delta, x_0 = a, \Gamma = Q, \theta \rangle$ with $\theta : (q, \sigma) \mapsto q$, and system $S' =$
 1443 $\langle Q', \Sigma, \delta', q'_0 = a, \Gamma' = Q', \theta' \rangle$ with $\theta' : (q, \sigma) \mapsto q$.

1444 The only possible state trajectories for either systems are the constant trajectories $x_n = x_0 = a$ and
 1445 $x'_n = x'_0 = a$. Thus, a function $\Sigma^+ \rightarrow \Gamma$ can be represented by either system if and only if it is
 1446 constant. So we have that both systems implement the same functions.

1447 However, there is no assignment (α, ι, ζ) from S to S' . This is because Γ' is a singleton, and so any
 1448 potential $\zeta : \Gamma' \rightarrow \Gamma$ must be constant. At the same time, it must hold that $\alpha(a), \alpha(b)$ are non-empty
 1449 and

$$\begin{aligned} \zeta \circ \theta(q', \iota(\sigma)) &= \theta(a, \sigma) = a \quad \forall q' \in \alpha(a), \\ \zeta \circ \theta(q', \iota(\sigma)) &= \theta(b, \sigma) = b \quad \forall q' \in \alpha(b). \end{aligned}$$

1450 This is a contradiction, as ζ must be constant. \blacksquare

1451 **Theorem 31.** Let D, D' be η -finite dynamics. Suppose that D' is a realisation of D . Then any
 1452 function implemented by a system with dynamics D can be implemented by some system with dynamics
 1453 D' .

1454 *Proof.* Let $D_A = \langle \overline{X}, \overline{U}, \overline{f} \rangle$ and $D_{A'} = \langle \overline{X'}, \overline{U'}, \overline{f'} \rangle$ be the canonical semiautomata of D and D' ,
 1455 respectively. Then $D_{A'}$ realises D_A .

1456 Let S be a system with dynamics D implementing function F . Its canonical automaton A_S has
 1457 dynamics D_A , and so by Lemma 29 there is an automaton $A' = \langle \overline{X'}, \overline{U'}, \overline{f'}, \overline{x'_0}, \Gamma', \theta' \rangle$ with dynamics
 1458 $D_{A'}$ which realises A_S .

1459 Consider the system $S' = \langle X', U', f', x'_0, X' \times U', \text{id} \rangle$, where $x'_0 \in X'$ is s.t. $[x'_0]_{\sim_{X'}} = \overline{x'_0}$. Its
 1460 canonical automaton is $A_{S'} = \langle \overline{X'}, \overline{U'}, \overline{f'}, \overline{x'_0}, \overline{X'} \times \overline{U'}, \text{id} \rangle$. $A_{S'}$ realises A' with the assignment
 1461 $\alpha : \overline{X'} \rightarrow \mathcal{P}_+(\overline{X'})$ g.b. $\overline{x'} \mapsto \{\overline{x'}\}$, $\iota : \overline{U'} \rightarrow \overline{U'}$ g.b. $\overline{u'} \mapsto \overline{u'}$ and finally $\zeta : \overline{X'} \times \overline{U'} \rightarrow \Gamma'$ g.b.
 1462 $(\overline{x'}, \overline{u'}) \mapsto \theta'(\overline{x'}, \overline{u'})$. Thus by Theorem 30, S' can implement all functions that S can implement. \square

1463 B.5 Cascade Decomposition and η -Finite Systems

1464 In this section we bridge the gap between the AAT decomposition results, which apply to serial
1465 cascading, and our η -finite framework, which focuses on feed-forward connections. We begin by
1466 showing how taking the canonical semiautomaton ‘commutes’ with feed-forward cascading.

1467 **Lemma 32.** *Let $D_1 \rightsquigarrow \dots \rightsquigarrow D_n$ be η -finite feed-forward cascade dynamics. Then we have*

$$\mathcal{C}(D_1 \rightsquigarrow \dots \rightsquigarrow D_n) \equiv \mathcal{C}(D_1) \rightsquigarrow \dots \rightsquigarrow \mathcal{C}(D_n),$$

1468 where ‘ \equiv ’ is as per Definition 15.

1469 *Proof.* By induction, it suffices to show the statement for $n = 2$.

1470 We have $D_1 = \langle X_1, U_1, f_1 \rangle$ and $D_2 = \langle X_2, U_1 \times X_1, f_2 \rangle$. Now, $\mathcal{C}(D_1) = \langle \overline{X}_1, \overline{U}_1, \overline{f}_1 \rangle$ and
1471 $\mathcal{C}(D_2) = \langle \overline{X}_2, \overline{U}_1 \times \overline{X}_1, \overline{f}_2 \rangle$. Note, that here we use that, by Lemma 19, $(\overline{U}_1 \times \overline{X}_1) = \overline{U}_1 \times \overline{X}_1$.

1472 Thus, we may write the cascade

$$\mathcal{C}(D_1) \rightsquigarrow \mathcal{C}(D_2) = \langle \overline{X}_1 \times \overline{X}_2, \overline{U}_1, \overline{f} \rangle$$

1473 where f is the dynamics function of the feed-forward cascade $\mathcal{C}(D_1) \rightsquigarrow \mathcal{C}(D_2)$.

1474 At the same time, writing $D_1 \times D_2 = \langle X_1 \times X_2, U_1, f' \rangle$, we have

$$\mathcal{C}(D_1 \rightsquigarrow D_2) = \langle \overline{X}_1 \times \overline{X}_2, \overline{U}_1, \overline{f'} \rangle,$$

1475 where again we use Lemma 19 to get $(\overline{X}_1 \times \overline{X}_2) = \overline{X}_1 \times \overline{X}_2$. It remains to show that $f = \overline{f'}$. For
1476 $[x_1]_{\sim_{X_1}} \in \overline{X}_1, [x_2]_{\sim_{X_2}}, [u]_{\sim_{U_1}}$ we have

$$\begin{aligned} f(\langle [x_1]_{\sim_{X_1}}, [x_2]_{\sim_{X_2}}, [u]_{\sim_{U_1}} \rangle) &= \langle \overline{f}_1([x_1]_{\sim_{X_1}}, [u]_{\sim_{U_1}}), \\ &\quad \overline{f}_2([x_2]_{\sim_{X_2}}, \langle [u]_{\sim_{U_1}}, \overline{f}_1([x_1]_{\sim_{X_1}}, [u]_{\sim_{U_1}}) \rangle) \rangle \\ &= \langle [f_1(x_1, u_1)]_{\sim_{X_1}}, [f_2(x_2, \langle u, f_1(x_1, u) \rangle)]_{\sim_{X_2}} \rangle \\ &= \langle [f_1(x_1, u_1), f_2(x_2, \langle u, f_1(x_1, u) \rangle)]_{\sim_{X_1 \times X_2}} \rangle \\ &= [f'(\langle x_1, x_2 \rangle, u)]_{\sim_{X_1 \times X_2}} = \overline{f'}(\langle [x_1, x_2]_{\sim_{X_1 \times X_2}}, [u]_{\sim_{U_1}} \rangle) \\ &= \overline{f'}(\langle [x_1]_{\sim_{X_1}}, [x_2]_{\sim_{X_2}}, [u]_{\sim_{U_1}} \rangle) \end{aligned}$$

1477 This concludes the proof. □

1478 Note: we treat objects such as $\overline{X}_1 \times \overline{X}_2$ and $(\overline{X}_1 \times \overline{X}_2)$ as identical, even though one is a product of
1479 equivalence classes, and the other is an equivalence class of a product. However, from Lemma 19,
1480 we can identify the two in a natural way, that is in a way that is consistent with applying functions
1481 component-wise.

1482 Next, we show that cascading interacts well with realisability, up to introducing a connection function.

1483

1484 **Lemma 33.** *Suppose $D_i = \langle X_i, U_i, f_i \rangle, D'_i = \langle X'_i, U'_i, f'_i \rangle$ are such that D'_i is a realisation for D_i ,
1485 for each $i \in [1..2]$. Then, for any feed-forward cascade $\rightsquigarrow D_1 \xrightarrow{h} D_2$ with input i and connection h ,
1486 there is a continuous function $g : U'_1 \times X'_1 \rightarrow U'_2$ such that $D'_1 \xrightarrow{g} D'_2$ realises $\rightsquigarrow D_1 \xrightarrow{h} D_2$.*

1487 *Proof.* Let $(\alpha_i, \iota_i, \zeta_i)$ be the assignment of $\mathcal{M}(\mathcal{C}(D_i)) = \langle \overline{X}_i, \overline{U}_i, \overline{f}_i, \overline{X}_i \times \overline{U}_i, \text{id} \rangle$ into
1488 $\mathcal{M}(\mathcal{C}(D'_i)) = \langle \overline{X}'_i, \overline{U}'_i, \overline{f}'_i, \overline{X}'_i \times \overline{U}'_i, \text{id} \rangle$, for each $i \in [1..2]$. We assume w.l.o.g. that $h = \text{id}$
1489 and $i = \text{id}$, i.e., we can consider the usual feed-forward cascade $D_1 \rightsquigarrow D_2$, by replacing D_1 with
1490 $D_{1,i}$ and D_2 with $D_{2,h}$. In that case, we have $U_2 = U_1 \times X_1$.

1491 Define $g : \overline{U}'_1 \times \overline{X}'_1 \rightarrow \overline{U}'_2$ given by $g(\overline{u}', \overline{x}') = \iota_2(\overline{u}, \overline{x}) \in U'_2$ where $(\overline{x}, \overline{u}) = \zeta_1(\overline{x}', \overline{u}') \in$
1492 $X_1 \times U_1 = U_2$.

1493 Define

$$\begin{aligned}
\alpha : (\overline{X}_1 \times \overline{X}_2) &\rightarrow \mathcal{P}_+(\overline{X}'_1 \times \overline{X}'_2) & \text{as } \alpha(\overline{x}_1, \overline{x}_2) &= \alpha_1(\overline{x}_1) \times \alpha_2(\overline{x}_2) \\
\iota : \overline{U}_1 &\rightarrow \overline{U}'_1 & \text{as } \iota &= \iota_1 \\
\zeta : \overline{X}'_1 \times \overline{X}'_2 \times \overline{U}'_1 &\rightarrow \overline{X}_1 \times \overline{X}_2 \times \overline{U}_1 & \text{as } \zeta(\langle \overline{x}'_1, \overline{x}'_2 \rangle, \overline{u}'_1) &= (a, b, c) \\
&& \text{where } (b, c, a) &= \zeta_2(x'_2, g(\overline{u}'_1, \overline{x}'_1))
\end{aligned}$$

1494 Let $(\overline{x}_1, \overline{x}_2) \in \overline{X}_1 \times \overline{X}_2$, $\overline{u}_1 \in \overline{U}_1$ and $(\overline{x}'_1, \overline{x}'_2) \in \alpha((\overline{x}_1, \overline{x}_2))$. Let \overline{f} and \overline{f}'_g be the
1495 dynamics functions of $\mathcal{C}(D_1) \rightsquigarrow \mathcal{C}(D_2)$ and $\mathcal{C}(D'_1) \rightsquigarrow \mathcal{C}(D'_2)_g$ respectively. We have that
1496 $\overline{f}'_g(\langle \overline{x}'_1, \overline{x}'_2 \rangle, \iota(\overline{u}_1)) = \langle \overline{x}'_{1,\text{new}}, \overline{x}'_{2,\text{new}} \rangle$, where

$$\overline{x}'_{1,\text{new}} = \overline{f}'_1(\overline{x}'_1, \iota_1(\overline{u}_1)) \in \alpha_1(\overline{f}_1(\overline{x}_1, \overline{u}_1))$$

1497 by Property I) of assignment, and

$$\overline{x}'_{2,\text{new}} = \overline{f}'_2(x'_2, g(\iota(\overline{u}), \overline{x}'_{1,\text{new}}))$$

1498 Now, by Property II) of assignment we have $\zeta_1(\overline{x}'_{1,\text{new}}, \iota(u)) = (f_1(\overline{x}_1, \overline{u}_1), \overline{u}_1)$, since $\overline{x}_{1,\text{new}} \in$
1499 $\alpha_1(\overline{f}_1(\overline{x}_1, \overline{u}_1))$. Thus

$$\overline{x}'_{2,\text{new}} = \overline{f}'_2(x'_2, \iota_2(\overline{u}_1, \overline{f}_1(\overline{x}_1, \overline{u}_1))) \in \alpha_2(\overline{f}_2(\overline{x}_2, \langle \overline{u}_1, \overline{f}_1(\overline{x}_1, \overline{u}_1) \rangle))$$

1500 So, altogether $\langle \overline{x}'_{1,\text{new}}, \overline{x}'_{2,\text{new}} \rangle \in \alpha(\overline{f}(\langle \overline{x}_1, \overline{x}_2 \rangle, \overline{u}_1))$, so Property I) of assignment is satisfied.

1501 Now

$$\zeta(\langle \overline{x}'_{1,\text{new}}, \overline{x}'_{2,\text{new}} \rangle, \iota(\overline{u}_1)) = (a, b, c)$$

1502 where $(b, c, a) = \zeta_2(\overline{x}'_{2,\text{new}}, g(\iota(\overline{u}_1), \overline{x}'_{1,\text{new}})) = \zeta_2(\overline{x}'_{2,\text{new}}, \iota_2(\overline{u}_1, \overline{f}_1(\overline{x}_1, \overline{u}_1)))$. Thus

$$\zeta(\langle \overline{x}'_{1,\text{new}}, \overline{x}'_{2,\text{new}} \rangle, \iota(\overline{u}_1)) = \overline{f}(\langle \overline{x}_1, \overline{x}_2 \rangle, \overline{u}_1)$$

1503 and so Property II) is satisfied. We may now choose a continuous $g' : U'_1 \times X'_1 \rightarrow U'_2$ such that
1504 $\overline{g}' = g$ by Lemma 22. Then we have that $\mathcal{C}(D'_{2,g'}) = \mathcal{C}(D'_2)_g$. Overall, the cascade $D'_1 \rightsquigarrow D'_{2,g'}$
1505 realises $D_1 \rightsquigarrow D_2$. \square

1506 The decomposition theorems of AAT are stated for serial cascades, while RNNs in practice usually
1507 work with feed-forward cascades. In Appendix G.2, we show how $D_1 \bowtie D_2$ can be realised by
1508 $D_1 \xrightarrow{g_1} R_X \xrightarrow{g_2} D_2$ for some continuous functions g_1, g_2 , and the repeat dynamics R_X over state-space
1509 X of D_1 .

1510 **Definition 24.** The repeat dynamics on state space X are the dynamics $R_X = \langle X^2, X, r_X \rangle$, where
1511 $r_X(\langle x_{\text{old}}, x_{\text{new}} \rangle, x) = \langle x_{\text{new}}, x \rangle$. \blacksquare

1512 Thus we have that with initial state $\langle a, b \rangle \in X^2$ and input sequence $(u_n)_{n \geq 1} \in X^\omega$, the state
1513 sequence is $(s_n = \langle x_{n-1}, x_n \rangle)_{n \geq 0} \in (X^2)^\omega$ with $x_{-1} = a, x_0 = b$. Note that a repeat dynamics is
1514 a Finite Context Dynamics.

1515 For η -finite spaces, R_X can be decomposed in terms of 2-state repeat dynamics.

1516 **Theorem 34.** Let X be an η -finite space. Then the repeat dynamics on X , R_X , are realised by a
1517 feed-forward cascade of the repeat dynamics R_2 on $\{0, 1\}$.

1518 *Proof.* Let X_1, \dots, X_n be the η -components of X . We can think of the canonical automaton as the
1519 repeat dynamics on \overline{X} , $R_{\overline{X}} = \{\overline{X}^2, \overline{X}, r_{\overline{X}}\}$.

1520 Consider $C_n = \xrightarrow{f_1} D_1 \xrightarrow{f_2} D_2 \dots \xrightarrow{f_n} D_n = \langle \{0, 1\}^{2 \times n}, \overline{X}, f_C \rangle$, with $D_i \equiv R_2$ for all $i \in [1..n]$,
 1521 and with $f_i : \overline{X} \times \{0, 1\}^{2 \times i-1} \rightarrow \{0, 1\}$ given by

$$f_i(\overline{x}_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

1522 Thus, each D_i works in parallel, treating inputs \overline{x}_i as 1, and others as 0. Then we can retrieve the
 1523 state of $R_{\overline{X}}$ by checking which D_i has 1 at the old position, and which D_j has 1 at new position.
 1524 This corresponds to state $\langle \overline{x}_i, \overline{x}_j \rangle$.

1525 The assignment this corresponds to is the following: define $\alpha : \overline{X}^2 \rightarrow \mathcal{P}_+(\{0, 1\}^{2n})$ by
 1526 $\alpha(\langle \overline{x}_i, \overline{x}_j \rangle) = \{E_{i,j}\}$, where $E_{i,j} \in \{0, 1\}^{2 \times n}$ is s.t. $[E_{i,j}]_{1,i} = 1, [E_{i,j}]_{2,j} = 1$ and remain-
 1527 ing entries are all 0. We also define $\iota : \overline{X} \rightarrow \overline{X}$ as the identity, and $\zeta : \{0, 1\}^{2 \times n} \times \overline{X} \rightarrow \overline{X}^2 \times X$
 1528 as mapping $(E_{i,j}, \overline{x}) \mapsto (\langle x_i, x_j \rangle, \overline{x})$, with other inputs mapped arbitrarily.

1529 □

1530 Altogether, we have a recipe for proving positive results. It is sufficient to show that an architecture
 1531 can realise FLIP-FLOP, to show that it can implement all group-free functions with *serial* cascades. If
 1532 it further can realize R_2 , then it can implement all group-free functions with *feed-forward* cascades.

1533 **Theorem 35.** *Suppose that η -finite dynamics D is a realisation of FLIP-FLOP, and η -finite dynamics*
 1534 *E a realization of R_2 . Then feed-forward cascades of D and E components can implement all group*
 1535 *free functions.*

1536 *Proof.* Let F be a group-free function. By Theorem 12, F is implemented by a serial cascade
 1537 of FLIP-FLOP's, say C . By the construction in Appendix G.2, we have that C is realised by
 1538 a feed-forward cascade of FLIP-FLOP's and repeat semiautomata, say C' . By Lemma 34, each
 1539 repeat semiautomaton is a feed-forward cascade of R_2 components. Therefore C' is realised by
 1540 a feed-forward cascade of FLIP-FLOP's components and R_2 components, say C'' . By Lemma 33,
 1541 a feed-forward cascade of D and E components realises C'' , say C''' . Thus, by transitivity of
 1542 realisability, C''' realises C , and thus by Theorem 31, C''' can implement F . □

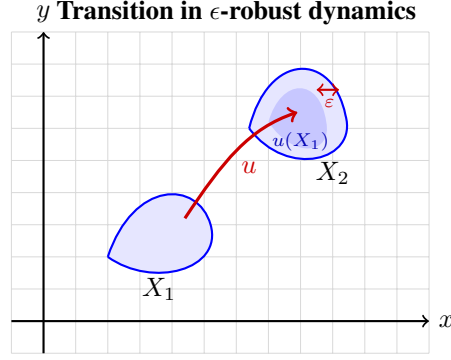


Figure 4: The image of an η -component under an ϵ -robust transition lies inside the target η -component, within ϵ -distance of its boundary.

C Robust Systems

In this appendix we introduce a central notion of *robustness* that allows us to extend Metric Automata Theory to the study of concrete finite-precision implementations.

Arithmetic operations with floating point numbers are difficult to analyse, since addition and multiplication are not exactly commutative, associative and distributive. Thus, for example, the recurrent form and the convolutional form of the SSM update are not exactly equivalent (also noted by Merrill et al. [2024]—see footnote 3 in Definition 2.1). A theoretical framework which specifies an explicit datatype either is hard to analyse, or introduces additional simplifying assumptions.

The central notion that allows us to extend Metric Automata Theory to the study of finite-precision implementations is the notion of ϵ -robustness. Intuitively, it describes stability of the dynamics under transition perturbations.

In Section C.1 we prove Theorem 2, thus showing that robustness provides a way to connect η -finite systems to their floating-point implementations on real-world computer architectures, without requiring us to commit to any particular standard of floating-point operations.

In Section C.2 we show that robustness provides stability under perturbing the parameters of a model which describes the dynamics. We will later present a strongly robust dynamics based on the sLSTM model, which uses a particular choice of parameters. Our results show, that in such cases the parameters may be perturbed by some amount and the robust system will retain its behaviour.

Lastly, in Section C.3 we prove Theorem 5 and further describe what kind of connecting functions are required for strongly robust η -finite cascades, by showing that 2-layer MLPs suffice.

Robustness marks the departure of Metric Automata Theory from Classical Automata and Formal Languages Theory, allowing us to study phenomena that do not occur with discrete state-spaces.

For completeness, we restate Definition 2 paying closer attention to the role of inputs in the notion of strong ϵ -robustness.

Definition 2. For $\epsilon > 0$ and $X \subseteq \Omega_X, U \subseteq \Omega_U$, dynamics $D = \langle X, U, f \rangle$ are ϵ -robust (in Ω_X) if, for every $x \in X$ and every $u \in U$, it holds that $\overline{B}_{\Omega_X}(f(x, u), \epsilon) \subseteq X$ —i.e., $y \in X$ for all $y \in \Omega_X$ s.t. $\|f(x, u) - y\| \leq \epsilon$. Furthermore, we say that dynamics D are strongly ϵ -robust (in Ω_X and Ω_U) if they are ϵ -robust (in Ω_X), each η -component of X contains an Ω_X -ball of radius at least ϵ and each η -component of U contains an Ω_U -ball of radius at least ϵ .

Note that the property of robustness is with respect to the ambient space Ω_X , which contains the state space X . Thus, it is possible that a dynamics is ϵ -robust w.r.t. some ambient space (e.g., \mathbb{R}), and not ϵ -robust w.r.t. another ambient space (e.g., \mathbb{C}). This captures the property, that for a η -finite dynamics, a function approximating f within ϵ , and taking values in Ω , will implement the same transitions.

Lemma 36. Let $C = D_1 \rightsquigarrow \dots \rightsquigarrow D_n$ be a cascade, with $D_i = \langle X_i, U \times X_{[1, i-1]}, f_i \rangle$ and $X_i \subseteq \Omega_i, U \subseteq \Omega_U$. Then C is (strongly) ϵ -robust w.r.t. $\Omega_1 \times \dots \times \Omega_n$ if D_i is (strongly) ϵ -robust w.r.t. Ω_i for all $i \in 1..n$.

1579 *Proof.* By induction, it suffices to show the statement for $n = 2$. First, suppose that D_i is ϵ -robust
 1580 for $i \in 1, 2$. Let $\langle x_1, x_2 \rangle \in X_1 \times X_2$, $u \in U$ and take $\langle y_1, y_2 \rangle \in \Omega_1 \times \Omega_2$ s.t. $\|f(\langle x_1, x_2 \rangle, u) -$
 1581 $\langle y_1, y_2 \rangle\|_2 \leq \epsilon$. We have, by def of cascading

$$f(\langle x_1, x_2 \rangle, u) = \langle x'_1, x'_2 \rangle \quad \text{where } x'_1 = f_1(x_1, u), x'_2 = f_2(x_2, \langle x'_1, u \rangle)$$

1582 By definition of the L_2 norm, since $\|\langle x'_1, x'_2 \rangle - \langle y_1, y_2 \rangle\| \leq \epsilon$, we also have

$$\|x'_1 - y_1\| \leq \epsilon \quad \text{and} \quad \|x'_2 - y_2\| \leq \epsilon$$

1583 Thus, by ϵ -robustness, we have that $y_i \in X_i$ for $i \in 1, 2$, and hence $\langle y_1, y_2 \rangle \in X_1 \times X_2$. All together,
 1584 C is ϵ -robust w.r.t. $\Omega_1 \times \Omega_2$.

1585 Suppose further that D_1, D_2 are strongly ϵ -robust. Let Z be a η -component of $X_1 \times X_2$. Then Z
 1586 is of the form $Z_1 \times Z_2$ for Z_i η -component of X_i , see proof of Lemma 19. We have by strongly-
 1587 robustness that $\overline{B}_{\Omega_i}(z_i, \epsilon) \subseteq Z_i$ for some $z_i \in Z_i$. By triangle inequality: $\overline{B}_{\Omega_1 \times \Omega_2}((z_1, z_2), \epsilon) \subseteq$
 1588 $\overline{B}_{\Omega_1}(z_1, \epsilon) \times \overline{B}_{\Omega_2}(z_2, \epsilon) \subseteq Z_1 \times Z_2$. Finally, the input space of $D_1 \rightsquigarrow D_2$ is the same as the
 1589 input space of D_1 , so by strongly-robustness we have that each η -component of U contains a closed
 1590 Ω_U -ball with radius ϵ . \square

1591 C.1 Finite Datatypes and Proof of Theorem 2

1592 We now consider approximations of dynamical systems using a finite datatype $\mathbb{D} \subseteq \Omega$. \mathbb{D} can for
 1593 example represent the Python float type. We simply consider \mathbb{D} as a discrete subset of Ω , abstracting
 1594 away the details regarding arithmetic properties of such a datatype.

1595 **Definition 25.** A *finite datatype* is a set $\mathbb{D} \subseteq \Omega = \mathbb{R}^d$ having finite cardinality. A *finite-datatype*
 1596 *implementation* of a system S is then a system whose input, state, and output spaces are finite
 1597 datatypes, and whose dynamics and output functions are implemented using floating-point operations.

1598 **Definition 26.** Call a set S an ϵ -covering of $X \subseteq \Omega$, if for all $x \in X$ there is a $s \in S$ s.t. $\|x - s\| \leq \epsilon$.

1599 **Definition 27.** Define $\text{int}_p^+ = \{0, \dots, 2^{p-1} - 1\}$ to be the p -bit unsigned integers. Define $\text{int}_p =$
 1600 $\{2^{p-1}, \dots, 0, \dots, 2^{p-1} - 1\}$ to be the p -bit signed integers. Define \mathbb{D}_p to be floating point numbers
 1601 with $2p$ -bit significand and p -bit exponent:

$$\mathbb{D}_p = \left\{ \frac{s}{2^{2p-1}} \cdot 2^e : s \in \text{int}_{2p}, e \in \text{int}_p \right\}$$

1602 Similarly, define \mathbb{D}'_p to be floating point numbers with p bits of integer precision and p bits of
 1603 fractional precision:

$$\mathbb{D}'_p = \left\{ a + \frac{b}{2^p} : a \in \text{int}_p, b \in \text{int}_p^+ \right\}$$

1604 **Lemma 37.** Let $X \subseteq \Omega = \mathbb{R}^d$ be compact. Then, for p sufficiently large, i.e. with sufficient precision,
 1605 \mathbb{D}_p^d is an ϵ -covering of X .

1606 *Proof.* X is a compact subspace of Ω , and therefore bounded. So, there is some integer $k \geq 1$ s.t.
 1607 $X \subseteq [-2^k, 2^k - 1]^d$. There is also some integer $l \geq 1$ s.t. $\epsilon/\sqrt{d} \geq 2^{-l}$. Take $p \geq \max(k, l)$. The
 1608 set \mathbb{D}'_p is an 2^{-p} -cover of $[2^{-p}, 2^p - 1]$. Now for any $x \in X \subseteq [-2^p, 2^p - 1]^d$, we have that for
 1609 each $i \in 1 \dots d$ there is $y_i \in \mathbb{D}'_p$ s.t. $|[x]_i - y_i| \leq 2^{-p}$. Therefore, writing $y \in [-2^p, 2^p - 1]^d$ for
 1610 (y_1, \dots, y_d)

$$\|x - y\| = \left(\sum_{i=1}^d |[x]_i - [y]_i|^2 \right)^{1/2} \leq \epsilon$$

1611 Therefore $(\mathbb{D}'_p)^d$ an ϵ -cover of X . \square

1612 **Lemma 38.** Let $X \subseteq \Omega = \mathbb{R}^d$ be compact. Then, for p sufficiently large, i.e. with sufficient precision,
 1613 \mathbb{D}_p^d is an ϵ -covering of X .

1614 *Proof.* By the previous Lemma, for some p we have that \mathbb{D}'_p is an ϵ -cover of X . We have for each
 1615 $a \in \text{int}_p, b \in \text{int}_p^+$:

$$a + \frac{b}{2^p} = \frac{2^p \cdot a + b}{2^{2p+1}} \cdot 2^{p+1}$$

1616 Now, $2^p a + b \geq 2^p(-2^p) - 2^p > -2^{2p+1}$ and $2^p a + b \leq 2^p \cdot 2^p + 2^p < 2^{2p+1}$, so that $2^p a + b \in$
 1617 $\text{int}_{2^{p+2}}$. Since $p+1 < 2^{p+1}$, we have $p+1 \in \text{int}_{p+1}$. So, $\mathbb{D}'_p \subseteq \mathbb{D}_{p+1}$, and therefore \mathbb{D}_{p+1} is also
 1618 an ϵ -cover for X , for sufficiently large p . \square

Definition 28. Let X, U be η -finite spaces having components $X_{[1..r]}, U_{[1..s]}$ and subspaces $X' \subseteq X$, $U' \subseteq U$, respectively. Let us consider dynamics

$$D = \langle X, U, f \rangle \quad \text{and} \quad \hat{D} = \langle X', U', \hat{f} \rangle.$$

1619 We say that dynamics D are *simulated* by dynamics \hat{D} , with error at most ϵ , if we have that the
 1620 disjointness condition (C1) holds for every $i \in [1..r]$, the disjointness condition (C2) holds for every
 1621 $j \in [1..s]$, and the approximation condition (C3) holds.

$$(C1) \quad X' \cap X_i \neq \emptyset, \quad (C2) \quad U' \cap U_j \neq \emptyset, \quad (C3) \quad \sup_{x \in X', u \in U'} \|f(x, u) - \hat{f}(x, u)\| \leq \epsilon.$$

1622 **Lemma 39.** Suppose η -finite dynamics $D = \langle X, U, f \rangle$ are ϵ -robust, and are simulated by η -finite
 1623 dynamics $\hat{D} = \langle X', U', \hat{f} \rangle$ with error ϵ . Then \hat{D} is a realisation of D .

1624 *Proof.* Consider the canonical semiautomata $D_A = \langle \overline{X}, \overline{U}, \overline{f} \rangle$ and $\hat{D}_A = \langle \overline{X}', \overline{U}', \overline{f}' \rangle$

1625 Define $\alpha : \overline{X} \rightarrow \mathcal{P}_+(\overline{X}')$ as

$$\alpha([x]_{\sim_X}) = \{[x']_{\sim_{X'}} \in \overline{X}' : x' \in [x]_{\sim_X}\}$$

1626 which is indeed non-empty by definition of simulation, and well-defined as $X' \subseteq X$, and so if
 1627 $x'_1 \sim_{X'} x'_2$ then also $x'_1 \sim_X x'_2$. Also define $\iota : \overline{U} \rightarrow \overline{U}'$ by

$$\iota([u]_{\sim_U}) = [u']_{\sim_{U'}} \quad \text{where } u' \in U' \cap [u]_{\sim_U} \text{ is arbitrary,}$$

1628 and $\zeta : (\overline{X}' \times \overline{U}') \rightarrow (\overline{X} \times \overline{U})$ by

$$\zeta([x']_{\sim_{X'}}, [u']_{\sim_{U'}}) = ([x']_{\sim_X}, [u']_{\sim_U})$$

1629 ζ is indeed well-defined: suppose $x'_1, x'_2 \in [x']_{\sim_{X'}}$ and $u'_1, u'_2 \in [u']_{\sim_{U'}}$. Then since $X' \subseteq X$ and
 1630 $U' \subseteq U$ we also have $x'_1, x'_2 \in [x']_{\sim_X}$, since $x'_1 \sim_{X'} x'_2$ and $u'_1, u'_2 \in [u']_{\sim_U}$, since $u'_1 \sim_{U'} u'_2$.

1631 Now, (α, ι, ζ) is an assignment of $\mathcal{M}(D_A)$ into $\mathcal{M}(\hat{D}_A)$: for all $[x]_{\sim_X} \in \overline{X}$ and $[u]_{\sim_U} \in \overline{U}$, and for
 1632 all $[x']_{\sim_{X'}} \in \alpha([x]_{\sim_X})$ we have

$$\overline{f}'([x']_{\sim_{X'}}, \iota([u]_{\sim_U})) = [f'(x', u')]_{\sim_{X'}}, \quad \text{where } [u']_{\sim_{U'}} = \iota([u]_{\sim_U})$$

1633 On the other hand, we have

$$\alpha(\overline{f}([x]_{\sim_X}, [u]_{\sim_U})) = \alpha([f(x, u)]_{\sim_X})$$

1634 We have that $x' \in [x]_{\sim_X}$, since $[x'] \in \alpha([x]_{\sim_X})$. We have by simulation with error at most ϵ

$$\|f'(x', u') - f(x, u)\| \leq \epsilon$$

1635 and so $f'(x', u') \in [f(x, u)]_{\sim_X}$, since D is ϵ -robust. Hence $\overline{f}'([x']_{\sim_{X'}}, \iota([u]_{\sim_U})) \in$
 1636 $\alpha(\overline{f}([x]_{\sim_X}, [u]_{\sim_U}))$. Thus Part I) of definition of assignment is satisfied.

1637 Moreover, we have

$$\zeta([x']_{\sim_{X'}}, \iota([u]_{\sim_U})) = ([x']_{\sim_X}, [u']_{\sim_U}) = ([x]_{\sim_X}, [u]_{\sim_U})$$

1638 so that Part II) of the definition is satisfied. \square

Strongly robust system + ϵ -covering approximation

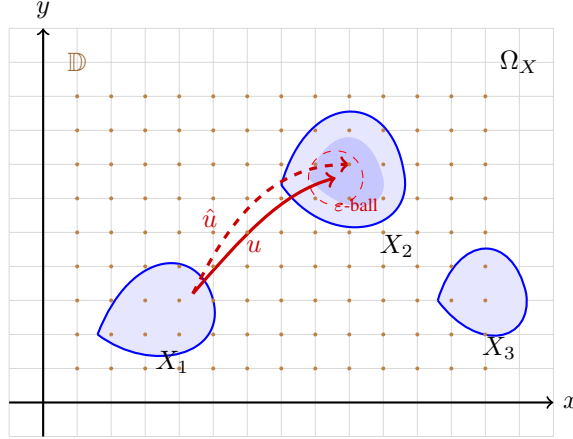


Figure 5: Given sufficient precision, the transitions of strongly ϵ -robust dynamics can be realized with approximate dynamics on a finite datatype, which gives a ϵ -covering for the state-space.

1639 **Lemma 40.** Consider η -finite dynamics $D = \langle X, U, f \rangle$, s.t. each component of X and U contains a
 1640 closed ball of radius ϵ (in Ω_X, Ω_U resp.)

1641 Then given datatypes $\mathbb{D}_X \subseteq X, \mathbb{D}_U \subseteq U$ with sufficient precision, there is a function $\hat{f} : \mathbb{D}_X \times \mathbb{D}_U \rightarrow$
 1642 \mathbb{D}_X s.t. $\langle \mathbb{D}_X, \mathbb{D}_U, \hat{f} \rangle$ simulates $\langle X, U, f \rangle$ with error ϵ .

1643 *Proof.* Suppose \mathbb{D}_X is an ϵ -covering of X , and \mathbb{D} is an ϵ -covering of U . Let $X_1, \dots, X_r, r \geq 1$ be the
 1644 connected components of X . Let $i \in 1..r$, we have by assumption, that for some $x_i \in X_i$

$$\overline{B}(x_i, \epsilon) \subseteq X_i$$

1645 Since \mathbb{D}_X is an ϵ -covering of X , there is some $d_i \in \mathbb{D}_X$ s.t. $\|x_i - d_i\| \leq \epsilon$, and therefore
 1646 $d_i \in \overline{B}(x_i, \epsilon) \subseteq X_i$.

1647 Similarly, there is an element of \mathbb{D}_U in each component of U . Now, we may construct \hat{f} as follows:
 1648 for $x \in \mathbb{D}_X$ and $u \in \mathbb{D}_U$

$$\hat{f}(x, u) = \arg \min_{y \in \mathbb{D}_X} \|y - f(x, u)\|$$

1649 with ties broken arbitrarily. Then, as \mathbb{D}_X is an ϵ -covering of X , $\|\hat{f}(x, u) - f(x, u)\| \leq \epsilon$ as
 1650 desired. \square

1651 We now have the setup, and necessary results for Theorem 2.

1652 **Theorem 2.** Every η -finite system with strongly robust dynamics can be implemented with floating-
 1653 point operations, given sufficient precision.

1654 *Proof.* Apply Lemma 39 and Lemma 40 to obtain a realisation of S using a finite datatype, e.g. using
 1655 \mathbb{D}_p or \mathbb{D}'_p for sufficiently large p . \square

1656 C.2 Parametrised Systems

1657 The stability of robust dynamics can also be a desirable property in the context of learning. Consider
 1658 a parametrised model describing the trained model. If the system described by the model is ϵ -robust
 1659 and it is sufficiently smooth with respect to its parameters, then perturbing the model parameters will
 1660 not change the behaviour of the system. Thus a robust system is intuitively more likely to be attained
 1661 by a learning algorithm.

1662 **Definition 29.** Let $f : \Theta \times \Omega_X \times \Omega_U \rightarrow \Omega$ be continuous. Write f_θ for the function $f(\theta, -, -)$. A
 1663 dynamics parametrised by Θ is of the form $D_\theta = \langle X, U, f_\theta \rangle$.

1664 **Theorem 41.** (Corollary 36.20 of [Willard, 2012]) A continuous functions on a compact metric
 1665 space X is uniformly continuous, that is for all $\epsilon > 0$ there exists $\delta > 0$ such that for all $x, y \in X$
 1666 $\|x - y\| \leq \delta \implies \|f(x) - f(y)\| \leq \epsilon$.

1667 **Theorem 42.** Let η -finite dynamics $D_\theta = \langle X, U, f_\theta \rangle$ be parametrised by Θ , and let Θ be compact.
 1668 Suppose D_θ is ϵ -robust (w.r.t Ω_X). Then for some $\delta > 0$, we have that for $\rho \in \Theta$ s.t. $\|\theta - \rho\| \leq \delta$
 1669 the dynamics $D_\rho = \langle X, U, f_\rho \rangle$ is well-defined. Moreover, for any system S_θ with dynamics D_θ , the
 1670 system S_ρ obtained by switching out D_θ for D_ρ has the same canonical automaton.

1671 *Proof.* Since D_θ is η -finite, we have that X and U are compact. Thus the Cartesian product
 1672 $\Theta \times X \times U$ is compact. Thus, by Theorem 41 for all $\epsilon > 0$ we have some $\delta > 0$ such that for all
 1673 $(\theta, x, u), (\rho, x, u) \in \Theta \times X \times U$

$$\|(\theta, x, u) - (\rho, x, u)\| \leq \delta \implies \|f(\theta, x, u) - f(\rho, x, u)\| \leq \epsilon$$

1674 Now, take $\rho \in \overline{B}_\Theta(\theta, \delta)$. We have for all $x \in X$ and $u \in U$ that

$$\begin{aligned} \|(\theta, x, u) - (\rho, x, u)\| &= \|\theta - \rho\| \leq \delta \\ \therefore \|f(\theta, x, u) - f(\rho, x, u)\| &\leq \epsilon \end{aligned}$$

1675 Thus $f(\rho, x, u) \in \overline{B}(f(\theta, x, u), \epsilon) \subseteq X$, since D_θ is ϵ -robust. Moreover, letting X_1, \dots, X_r be the
 1676 components of X and U_1, \dots, U_s be the components of U , we have that $X \cap X_i \neq \emptyset$ for $i \in 1..r$ and
 1677 $U \cap U_i$ for $i \in 1..s$. Thus D_ρ simulates D_θ with error ϵ .

1678 Now, the canonical semiautomaton for D_θ is $\langle \overline{X}, \overline{U}, \overline{f_\theta} \rangle$ and the canonical semiautomaton for D_ρ
 1679 is $\langle \overline{X}, \overline{U}, \overline{f_\rho} \rangle$. By Lemma 39, we have that $\overline{f_\theta}$ and $\overline{f_\rho}$ give the same transitions. Therefore the two
 1680 semiautomata are the exact same. Taking S_θ, S_ρ as in the statement, we see that they indeed must
 1681 have the same canonical automaton. \square

1682 C.3 Robust Cascade Decomposition and Proof of Theorem 5

1683 Coming back to connecting functions discussed in Appendix B.5, we have the following refinement
 1684 of the result.

1685 **Theorem 43.** Let D be a strongly robust η -finite dynamics, which are a realisation of FLIP-FLOP.
 1686 Then all group-free functions can be implemented by some strongly robust serial cascade of D
 1687 components. Moreover, the connection functions in such cascade can be given by depth-2 MLPs.

1688 *Proof.* Say $D = \langle X, U, f \rangle$ is strongly ϵ -robust. By Theorem 35, for any group-free function F ,
 1689 there is a serial cascade C of D -components which can implement it. By Lemma 36, C is also
 1690 strongly robust. Say, $C = \xrightarrow{g_1} D_1 \dots \xrightarrow{g_L} D_L = \langle X^L, U', f_C \rangle$, with U' an η -finite space, $D_i \equiv D$
 1691 and $g_i : U' \times X^{i-1} \rightarrow U$.

1692 Let U_1, \dots, U_n be the η -components of U . By strong robustness, for each $i \in [1..n]$, there is $u_i \in U_i$
 1693 s.t. $B_{\Omega_U}(u_i, \epsilon) \subseteq U_i$. By Lemma 22, we can w.l.o.g. assume that g_i has its image in $\{u_1, \dots, u_n\}$,
 1694 while still inducing the same mapping $\overline{U'} \times \overline{X}^{i-1} \rightarrow \overline{U}$.

1695 By Theorem 16, there is a MLP $N_i : \Omega_{U'} \times \Omega_X^{i-1} \rightarrow \Omega_U$ which ϵ -approximates g_i , since $U' \times X^{i-1}$ is
 1696 compact and g_i continuous. For $\langle u', x_1, \dots, x_{i-1} \rangle \in U' \times X^{i-1}$ we have $f_i(\langle u', x_1, \dots, x_{i-1} \rangle) =$
 1697 u_j for some $j \in [1..n]$, so

$$N(\langle u', x_1, \dots, x_{i-1} \rangle) \in B_{\Omega_U}(u_j, \epsilon) \subseteq U_j$$

1698 Thus N_i sends elements of $U' \times X^{i-1}$ to the same η -components of U as g_i . Moreover, N_i is
 1699 continuous.

1700 Overall, the canonical automaton for $\xrightarrow{g_1} D_1 \xrightarrow{g_2} \dots \xrightarrow{g_L} D_L$ is the same as the canonical automaton for
 1701 $\xrightarrow{N_1} D_1 \xrightarrow{N_2} \dots \xrightarrow{N_L} D_L$. Thus the strongly robust cascade with D components and MLP connections
 1702 can implement F . \square

1703 Appendix G.3 shows constructions for strongly robust η -finite xLSTM FLIP-FLOP and R_2 dynamics.
 1704 All together, we obtain Theorem 5:

1705 **Theorem 5** (xLSTM does start-free robustly). *All star-free languages can be recognised by xLSTM*
1706 *cascades, as well as by floating-point implementations of xLSTM cascades given sufficient precision.*

1707 *Proof.* We have that there are strongly robust xLSTM dynamics that realise FLIP-FLOP and R_2 .
1708 Thus by Theorem 43, every group-free function can be implemented by a cascade of strongly robust
1709 xLSTM dynamics. Any such cascade is itself strongly robust, by Lemma 36, and thus can be realized
1710 by floating-point operations, given sufficient precision, by Theorem 2 \square

1711 Moreover, by Theorem 43 we know that for these cascades, it suffices to use MLP connecting
1712 functions. By Theorem 42 we also have that the parametrizations of sLSTM blocks which yields
1713 FLIP-FLOP and R_2 can also be changed, within some δ , retaining the behaviour of the dynamics.

D Expressivity Results for State Space Models

In this Appendix we reap rewards of establishing the preliminary framework of Metric Automata Theory for η -finite dynamics. We can now prove expressivity results by establishing structural properties of dynamics, which are preserved by feed-forward cascades, and which are generally applicable.

In Section D.1 we introduce the notion of *contracting* dynamics, which describes dynamics that are not able to keep track of a state over unbounded input lengths. We use this notion to prove Theorems 3 and 4.

In Section D.2 we introduce another structural property, called *aperiodicity*. It is the η -finiteness corresponding notion to group-freeness in Finite Automata. We use aperiodicity to prove Theorem 6.

Finally, in Section D.3 we focus on the SSM parametrisation of Mamba, and prove Theorem 7.

D.1 Contracting Dynamics and Proofs of Theorems 3 and 4

Definition 30. Call η -finite dynamics $\langle X, U, f \rangle$ a *contracting dynamics*, if for any initial points $x_0, x'_0 \in X$ and eventually equivalent input sequences $(u_n)_{n \geq 1}, (u'_n)_{n \geq 1} \subseteq U$, we have that the corresponding state sequences $(x_n)_{n \geq 1}, (x'_n)_{n \geq 1} \subseteq U$ are eventually equivalent.

Thus, for a contracting dynamics, it does not matter what state the evaluation of the inputs starts from—eventually all initial states lead to the same behaviour under a fixed input sequence. The intuition behind the name is the following—eventually all possible states that the dynamics could be in under the input sequence collapse to a single η -component.

Example 9. Clearly, all Finite Context Dynamics (Definition 18) are contracting. ■

Lemma 44. Let $C = D_1 \rightsquigarrow \dots \rightsquigarrow D_n$ be a cascade of η -finite contracting dynamics. Then C is a contracting dynamics.

Proof. By induction, it is sufficient to show the statement for $n = 2$.

Let us consider $C = D_1 \rightsquigarrow D_2$ with $D_1 = \langle X, U, f_1 \rangle$ and $D_2 = \langle Z, U \times X, f_2 \rangle$. The dynamics function of the cascade is:

$$f(\langle x, z \rangle, u) = \langle f_1(x, u), f_2(z, u') \rangle \quad \text{where } u' = \langle u, f_1(x, u) \rangle$$

Consider arbitrary $\langle x_0, z_0 \rangle, \langle x'_0, z'_0 \rangle \in X \times Z$ and $(u_t)_{t \geq 1}, (u'_t)_{t \geq 1} \in U^\omega$, eventually equivalent in U . Take

$$\langle x_n, z_n \rangle = (D_1 \rightsquigarrow D_2)(\langle x_0, z_0 \rangle, u_{[1..n]}); \quad \langle x'_n, z'_n \rangle = (D_1 \rightsquigarrow D_2)(\langle x'_0, z'_0 \rangle, u'_{[1..n]})$$

By inductive hypothesis, D_1 is contracting, and so since we have

$$x_n = D(x_0, u_{[1..n]}); \quad x'_n = D(x'_0, u'_{[1..n]})$$

we have that $(x_n)_{n \geq 1}, (x'_n)_{n \geq 1} \in X^\omega$ are eventually equivalent. Thus also $(\langle u_n, x_{n+1} \rangle)_{n \geq 1}, (\langle u'_n, x'_{n+1} \rangle)_{n \geq 1} \in (U \times X)^\omega$ are eventually equivalent.

Note that we have $z_{n+1} = f_2(z_n, \langle u_n, x_{n+1} \rangle)$ and $z'_{n+1} = f_2(z'_n, \langle u'_n, x'_{n+1} \rangle)$. Since D_2 is by assumption contracting, and the two input sequence are eventually equivalent by continuity of f_n , we get that $(z_n)_{n \geq 1}, (z'_n)_{n \geq 1} \in Z^\omega$ are eventually equivalent.

So, overall $(\langle x_n, z_n \rangle), (\langle x'_n, z'_n \rangle) \in (X \times Z)^\omega$ are eventually equivalent. □

Lemma 45. Suppose a η -finite Linear Recurrent Dynamics D is ϵ -robust. Then D is contracting.

Proof. Suppose that $D = \langle X, U, f \rangle$ is ϵ -robust.

Let $x_0, x'_0 \in X$ and $(u_n)_{n \geq 1}, (u'_n)_{n \geq 1} \subseteq U$ which are eventually equivalent—say for $n \geq N$. For each component of U , say U_1, \dots, U_k , define a representative element r_1, \dots, r_k . Define $(\tilde{u}_n)_{n \geq 1} \subseteq U$ to be such that $\tilde{u}_n = r_c$ where U_c is the component containing u_{n+N} . Thus $(\tilde{u}_n)_{n \geq 1}$ is equivalent to $(u_{n+N})_{n \geq 1}$ and $(u'_{n+N})_{n \geq 1}$.

1754 Write $A_n = A(\tilde{u}_n)$ and $B_n = B(\tilde{u}_n)$ and $f_n(x) = f(x, \tilde{u}_n)$. For $S \subseteq \Omega$, define

$$\Delta S = \{\alpha \cdot (x - y) : \alpha \in [0, 1], x, y \in S\}$$

1755 For $\beta \in \mathbb{R}_{\geq 0}$, write $\beta \cdot S = \{\beta \cdot s : s \in S\}$. Take $M = \sup_{x, y \in X} \|x - y\|$. We have that M is
 1756 finite, since X is compact, and hence bounded. Also, denote $X^{(0)} = X$, $X^{(n+1)} = \{f(x, \tilde{u}_n) : x \in$
 1757 $X^{(n)}\} = \{D(x)\}$.

1758 We have, by induction that $\Delta(X^{(n)}) \subseteq (\frac{M}{M+2n\epsilon}) \cdot \Delta(X)$: for $n = 0$ this is immediate.

1759 For $n \geq 1$, by inductive hypothesis we have $\Delta(X^{(n-1)}) \subseteq (\frac{M}{M+2(n-1)\epsilon}) \cdot \Delta(X)$. Consider $u \neq 0$,
 1760 $u \in \Delta(X^{(n)})$. Take $v = \frac{u}{\|u\|}$. We have that

$$u = \beta \cdot (f_n(x) - f_n(y)) = \beta \cdot A_n(x - y)$$

1761 for some $x, y \in X^{(n-1)}$ and $\beta \in [0, 1]$. We have that $x - y \in \Delta(X^{(n-1)}) \subseteq (\frac{M}{M+2(n-1)\epsilon}) \cdot \Delta(X)$,
 1762 so for some $x', y' \in X$ we have

$$x' - y' = \left(\frac{M}{M+2(n-1)\epsilon}\right)^{-1} \cdot (x - y)$$

1763 Now:

$$\|f_n(x') - (f_n(x') + \epsilon \cdot v)\| = \epsilon \quad \text{and} \quad \|f_n(y') - (f_n(y') + \epsilon \cdot v)\| = \epsilon$$

1764 so by robustness, $f_n(x') + \epsilon \cdot v \in X$ and $f_n(y') - \epsilon \cdot v$. Thus

$$\begin{aligned} \Delta X &\ni (f_n(x') + \epsilon \cdot v) - (f_n(y') - \epsilon \cdot v) \\ &= f_n(x') - f_n(y') + 2\epsilon \cdot v \\ &= A_n(x' - y') + 2\epsilon \cdot v \\ &= \left(\frac{M}{M+2(n-1)\epsilon}\right)^{-1} \cdot A_n(x - y) + 2\epsilon \cdot v \\ &= \left(\left(\frac{M}{M+2(n-1)\epsilon}\right)^{-1} \cdot \beta^{-1} + \frac{2\epsilon}{\|u\|}\right) \cdot u \end{aligned}$$

1765 So, we have $u = \gamma \cdot l$ for some $l \in \Delta X$ and

$$\begin{aligned} \gamma^{-1} &= \left(\frac{M}{M+2(n-1)\epsilon}\right)^{-1} \cdot \beta^{-1} + \frac{2\epsilon}{\|u\|} \\ &= \frac{M+2(n-1)\epsilon}{M} \cdot \beta^{-1} + \frac{2\epsilon}{\|u\|} \\ \text{as } \beta^{-1} \geq 1 \text{ and } \|u\| \leq M &\geq \frac{M+2(n-1)\epsilon}{M} + \frac{2\epsilon}{M} \\ &= \frac{M+2n\epsilon}{M} \end{aligned}$$

1766 So $u \in (\frac{M+2n\epsilon}{M}) \cdot \Delta(X)$, and thus indeed $\Delta(X^{(n)}) \subseteq (\frac{M}{M+2n\epsilon}) \cdot \Delta(X)$. Therefore
 1767 $\sup_{x, x' \in X^{(n)}} \|x - x'\| \rightarrow 0$ as $n \rightarrow \infty$.

1768 Now, consider the state-sequences $(D(x_0, u_{[1..n]}))_{n \geq 1}, (D(x'_0, u'_{[1..n]}))_{n \geq 1}$. We have by Lemma 24

$$\begin{aligned} D(x_0, u_{[1..(n+N)]}) &= D(x_N, u_{[(N+1)..(n+N)]}) \\ &\sim_X D(x_N, \tilde{u}_{[1..n]}) \end{aligned}$$

1769 and similarly $D(x'_0, u'_{[(1+N)..(n+N)]}) \sim_X D(x'_N, \tilde{u}_{[1..n]})$. Now, $D(x'_N, \tilde{u}_{[1..n]}), D(x_N, \tilde{u}_{[1..n]}) \in$
 1770 $X^{(n)}$. Thus we have

$$\|D(x'_N, \tilde{u}_{[1..n]}) - D(x_N, \tilde{u}_{[1..n]}) \in X^{(n)}\| \rightarrow 0 \text{ as } n \rightarrow \infty$$

1771 Therefore, eventually $D(x'_N, \tilde{u}_{[1..n]})$ and $D(x_N, \tilde{u}_{[1..n]}) \in X^{(n)}$ are in the same η -component of
 1772 X . \square

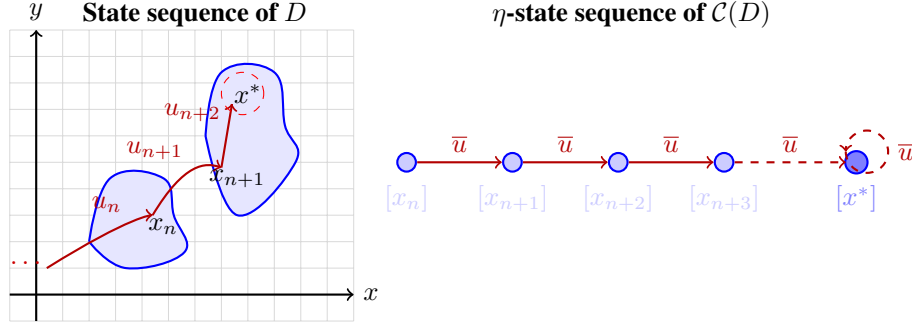


Figure 6: State sequence of aperiodic dynamics under iterated input always η -converges.

Theorem 3 (Non-robustness of LRDs). *Suppose an η -finite LRD D is such that its canonical semiautomaton D_A has at least two states, and an input inducing an identity transformation. Then D cannot be ϵ -robust for any $\epsilon > 0$.*

Proof. Let $D = \langle X, U, f \rangle$ be an η -finite LRD, such that its canonical semiautomaton $D = \langle \bar{X}, \bar{U}, \bar{f} \rangle$ has at least two distinct η -states, say \bar{x}, \bar{x}' , and an input \bar{u} inducing identity transformation on \bar{X} .

For contradiction suppose that D is robust. Then by Lemma 45, D is contracting. Thus for $x_0 \in \bar{x}, x'_0 \in \bar{x}'$ and $u \in \bar{u}$ we have that the sequences $(x_n = D(x_0, u^n))_{n \geq 1}, (x'_n = D(x'_0, u^n))_{n \geq 1} \in X^\omega$ are eventually equivalent. Since $[u]_{\sim_U} = \bar{u}$ induces the identity transformation of \bar{X} , we have that the corresponding sequences $([x_n]_{\sim_X})_{n \geq 1}, ([x'_n]_{\sim_X})_{n \geq 1} \in \bar{X}^\omega$ are constant, equal \bar{x}, \bar{x}' respectively. Thus necessarily $\bar{x} = \bar{x}'$. This is a contradiction. \square

Lemma 46. *Contracting dynamics cannot implement the state-sequence function of FLIP-FLOP.*

Proof. Consider a system S with some encoder $\text{enc} : \{\text{set}, \text{reset}, \text{id}\} \rightarrow U$ and decoder $\text{dec} : Y \rightarrow \{\text{low}, \text{high}\}$. Suppose that the dynamics $D = \langle X, U, f \rangle$ of S are contracting. Consider $x_0 \in X$ and input sequences $(u_n)_{n \geq 1}, (u'_n)_{n \geq 1} \subseteq U$, given by

$$u_1 = h; \quad u'_1 = l; \quad u_n = u'_n = i \quad \text{for } n > 1$$

They are eventually equivalent, and so the corresponding state sequences $x_n = D(x_0, \langle u_{1..n} \rangle)$ and $x'_n = D(x_0, \langle u'_{1..n} \rangle)$ are also eventually equivalent. Thus

$$\text{dec} \circ S(\text{enc}(u_{1..n})) = \text{dec} \circ S(\text{enc}(u'_{1..n})) \in \{\text{high}, \text{low}\}$$

for large enough n , since $\{\text{high}, \text{low}\}$ is a discrete space.

However, the two sequences of inputs correspond to different flip flop states - thus D cannot be a dynamics for a system that implements a flip flop. \square

Theorem 4 (LRDs cannot do FLIP-FLOP robustly). *FLIP-FLOP cannot be implemented by a cascade of η -finite ϵ -robust LRDs for any $\epsilon > 0$.*

Proof. A cascade of such LRDs is contracting by Lemmas 45 and 44. Thus, by Lemma 46, it cannot implement FLIP-FLOP. \square

D.2 Aperiodic Dynamics and Proof of Theorem 6

Definition 31. For a η -finite space X , we say a sequence $(x_n)_{n \geq 1} \subseteq X$ η -converges in X , if eventually all its terms lie in the same η -component of X .

If the sequence of states of a system η -converges, it means that the behaviour of that system is eventually the same.

Definition 32. Call a η -finite dynamics $D = \langle X, U, f \rangle$ *aperiodic*, if for all $x_0 \in X$ and input sequences $(u_n)_{n \geq 1} \subseteq U$ η -convergent in U , we have that the corresponding state sequence $(x_n)_{n \geq 1} \subseteq X$ is η -convergent in X .

1804 An example of a aperiodic dynamics is given by the FLIP-FLOP dynamics. An input sequence that
 1805 η -converges must eventually be constantly `set` or `reset`. In that case, the state is eventually high,
 1806 low respectively.

1807 **Lemma 47.** *Let D be a η -finite Linear Recurrent Dynamics, with $A(u)$ having all its eigenvalues*
 1808 *being non-negative. Then D is aperiodic.*

1809 *Proof.* This is a similar argument as for Theorem 1 in Grazzi et al. [2025], with some simplifications
 1810 stemming from the fact that we can use associativity of linear operations freely.

1811 Let $D = \langle X, U, f \rangle$ be an η -finite Linear Recurrent Dynamics, with $X \subseteq \mathbb{R}^d$, s.t. $A(u)$ has all its
 1812 eigenvalues being real, for all $u \in U$. Say $f(x, u) = A(u) \cdot x + B(u)$.

1813 Consider a sequence $(u_n)_{n \geq 1} \in U^\omega$, η -convergent in U , and $x_0 \in X$. Let $(x_n =$
 1814 $D(x_0, u_{1..n}))_{n \geq 1} \in X^\omega$ be the corresponding state sequence. We have some N s.t. for $n \geq N$ all
 1815 u_n are contained in the same component of U , we may pick a representative $r \in U$ of that component.

1816 Write $A = A(r)$, $B = B(r)$. By Lemma 25, we have for $n \geq N$ that

$$x_{n+N} \sim_X x'_n = D(x_N, r^{n-N})$$

1817 We consider the state sequence in the diagonalized space of A . Write $A = P^{-1}JP$ for the Jordan
 1818 normal form of A . Here J is block diagonal, with say blocks J_1, \dots, J_s , $J_b \in \mathbb{R}^{m_b \times m_b}$ being a
 1819 Jordan Block with λ_b on the diagonal being an eigenvalue of A , and 1 on the right off-diagonal. Also
 1820 $P \in \mathbb{R}^{d \times d}$, since all eigenvalues of A are real.

1821 Take $\bar{x}_n = Px'_n$, then we have

$$\begin{aligned} \bar{x}_{n+1} &= P(Ax'_n + B) = P(AP^{-1}\bar{x}_n + B) \\ &= P(P^{-1}JPP^{-1}\bar{x}_n + B) = J\bar{x}_n + PB \end{aligned}$$

1822 We will consider the difference $z_n = \bar{x}_{n+1} - \bar{x}_n$. Unrolling the recurrence we get

$$z_n = J^n(J\bar{x}_0 - \bar{x}_0) = J^n z_0$$

1823 The i -th entry of this difference, where i is in say the b -th block of J , is

$$[z_n]_i = \sum_{j=i}^{m_b} \lambda_b^{n+i-j} \binom{n}{j-1} [z_0]_j$$

1824 This is of the form considered in Lemma 64. Thus, $[z_n]_i \in \mathbb{R}$ is eventually monotone, and so it either
 1825 converges in \mathbb{R} or is unbounded as $n \rightarrow \infty$.

1826 Now, if $z_n \rightarrow 0$, that is $[z_n]_i \rightarrow 0$ for all $i \in [1..d]$, then we have that also, by continuity of linear maps,
 1827 $x'_{n+1} - x'_n = P^{-1}z_n \rightarrow 0$, so that x'_n must eventually be in the same component of X by Lemma
 1828 20. Therefore also $(x_n)_{n \geq 1}$ is η -convergent in X .

1829 Otherwise, one of the entries of z_n either is unbounded, or converges to a non-zero limit. In both
 1830 cases, the corresponding entry of x_n is unbounded as $n \rightarrow \infty$, and so this is impossible in a η -finite
 1831 space X .

1832 Overall, this shows that D must be aperiodic. □

1833 **Lemma 48.** *Let $D = \langle X, U, f \rangle$ be a η -finite Finite Context Dynamics. Then D is aperiodic.*

1834 *Proof.* Let l be the context length of D . Let $x_0 \in X$ and $(u_n)_{n \geq 1} \in U^\omega$ be η -convergent in U . Let
 1835 $\bar{u} \in U$ lie in the component of U which contains the tail of $(u_n)_{n \geq 1}$, say for $n \geq N$. For $n \geq N + l$
 1836 we have that $u_{n-l+1}, \dots, u_n \sim \bar{u}$, and so

$$x_n = C(\langle u_{n-l+1..n} \rangle) \sim C(\bar{u}^l)$$

1837 Thus x_n is in the component of X containing $C(\bar{u}^l)$. □

1838 **Lemma 49.** *Let $C = D_1 \rightsquigarrow \dots \rightsquigarrow D_k$ be a cascade of η -finite aperiodic dynamics D_1, \dots, D_k .*
 1839 *Then C is aperiodic.*

1840 *Proof.* By induction, it is sufficient to show the statement for $n = 2$.

1841 Let us consider $C = D_1 \rightsquigarrow D_2$ with $D_1 = \langle X, U, f_1 \rangle$ and $D_2 = \langle Z, U \times X, f_2 \rangle$. The dynamics
 1842 function of the cascade is:

$$f(\langle x, z \rangle, u) = \langle f_1(x, u), f_2(z, u') \rangle \quad \text{where } u' = \langle u, f_1(x, u) \rangle$$

1843 Consider a sequence $(u_t)_{t \geq 1} \in U^\omega$ η -convergent in U , and $\langle x'_0, x_0 \rangle \in X' \times X$.

1844 As D_1 is aperiodic, the corresponding sequence $(x_n)_{n \geq 1} \subseteq X^\omega$ is η -convergent in X . Equivalently,
 1845 $(x_{t+1})_{t \geq 1}$ is η -convergent in X . Moreover, then the sequence $(u'_n = \langle u_n, x_{n+1} \rangle)_{n \geq 1}$ is η -convergent
 1846 in $U \times X$. Since D_2 is aperiodic, the sequence $(z_n)_{n \geq 1} \in Z$ is therefore η -convergent in Z .

1847 All together, $(\langle x_n, z_n \rangle)_{n \geq 1}$ is η -convergent in $X \times Z$. \diamond \square

1848 **Theorem 50.** *η -finite dynamics are aperiodic if and only if their canonical semiautomaton is group-*
 1849 *free*

1850 *Proof.* Let $D = \langle X, U, f \rangle$ have canonical semiautomaton $D_A = \langle \bar{X}, \bar{U}, \bar{f} \rangle$

1851 (\Rightarrow) First, suppose that D_A is not group-free. By Theorem 13, there exist some $S \subseteq \bar{X}$ and $\bar{u} \in \bar{U}$
 1852 s.t. $\bar{f}(-, \bar{u})$ induces a non-trivial permutation on S . That is, since S is a finite set, we have $s \in \bar{X}$
 1853 s.t. $D_A(s, \bar{u}^n) \neq D_A(s, \bar{u}^{n+1})$ for all $n \geq 1$. Here \bar{u}^n denotes the word of length n consisting of
 1854 repeated symbol \bar{u} .

1855 Take $u \in U$ s.t. $[u]_{\sim_U} = \bar{u}$ and $x \in X$ s.t. $[x]_{\sim_X} = s$. Then, we have that for all $n \geq 1$ that

$$[D(x, u^n)]_{\sim_X} \neq [D(x, u^{n+1})]_{\sim_X}$$

1856 The input sequence $(u^n)_{n \geq 1}$ is η -convergent in U , but the corresponding state sequence
 1857 $(D(x, u^n))_{n \geq 1}$ is not. Thus, D is not aperiodic.

1858 (\Leftarrow) Now, suppose that D_A is group free. By Theorem 12, D_A can be realized by a *serial* cascade
 1859 of FLIP-FLOPS, say C . We also have, that C can be realized by a *feed-forward* cascade C' of
 1860 FLIP-FLOPS and repeat semiautomata, all of which are aperiodic (as repeat semiautomata are FCDs).
 1861 Thus by Lemma 49, C' is aperiodic. It remains to show that dynamics realised by aperiodic dynamics
 1862 are also aperiodic.

1863 Let (α, ι, ζ) be an assignment of D_A into C' . Consider an η -convergent input sequence $(\bar{u}_n)_{n \geq 1} \subseteq \bar{U}$
 1864 and $\bar{x}_0 \in \bar{X}$, with the corresponding state sequence $(\bar{x}_n = D_A(\bar{x}_0, \bar{u}_{[1..n]}))_{n \geq 0} \subseteq \bar{X}$. Since
 1865 $(\bar{u}_n) \subseteq \bar{U}$ is η -convergent, it is in fact eventually constant, since \bar{U} is a discrete space.

1866 Since C' realizes D_A , by Theorem 14, we have, for $\bar{x}'_0 \in \alpha(\bar{x}_0)$

$$\mathcal{M}(D_A)(\bar{x}_0, \bar{u}_{[1..n]}) = \zeta \circ \mathcal{M}(C)(\bar{x}'_0, \iota(\bar{u}_{[1..n]}))$$

1867 where $\mathcal{M}(D_A), \mathcal{M}(C)$ are the canonical machines for D_A, C , respectively. Now, (\bar{u}_n) is even-
 1868 tually constant and so also $(\iota(\bar{u}_n))$ is eventually constant. C is aperiodic, and so the sequence
 1869 $C(\bar{x}'_0, \iota(\bar{u}_{[1..n]}))$ is η -convergent (and thus eventually constant, as C is a semiautomaton). All
 1870 together

$$\mathcal{M}(D_A)(\bar{x}_0, \bar{u}_{[1..n]}) = \zeta \circ \mathcal{M}(C)(\bar{x}'_0, \iota(\bar{u}_{[1..n]})) = \zeta(C(\bar{x}'_0, \iota(\bar{u}_{[1..n]})), \iota(\bar{u}_n))$$

1871 by def. of canonical machines, and therefore this sequence is also eventually constant.

1872 Thus the state sequence $D_A(\bar{x}_0, \bar{u}_{[1..n]})$ itself is eventually constant.

1873 Equivalently, by Lemma 26, for any $s\eta$ -convergent sequence $(u_n) \subseteq U$ and $x_0 \in X$ the state
 1874 sequence $(D(x_0, u_{[1..n]})) \subseteq X$ is η -convergent, and so D is indeed aperiodic. \square

1875 D.3 Parametrisation of Mamba and Proof of Theorem 7

1876 Sarrof et al. [2024] show that any star-free language can be recognized by an SSM like Mamba (Gu
1877 et al. [2022]), using the Krohn and Rhodes Theorem from Algebraic Automata Theory. However, in
1878 their construction, they assume that gates of the form $A(u) = \mathbf{0}$ can be used, which is not the case for
1879 architectures utilizing strictly positive parametrization, like Mamba.

1880 We show in Construction 3 a modified η -finite system construction, which only requires gates with
1881 diagonal entries in the range $[\epsilon, 1]$, for a suitable $\epsilon > 0$. As it turns out, further restricting diagonal
1882 entries to lie in $(-1, 1)$ makes it impossible to implement a flip flop.

1883 Mamba ([Gu et al., 2022]) parametrization is of the form

$$A(u) = \text{Diag}(\exp(-\Delta_u \odot \exp(z_u))) \quad \text{where } z_u \in \mathbb{R}^d, \Delta_u \in (0, \infty)^d$$

1884 and \odot is the element-wise product $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$. This gives $-\Delta_u \odot \exp(z_u)_i < 0$ for $i \in 1 \dots d$,
1885 and thus $A(u)_i \in (0, 1)$ for $i \in 1 \dots d$. We will show in this section that an SSM using Mamba
1886 blocks cannot implement a flip flop for unbounded .

1887 However, experimental results in [Sarrof et al., 2024] show that this architecture does well in
1888 experimental evaluations and demonstrates length generalization for star-free modelling tasks. For
1889 tasks involving periodic modelling, the model fails to length generalize. This motivates us to
1890 investigate the geometric complexity of the state space when evaluated on sequences of bounded
1891 length in Appendix E.

1892 **Construction 3.** There is a η -finite system with Linear Recurrent Dynamics with diagonal entries in
1893 $[\epsilon, 1]$, for some $\epsilon > 0$, which realize FLIP-FLOP dynamics.

1894 Take $\epsilon = 1/4$. Consider $X = X_l \cup X_h \subseteq \mathbb{R}$, where

$$X_l = \bar{B}(1, \epsilon); \quad X_h = \bar{B}(2, \epsilon)$$

1895 Then X_{q_0}, X_l, X_h are the components of X , and X is η -finite. Take $U, e : \{s, r, i\} \rightarrow U$ and
1896 $f : X \times U \rightarrow X$ to be such that

$$f(x, e(\sigma)) = A_\sigma \cdot x + B_\sigma \quad \text{where } (A_\sigma, B_\sigma) = \begin{cases} (1, 0) & \text{if } \sigma = i \\ (\epsilon/4, 1) & \text{if } \sigma = r \\ (\epsilon/4, 2) & \text{if } \sigma = s \end{cases}$$

1897 We have $X \subseteq \bar{B}(0, 2 + \epsilon)$, and so $(\epsilon/4 \cdot -)(X) \subseteq \bar{B}(0, \epsilon/4 \cdot (2 + \epsilon)) \subseteq \bar{B}(0, \epsilon)$. Thus we see
1898 that f maps X to X_l under input r and to X_h under input s . Under input i , f acts as identity.
1899 Thus these dynamics indeed realize FLIP-FLOP, through assignment that identifies with α mapping
1900 $\text{high} \mapsto X_h, \text{low} \mapsto X_l, \iota \text{ mapping set} \mapsto s, \text{reset} \mapsto r, \text{id} \mapsto i$ and ζ mapping X_l to low and
1901 X_h to high.

1902 **Lemma 51.** Let $D = \langle X, U, f \rangle$ be an η -finite Linear Recurrent dynamics with $A(u)$ diagonal, with
1903 entries in $(-1, 1)$ for all $u \in U$. Then D is contracting.

1904 *Proof.* Let $x_0, x'_0 \in X$ and $(u_n)_{n \geq 1} \subseteq U$. For each component of U , say U_1, \dots, U_k , define a
1905 representative element r_1, \dots, r_k . Define $(u'_n)_{n \geq 1} \subseteq U$ to be such that $u'_n = r_c$ where U_c is the
1906 component containing u_n . Thus $(u'_n)_{n \geq 1}$ is equivalent to $(u_n)_{n \geq 1}$, and $(u'_n)_{n \geq 1}$ takes finitely many
1907 values r_1, \dots, r_k .

1908 Now, consider A_1, \dots, A_k , where $A_c = A(r_c)$. For each $c \in [1..k]$, let λ_c be the largest size
1909 eigenvalue of A_c . Then we have $|\lambda_c| < 1$, and

$$\|A_c \cdot x\|_2 \leq |\lambda_c| \cdot \|x\|_2 \quad \forall x \in X$$

1910 Let $\lambda \in \arg \max_{c \in 1..k} |\lambda_c|$, then we have $|\lambda| < 1$ and

$$\|A(r_c) \cdot x\|_2 \leq |\lambda| \cdot \|x\|_2 \quad \forall x \in X, c \in 1..k$$

1911 Now, we have that for the state sequences $(x_n)_{n \geq 1}, (x'_n)_{n \geq 1}$ corresponding to initial states x_0, x'_0
 1912 resp., and the input sequence $(u'_n)_{n \geq 1}$, the following holds:

$$\begin{aligned}
 \|x_n - x'_n\|_2 &= \left\| (A(u'_n) \cdot x_{n-1} + B(u'_n)) - (A(u'_n) \cdot x'_{n-1} + B(u'_n)) \right\|_2 \\
 &= \left\| A(r_c) \cdot (x_{n-1} - x'_{n-1}) \right\|_2 \quad \text{for some } c \in [1..k] \\
 &\leq |\lambda| \cdot \|x_{n-1} - x'_{n-1}\|_2 \\
 &\leq \dots \\
 &\leq |\lambda|^n \cdot \|x_0 - x'_0\|_2 \rightarrow 0 \quad \text{as } n \rightarrow \infty
 \end{aligned}$$

1913 Thus eventually x_n and x'_n must be in the same component of X . □

1914 Altogether, we arrive at the following result (for η -finite dynamics), restated here more precisely than
 1915 in the main body.

1916 **Theorem 7.** *SSMs with Mamba parametrisation cannot recognise FLIP-FLOP as η -finite systems.*

1917 *Proof.* Mamba blocks are feed-forward cascades of LRDs of the type considered in Lemma 51 and
 1918 convolution blocks (FCDs)—see Figure 2. Thus η -finite feed-forward cascades of Mamba blocks are
 1919 contracting, and so by Lemma 46, cannot implement FLIP-FLOP. □

E Geometrically Constrained Systems

In this appendix, we depart the setting of η -finiteness, and explore *geometrically-constrained systems* (GCSs). This setting allows for systems implementing functions beyond regular, but shares many properties with the η -finite setting. We develop the theory of GCS to explain empirical capabilities of Mamba, and to showcase the flexibility and generalizability of Metric Automata Theory.

In Section E.1 we develop a notion analogous to *aperiodicity* from Section D.2. We then prove Theorem 9.

In Section E.2 we introduce a generalisation of η -finiteness, called *weak η -finiteness*. We use it to argue that the cascade decomposition results for η -finite dynamics still apply to dynamics with convex-covering state-spaces.

In Section E.3 we show that η -finite dynamics are a special case of convex-constrained dynamics. Finally, we show a construction of a FLIP-FLOP using a Mamba convex-constrained SSM, and argue using weakly η -finiteness that Theorem 8 holds.

Definition 33. For $\Omega = \mathbb{R}^d$ or \mathbb{C}^d , we call $C \subseteq \Omega$ a *convex-covering* if C is a finite union of open, convex sets in Ω . We say that $X \subseteq \Omega$ is *convex-covered* by C if $X \subseteq C$.

We say X is *convex-separated* by C if (i) it is convex-covered by C and (ii) each path-connected component of C contains at most one path-connected component of X . ■

Note: any convex set in $\Omega = \mathbb{R}^d$ or \mathbb{C}^d is path-connected. Thus any convex-covering C has finitely many path-connected components.

Definition 34. Let $\Omega = \mathbb{R}^d$ or $\Omega = \mathbb{C}^d$, and let $C \subseteq \Omega$. We say that dynamics $D = \langle X, U, f \rangle$ are *convex-covered* by C if X is convex-covered by C . We define a *system geometrically-constrained by C* as a tuple $S_C = \langle X, U, f, C, x_0, Y, h \rangle$, where its dynamics $\langle X, U, f \rangle$ is a dynamics convex-covered by C , $x_0 \in X$ is the initial state, and $h : C \times U \rightarrow Y$ is the continuous output function. ■

The difference between a shortcut system and a system is that the dynamics function is defined only on X , while the output function is define on the convex-covering C .

We extend the definition of implementing a function to shortcut systems: S_C implements $F : \Sigma^+ \rightarrow \Gamma$ with encoder $\text{enc} : \Sigma \rightarrow U$ and decoder $\text{dec} : Y \rightarrow \Gamma$ if enc, dec are continuous and $F(w) = \text{dec} \circ S(\text{enc}(w))$.

Lemma 52. For a cascade $\mathcal{D} = D_1 \rightsquigarrow \dots \rightsquigarrow D_n$ with D_i convex-covered/convex-separated by C_i we have that \mathcal{C} is convex-covered/convex-separated by $\mathcal{C} = C_1 \times \dots \times C_n$

Proof. Suppose D_i is convex-covered by C_i for $i \in [1..n]$. First, $C_1 \times \dots \times C_n$ is indeed a convex-covering. A product of convex sets is convex, and so a product of finite unions of convex sets is also a finite union of convex sets (by commutativity of set product and union, see proof of Lemma 19). Thus, $X_1 \times \dots \times X_n \subseteq \mathcal{C}$ and \mathcal{D} is convex-covered by \mathcal{C} .

Now, suppose further that D_i is convex-separated by C_i for $i \in [1..n]$. The path-connected components of \mathcal{C} are of the form $\prod_{i=1}^n G_i$, where G_i is a path-connected component of C_i . Similarly, path-connected components of $X = X_1 \times \dots \times X_n$ are of the form $\prod_{i=1}^n Z_i$ where Z_i is a path-connected component of X_i .

We have that $\prod_{i=1}^n Z_i$ intersects $\prod_{i=1}^n G_i$ precisely when Z_i intersects G_i for each $i \in [1..n]$. Hence, there is exactly one component of \mathcal{C} intersecting $\prod_{i=1}^n Z_i$, i.e., \mathcal{C} convex-separates \mathcal{D} . □

We begin by defining a restricted type of cascade. This model corresponds more to the idea of joining the cascade components by their respective output function. Thus, we require that the connection between sequential blocks respects convex-coverings.

Definition 35. A *constrained cascade* $D_1 \overset{C_1}{\rightsquigarrow} \dots \overset{C_{n-1}}{\rightsquigarrow} D_n$ w.r.t. covering $C_1 \times \dots \times C_n$ is a dynamics $D_1 \rightsquigarrow \dots \rightsquigarrow D_n$, where $D_i = \langle X_i, U \times C_{[1..(i-1)]}, f_i \rangle$ and D_i is convex-covered by C_i .

We can think of a constrained cascade as a feed-forward cascade with connections $D_1 \overset{g_1}{\rightsquigarrow} \dots \overset{g_{n-1}}{\rightsquigarrow} D_n$ where each g_i is continuous on $U \times C_{[1..i-1]}$.

1967 E.1 Aperiodic Convex-covered Dynamics and Proof of Theorem 9

1968 We define an analogous notion of aperiodicity for convex-covered dynamics. First we extend the
1969 notion of η -convergence to convex-coverings.

1970 **Definition 36.** For a space X , we say a sequence $(x_n)_{n \geq 1} \in X^\omega$ *PC-converges* in X , if eventually
1971 all its terms lie in the same path-connected component of X . ■

1972 This is an identical notion to η -convergence, but we give it a different name, since it applies to
1973 non- η -finite spaces.

1974 **Definition 37.** Call dynamics $D = \langle X, U, f \rangle$ *aperiodic* w.r.t. convex-covering C , if D is convex-
1975 covered by C and if for every sequence $(u_n)_{n \geq 1} \in U^\omega$ PC-convergent in U and $x_0 \in X$, the state
1976 sequence $(D(x_0, u_1 \dots u_n))_{n \geq 1} \in X^\omega \subseteq C^\omega$ is PC-convergent in C .

1977 Note the difference in definition: we require that the state sequence is eventually in the same
1978 component of C , instead of the same component of X !

1979 **Lemma 53.** Let $\mathcal{D} = D_1 \rightsquigarrow \dots \rightsquigarrow D_n$ be a cascade s.t. D_i is aperiodic w.r.t. convex-covering C_i
1980 for $i \in [1..n]$. Then \mathcal{D} is aperiodic w.r.t. convex-covering $\mathcal{C} = C_1 \times \dots \times C_n$.

1981 *Proof.* Analogous to proof of Lemma 49, applied to the cascade $D'_1 \rightsquigarrow \dots \rightsquigarrow D'_n$, where $D'_i =$
1982 $\langle C_i, U \times C_{[1, \dots, i-1]}, f_i \rangle$. ■

1983 **Definition 38.** We call a function $F : \Sigma^+ \rightarrow \Gamma$ *alternating* if, for some $\sigma \in \Sigma$, the sequence
1984 $(F(\sigma^n))_{n \geq 1} \in \Gamma^\omega$ changes value infinitely many times. ■

1985 **Theorem 54.** Let D be a dynamics aperiodic w.r.t. convex-covering C . Let S_C be a shortcut system
1986 constrained by C with dynamics D . Then S_C can not implement any alternating function.

1987 *Proof.* Say $D = \langle X, U, f \rangle$ and $S_C = \langle X, U, f, x_0, C, Y, h \rangle$. Suppose for contradiction that S_C with
1988 encoder $\text{enc} : \Sigma \rightarrow U$ and decoder $\text{dec} : \text{Im } h \rightarrow \Gamma$ implements an alternating function $F : \Sigma^+ \rightarrow \Gamma$.
1989 Let $\sigma \in \Sigma$ be a symbol such that $(F(\sigma^n))_{n \geq 1}$ changes value infinitely many times. Since D
1990 is aperiodic w.r.t. C we have that $(D(x_0, \text{enc}(\sigma^n)))_{n \geq 1} \subseteq X \subseteq C$ is eventually in the same
1991 path-connected component of C . As $\text{dec} \circ h : C \times U \rightarrow \Gamma$ is continuous we thus have that

$$F(\sigma^n) = \text{dec} \circ h(D(x_0, \text{enc}(\sigma^n)), \text{enc}(\sigma))$$

1992 is eventually in the same path-connected component of Γ , i.e. eventually constant. This is a
1993 contradiction. ■

1994 We now introduce an elementary theorem about convex sets in \mathbb{R}^d (or \mathbb{C}^d).

1995 **Theorem 55** (Minkowski's Hyperplane Separation Theorem). Let $A, B \subseteq \mathbb{R}^d$ be two disjoint, non-
1996 empty convex sets. If both are open, then there exists a non-zero vector $v \subseteq \mathbb{R}^d$ and constant $c \in \mathbb{R}$
1997 s.t.

$$\langle a, v \rangle > c \quad \text{and} \quad \langle b, v \rangle < c \quad \forall a \in A, b \in B$$

1998 with $\langle \cdot, \cdot \rangle$ being the dot product.

1999 *Proof.* By Section 2.5.1 of [Boyd and Vandenberghe, 2006 - 2004], we have that there exists a
2000 non-zero vector $v \subseteq \mathbb{R}^d$ and constant $c \in \mathbb{R}$ s.t.

$$\langle a, v \rangle \geq c \quad \text{and} \quad \langle b, v \rangle \leq c \quad \forall a \in A, b \in B$$

2001 Now, these inequalities in fact must be strict. For contradiction suppose that $\langle a, v \rangle = c$ for some
2002 $a \in A$. Since A is open, we have that for some $\epsilon > 0$ $B_{\mathbb{R}^d}(a, \epsilon) \subseteq A$. Thus $a + \epsilon \cdot \frac{v}{\|v\|_2^2} \in A$
2003 ($\|v\|_2 \neq 0$ as v is a non-zero vector). But then $\langle a + \epsilon \cdot \frac{v}{\|v\|_2^2}, v \rangle = a + \epsilon > a$ by linearity of the dot
2004 product. Similarly for B . ■

2005 **Theorem 9.** Let D be an η -finite Linear Recurrent Dynamics, with its state-transition gates having
2006 all non-negative eigenvalues. Let C be a convex-regular covering of D . Then D is aperiodic w.r.t. C .

2007 *Proof.* Let $D = \langle X, U, f \rangle$ be a Linear Recurrent Dynamics, with $X \subseteq \mathbb{R}^d$, convex-covered by C , s.t.
 2008 $A(u)$ has all its eigenvalues being real, for all $u \in U$. Say $f(x, u) = A(u) \cdot x + B(u)$.

2009 Consider a sequence $(u_n)_{n \geq 1} \in U$, state-convergent in U , and $x_0 \in X$. Let $(x_n = D(x_0, u_{1..n})) \subseteq$
 2010 X be the corresponding state sequence. We have some N s.t. for $n \geq N$ all u_n are contained in the
 2011 same component of U , we may pick a representative $r \in U$ of that component.

2012 Write $A = A(r)$, $B = B(r)$. By Lemma 25, we have for $n \geq N$ that

$$x_{n+N} \sim_X x'_n = D(x_N, r^{n-N})$$

2013 Like in proof of Theorem 47, we consider the state sequence in the diagonalized space of A . Write
 2014 $A = P^{-1}JP$ for the Jordan normal form of A . Here J is block diagonal, with say blocks J_1, \dots, J_s ,
 2015 $J_b \in \mathbb{R}^{m_b \times m_b}$ being a Jordan Block with λ_b —eigenvalue of A —on the diagonal, and 1 on the right
 2016 off-diagonal.

2017 Define $y_n = x_{n+1} - x_n$ and $y'_n = P(x'_{n+1} - x'_n)$, then

$$\begin{aligned} y'_{n+1} &= P \cdot (x'_{n+2} - x'_{n+1}) \\ &= P \cdot (Ax'_{n+1} + B - Ax'_n - B) \\ &= PA \cdot (x'_{n+1} - x'_n) = Jy'_n \end{aligned}$$

2018 Thus, unrolling the recurrence we get

$$y'_n = J^n y'_0$$

2019 The i -th component of y'_n , where i is in say the b -th block of J , is

$$[y'_n]_i = \sum_{j=i}^{m_b} \lambda_b^{n+i-j} \binom{n}{j-1} [y'_0]_j$$

2020 The binomial coefficients are polynomial in n . Thus we may write $[y'_n]_i = \sum v_j \cdot n^{b_j} \cdot a_j^n$, where
 2021 $b_j \in \mathbb{Z}_{\geq 0}$ and $a_j = \lambda_b \geq 0$, which is of the form in Lemma 64. Since $y_n = Py'_n$, we have

$$[y_n]_i = \sum_{j=1}^d [P]_{i,j} \cdot [y'_n]_j$$

2022 which again is of the form in Lemma 64.

2023 Now, for contradiction suppose that x'_n is not state-convergent in C . Then, since C has finitely many
 2024 components, there are two distinct components of C , say C_1, C_2 such that x'_n is in both C_1 and in C_2
 2025 infinitely often. Furthermore, since C_1, C_2 are finite unions of open convex sets, there are convex,
 2026 open sets S_1, S_2 which are disjoint, non-empty, and x'_n is in both S_1 and S_2 infinitely often (*).

2027 By Theorem 55, there is a non-zero vector $v \in \mathbb{R}^d$ and constant $c \in \mathbb{R}$ s.t. $\langle s_1, v \rangle > c \quad \forall s_1 \in S_1$
 2028 and $\langle s_2, v \rangle > c \quad \forall s_2 \in S_2$.

2029 Thus, $\langle x'_n, v \rangle > c$ infinitely often, and $\langle x'_n, v \rangle < c$ infinitely often.

2030 We have

$$\langle y_n, v \rangle = \sum_{i=1}^d v_i \cdot [y_n]_i$$

2031 is again in the form from Lemma 64. Thus it is eventually monotone. Therefore eventually $\langle y_n, v \rangle \leq$
 2032 0, in or $\langle y_n, v \rangle \geq 0$. By linearity of the inner product

$$\langle y_n, v \rangle = \langle x_{n+1}, v \rangle - \langle x_n, v \rangle$$

2033 Thus, eventually also $\langle x_n, v \rangle$ is monotone—contradiction with (*). □

2034 E.2 Weakly η -finite Dynamics

2035 In this section we introduce the topological notion of *connectedness*, as well as the necessary results to
 2036 establish the finite state properties of GCSs where the state-space coincides with the convex-covering.

2037 **Definition 39.** A topological space X is called *disconnected*, if there are disjoint non-empty sets
 2038 H, K in X such that $X = H \cup K$. Then X is called *connected* if it is not disconnected.

2039 Connectedness is, as it turns out, a generalization of path-connectedness.

2040 **Fact E.2.1.** (Theorem 27.2, [Willard, 2012]) Every path-connected space is connected.

2041 Similarly to compactness and path-connectedness, connectedness is preserved by continuous map-
 2042 pings and products.

2043 **Fact E.2.2.** (Theorem 26.2, [Willard, 2012]) The continuous image of a connected space is connected.

2044 **Fact E.2.3.** (Theorem 26.10, [Willard, 2012]) A nonempty product space is connected iff each factor
 2045 space is connected.

2046 Similarly to path-connectedness, connectedness induces an equivalence on the space.

2047 **Definition 40.** For $x \in X$, define C_x as the union of connected subspaces of X containing x . We
 2048 call it the *C-component* at x . We write $x \approx_X y$ when $y \in C_x$.

2049 Note, that in [Willard, 2012] C-components are simply referred to as *components*.

2050 **Fact E.2.4.** \approx_X is an equivalence relation, partitioning X into maximal (with respect to inclusion)
 2051 connected subspaces of X . C_x is the equivalence class of \approx_X containing x . See Theorem 26.7 and
 2052 Definition 26.11 of [Willard, 2012] for details.

2053 **Fact E.2.5.** (Theorem 26.12, [Willard, 2012]) The C-components of X are closed in X .

2054 Thus, we think of C-components as a partition of the space that is a coarsening of the path-connected
 2055 components. For an example of a space that has one C-connected component and 2 path-connected
 2056 components, see the *topologist's sine curve* (Example 27.3, [Willard, 2012]).

2057 **Definition 41.** We call a space X *weakly η -finite*, if it has finitely many C-components.

2058 *Example 10.* Any finite alphabet is weakly η -finite, with each symbol being in a separate C-
 2059 component. ■

2060 Our goal now is to show that weakly η -finiteness enjoys the same favourable theoretical properties as
 2061 η -finiteness.

2062 **Lemma 56.** A continuous image of a weakly η -finite space is weakly η -finite.

2063 *Proof.* Let C_1, \dots, C_n be the C-components of X , and let $f : X \rightarrow Y$ be continuous. Each $f(C_i)$
 2064 is connected, and so $\text{Im } f$ is a union of finitely many connected spaces $f(C_1), \dots, f(C_n)$. Thus, the
 2065 equivalence classes of $\approx_{\text{Im } f}$ must be unions of these images. Thus $\approx_{\text{Im } f}$ must have finitely many
 2066 equivalence classes. □

2067 **Lemma 57.** The Cartesian product $X \times Y$ space of weakly η -finite spaces is weakly η -finite. The
 2068 C-components of $X \times Y$ are the products of C-components of X and C-components of Y .

2069 *Proof.* Let C_1, \dots, C_n and E_1, \dots, E_m be the C-components of X, Y respectively. We have $X =$
 2070 $\bigcup_{i=1}^n C_i, Y = \bigcup_{j=1}^m E_j$ and so

$$X \times Y = \left(\bigcup_{i=1}^n C_i \right) \times \left(\bigcup_{j=1}^m E_j \right) = \bigcup_{i=1}^n \bigcup_{j=1}^m C_i \times E_j$$

2071 By Fact E.2.2 each $C_i \times E_j$ is connected. Thus, the C-components of $X \times Y$ are unions of the
 2072 products $C_i \times E_j$. Now, fix $i \in [1..n], j \in [1..m]$. Let Z be the C-component of $X \times Y$ containing
 2073 $C_i \times E_j$. consider the projection map $\pi_X : X \times Y \rightarrow X$. As the projection is continuous, the image,
 2074 $\pi_X(Z)$ is connected in X . Moreover, $C_i \in \pi_X(Z)$. Thus, as C_i is a maximal connected subspace
 2075 of X , we have $C_i = \pi_X(Z)$. Similarly, considering the projection $\pi_Y : X \times Y \rightarrow Y$, we have
 2076 $E_j = \pi_Y(Z)$. Since $C_i \times E_j \subseteq Z$, we therefore must have $C_i \times E_j = Z$. Therefore $X \times Y$ has
 2077 finitely many C-components, and they are the products of C-components of X and C-components of
 2078 Y . □

2079 **Lemma 58.** *Let X be weakly η -finite and Σ be a finite alphabet. Then $f : X \rightarrow \Sigma$ is continuous if*
 2080 *and only if it is constant on the C-components of X .*

2081 *Proof.* (\Rightarrow) Let $f : X \rightarrow \Sigma$ be continuous. Let C be a C-component of X . By Fact E.2.2, $f(C) \subseteq \Sigma$
 2082 is connected, and so $f(C) = \{\sigma\}$ for some $\sigma \in \Sigma$. I.e., f is constant on the C-components of X .

2083 (\Leftarrow) Let $f : X \rightarrow \Sigma$ be constant on the C-components. Let $Y \subseteq \Sigma$ be closed. Then $f^{-1}(Y) \subseteq X$
 2084 must be a union of finitely many C-components, since X is weakly η -finite. By Fact E.2.5, we have
 2085 that each C-component is closed, and therefore also $f^{-1}(Y)$ is closed, as a finite union of closed sets.
 2086 Thus f is continuous. \square

2087 Now, we have all the properties needed to carry out the arguments in Appendix B.3.

2088 **Definition 42.** We call dynamics $D = \langle X, U, f \rangle$ *weakly η -finite* if X and U are weakly η -finite. We
 2089 call a system S *weakly η -finite* if its dynamics are weakly η -finite.

2090 By Lemma 57, we immediatly have that cascades of weakly η -finite dynamics are weakly η -finite.

2091 *Example 11.* η -finite dynamics are weakly η -finite. \blacksquare

2092 **Theorem 59.** *A convex-covering C is weakly η -finite, with its C-components coinciding with its*
 2093 *path-connected components.*

2094 *Proof.* Let C_1, \dots, C_n be path-connected components of C . Each C_i is a union of finitely many
 2095 open (in \mathbb{R}^d) convex sets, and so is also open. Let Z be a C-component of C . Then Z is a union of
 2096 the path-connected components, and so Z is also open. An open, connected subspace of \mathbb{R}^d is path-
 2097 connected, see Corollary 27.6 of [Willard, 2012]. Thus Z must actually be one of the path-connected
 2098 components. \square

2099 **Lemma 60.** *Let $D = \langle X, U, f \rangle$ be a geometrically-constrained system, convex-covered by C , with*
 2100 *$X = C$. Then D is weakly η -finite, and the C-components of X are the path-connected components.*

2101 *Proof.* C has finitely many path-connected components, and so it is weakly η -finite, since path-
 2102 connectedness implies connectedness. Now, each C-component of C is a union of the path-connected
 2103 components, all of which are open in $\Omega = \mathbb{R}^d$. Hence each C-component of C is open in Ω . By
 2104 Corollary 27.6 of [Willard, 2012], C-components of C are therefore path-connected. Thus the
 2105 path-connected components and C-connected components of C coincide. \square

2106 Since a C-component has to be mapped by a continuous function into a single C-component, we
 2107 have that a version of Lemma 24 also holds for weakly η -finite dynamics. For a weakly η -finite
 2108 system $S = \langle X, U, f, x_0, Y, h \rangle$ and weakly η -finite dynamics $D = \langle X, U, f \rangle$, we can thus define the
 2109 analogous canonical automata

$$\begin{aligned} \mathcal{C}_{\text{weakly}}(S) &= \langle X/\approx_X, U/\approx_U, \tilde{f}, [x_0]_{\approx_X}, \text{Im } h/\approx_{\text{Im } h}, \tilde{h} \rangle \\ \mathcal{C}_{\text{weakly}}(D) &= \langle X/\approx_X, U/\approx_U, \tilde{f} \rangle \end{aligned}$$

2110 with $\tilde{f} : ([x]_{\approx_X}, [u]_{\approx_U}) \mapsto [f(x, u)]_{\approx_X}$ and $\tilde{h} : ([x]_{\approx_X}, [u]_{\approx_U}) \mapsto [h(x, u)]_{\approx_{\text{Im } h}}$.

2111 Similarly, replacing path-equivalence \sim with C-component-equivalence \approx in Lemmas 26, 27, 28 and
 2112 Theorem 1, we get that the canonical automata of weakly η -finite systems have the same capability in
 2113 terms of implementing functions.

2114 Likewise, the realization results of Appendix B.4 and Appendix B.5 carry over to the setting of
 2115 weakly η -finiteness. Thus we may apply the structural theorems of Algebraic Automata Theory in
 2116 the case of weakly η -finite dynamics. We defer exploring the properties of weakly η -finite dynamics
 2117 in detail to future work.

2118 E.3 η -finite Systems as GCSs and Proof of Theorem 8

2119 We start by showing that η -finite dynamics that are convex-separated by C can implement exactly the
 2120 same functions in a η -finite system as in a GCS constrained by C .

2121 **Lemma 61.** *Suppose η -finite dynamics D are convex-separated by C . The following are equivalent:*

- 2122 • *There is a system with dynamics D that can implement $F : \Sigma^+ \rightarrow \Gamma$.*
- 2123 • *There is a shortcut system S_C constrained by C with dynamics D that can implement $F : \Sigma^+ \rightarrow \Gamma$.*

2124 *Proof.* (\Rightarrow) Let $S = \langle X, U, f, x_0, Y, h \rangle$ be a system with dynamics D that implements F with some
 2125 encoder $\text{enc} : \Sigma \rightarrow U$ and decoder $\text{dec} : Y \rightarrow \Gamma$.

2126 Let C_1, \dots, C_s be the path-connected components of C . Fix $\gamma \in \Gamma$ and define $h' : C \times U \rightarrow \Gamma$
 2127 as follows: for $i \in 1 \dots s$, if $C_i \cap X = \emptyset$, take $h'(c, u) = \gamma$, where $\gamma \in \Gamma$. If $C_i \cap X \neq \emptyset$,
 2128 take $h'(c, u) = \text{dec} \circ h(x, u)$ for $(c, u) \in C_i \times U$, where $x \in X_i$. This is well-defined: For
 2129 all $x, x' \in C_i \cap X$, since C is a convex-separator of X , we have that x and x' are in the same
 2130 path-connected component of X . Therefore necessarily $\text{dec} \circ h(x, u) = \text{dec} \circ h(x', u)$.

2131 *Want to show:* h' is continuous. Let $((c_n, u_n))_{n \geq 1} \subseteq C \times U$ be a sequence converging to $(c, u) \in$
 2132 $C \times U$. Then $(c_n)_{n \geq 1}$ converges to c in C and $(u_n)_{n \geq 1}$ converges to u in U .

2133 Let C_i be the component that contains c . Since C_i is open, there is some $\epsilon > 0$ s.t. $B_\Omega(c, \epsilon) \subseteq C_i$.
 2134 Since $c_n \rightarrow c$, we must have that eventually (c_n) lies in $B_\Omega(c, \epsilon) \subseteq C_i$. Similarly, let U_j be the
 2135 η -component of U that contains u . Then, by Lemma 20, as $u_n \rightarrow u$, we must have that eventually
 2136 (u_n) lies in U_j . Thus eventually $((c_n, u_n))_{n \geq 1}$ lies in $C_i \times U_j$. By definition of h' , it is constant on
 2137 $C_i \times U_j$. Thus $(h'(u_n, c_n))_{n \geq 1}$ is eventually equal $h'(c, u)$.

2138 Now, define $S_c = \langle X, U, f, x_0, C, Y, h' \rangle$. As $h' : C \times U \rightarrow Y$ is continuous, this is a well-def.
 2139 shortcut system constrained by C . Moreover, since h' constrained to $X \times U$ is equal to $\text{dec} \circ h$, we
 2140 have that S_C with encoder enc and decoder $\text{id} : \Gamma \rightarrow \Gamma$ implement F .

2141 (\Leftarrow) Let $S_c = \langle X, U, f, x_0, C, Y, h \rangle$ be a shortcut constrained by C . Suppose that S_C implements
 2142 F with some encoder $\text{enc} : \Sigma \rightarrow U$ and $\text{dec} : Y \rightarrow \Gamma$. Then taking $h : X \times U \rightarrow Y$ to be the
 2143 restriction of h , we get that the system $S = \langle X, U, f, x_0, Y, h' \rangle$ with encoder enc and decoder dec
 2144 implements F . \square

2145 **Lemma 62.** *Let X be a η -finite space. Then X is convex-separated by some convex-covering C .*

2146 *Proof.* Let X_1, \dots, X_k be the components of X . Take

$$\delta = \min_{1 \leq i < j \leq n} \inf_{x_i \in X_i, x_j \in X_j} d(x_i, x_j)$$

2147 Then we have $\delta > 0$ by Lemma 20. Define

$$C_i^\delta = \{B(x_i, \delta/2) \mid x_i \in X_i\}$$

2148 Then C_i^δ is an open cover of X_i . Since X_i is compact, by definition of compactness there is a
 2149 finite subcover $\bar{C}_i^\delta \subseteq C_i^\delta$ which also covers X_i . Moreover, by definition of δ , this subcover does
 2150 not intersect other components of X . Taking $C_i = \bigcup \bar{C}_i^\delta$ we have that $C = C_1 \cup \dots \cup C_k$ is a
 2151 convex-covering that convex-separates X . \square

2153 **Construction 4.** FLIP-FLOP dynamics can be implemented by a Linear Recurrent Dynamics with
 2154 entries in $[\delta, 1 - \delta]$, for some $\delta > 0$.

2155 Let $\epsilon < 1$. Take $D = \langle X, U, f \rangle$ with $X = X_l \cup X_h$, where $X_l = (-1, 0)$, $X_h = (0, 1)$ and U, f
 2156 such that:

$$f(x, e(\sigma)) = A_\sigma \cdot x + B_\sigma \quad \text{where} \quad \langle A_\sigma, B_\sigma \rangle = \begin{cases} \langle 1 - \epsilon, 0 \rangle & \text{if } \sigma = i \\ \langle \epsilon/4, -1/2 \rangle & \text{if } \sigma = l \\ \langle \epsilon/4, 1/2 \rangle & \text{if } \sigma = h \end{cases}$$

2157 With output function $X_l \mapsto \text{low}$ and $X_h \mapsto \text{high}$, this implements FLIP-FLOP. The set $C = X$ is a
 2158 convex-covering of this dynamics.

2159 Hence, Mamba can implement FLIP-FLOP as a constrained system, and so constrained cascades of
 2160 Mamba blocks can implement any star-free language.

2161 **Corollary 63.** *η -finite dynamics are in particular convex-separated dynamics, and implement the*
2162 *same functions in η -finite systems and in GCSs.*

2163 **Theorem 8.** *SSMs with Mamba parametrisation can recognise all star-free languages as GCSs.*

2164 *Proof.* By Construction 4, there is a Mamba block dynamics D , with a convex-covering state space,
2165 and η -finite input space, that realise FLIP-FLOP as weakly η -finite dynamics. A Mamba block
2166 can also have a convolution, and so there is a Mamba block dynamics E , with a convex-covering
2167 state space, and η -finite input space, that realise R_2 as weakly η -finite dynamics (details omitted.
2168 Also a sLSTM-like η -finite construction is possible, see Appendix G.3). Thus, by weakly η -finite
2169 analogue of Theorem 14, all group-free functions can be realized by feed-forward cascades of D and
2170 E components. Such cascades are actually constrained cascades of Mamba block GCSs, since the
2171 convex-coverings of D and E coincide with their state-spaces. \square

2172 **F Details of The Experiments**

2173 We have created visualizations based on the [Liu et al., 2023] FLIP-FLOP task. The dataset is
2174 available at <https://huggingface.co/datasets/synthseq/flipflop/>. The objective of the
2175 task is to predictively model a sequence of instructions of the form sx , where $s \in \mathbf{w}, \mathbf{r}, \mathbf{i}, x \in 0, 1$. \mathbf{w}
2176 indicates that the next symbol is to be stored, \mathbf{r} indicates that the next symbol should be the retrieved
2177 value and \mathbf{i} indicates no action. The specific task we trained on corresponds to the "clean" prediction
2178 mode, where only prediction following an \mathbf{r} instruction need to be predicted. We note that the aim
2179 of our experiments was to obtain empirical evidence of Mamba having contracting dynamics, and a
2180 comprehensive experimental study is beyond the scope of our paper.

2181 We trained 1-layer Mamba on sequence lengths 32, 64, and 512, observing similar state-collapse
2182 phenomena, as predicted by our results. Additionally [Sarraf et al., 2024] note that in their experiments
2183 Mamba needed more training steps to converge than reported by Liu et al. [2023] for an LSTM. This
2184 is another evidence towards the influence of robustness on stability of training.

2185 The code used to perform the experiments is based on the repository shared in Grazzi et al. [2025],
2186 with some environment modifications to make it work on the 2025-04-09 Google Colab release. The
2187 forked repository is available at https://anonymous.4open.science/r/unlocking_state_tracking-58C4/, with a Google Colab notebook file containing the set-up, simple training loop,
2188 and hidden state visualisation code.
2189

2190 G Additional Proofs and Constructions

2191 G.1 Monotone Sequence Lemma

2192 **Lemma 64.** *Let $d \geq 1$, $a_1, \dots, a_d \geq 0$, $b_1, \dots, b_d \in \mathbb{Z}_{\geq 0}$ and $v_1, \dots, v_d \in \mathbb{R}$. The sequence*

$$x_n = \sum_{i=1}^d v_i \cdot n^{b_i} \cdot a_i^n$$

2193 *is eventually monotone.*

2194 *Proof.* If all $v_i = 0$, then $x_n = 0$ for all n , in particular the sequence is monotone. Otherwise, we
2195 may assume that $v_i \neq 0$ for all i , and that

$$a_i > a_{i+1} \quad \text{or} \quad a_i = a_{i+1} \text{ and } b_i > b_{i+1}$$

2196 If $a_1 = 0$, then again $x_n = 0$, and it is monotone. Otherwise, we can take $d_1 : 1 \leq d_1 \leq d$ such that

$$a_i = a_1 \text{ for } 1 \leq i \leq d_1 \quad \text{and} \quad a_i < a_1 \text{ for } i \geq d_1 + 1$$

2197 We may write

$$x_n = a_1^n \cdot P(n) + \sum_{i=d_1+1}^d v_i \cdot n^{b_i} \cdot a_i^n$$

2198 where $P(n)$ is the polynomial $\sum_{i=1}^{d_1} v_i \cdot n^{b_i}$.

2199 *Case 1:* $a_1 \neq 1$. We have $a_1 > 0$ and

$$\frac{x_n}{a_1^n} = P(n) + \sum_{i=d_1+1}^d v_i \cdot n^{b_i} \cdot (a_i/a_1)^n$$

2200 We have that $(a_i/a_1) \rightarrow 0$ as $n \rightarrow \infty$, since $a_1 > a_i$ for $d_1 + 1 \leq i \leq d$. On the other hand, $P(n)$ is
2201 a non-zero polynomial, since its leading term is $v_1 \cdot n^{b_1}$ and $v_1 \neq 0$, and so $P(n) \rightarrow \pm\infty$ as $n \rightarrow \infty$.
2202 Thus, $x_n \neq 0$ for sufficiently large n . Moreover,

$$\frac{x_{n+1}}{x_n} = \frac{a_1^{n+1}}{a_1^n} \cdot \frac{(n+1)^{b_1}}{n^{b_1}} \cdot \frac{P(n+1)/(n+1)^{b_1} + \sum_{i=d_1+1}^d v_i (n+1)^{b_i-b_1} (a_i/a_1)^{n+1}}{P(n)/n^{b_1} + \sum_{i=d_1+1}^d v_i \cdot n^{b_i-b_1} (a_i/a_1)^n}$$

2203 We have $P(n)/n^{b_1} \rightarrow v_1$ as $n \rightarrow \infty$, since $v_1 \cdot n^{b_1}$ is the leading term of $P(n)$. Also $n^{b_i-b_1}$ grows
2204 at most polynomially, while $(a_i/a_1)^n$ goes to 0 exponentially, since $a_i < a_1$ for $d_1 + 1 \leq i \leq d$.
2205 Therefore $\sum_{i=d_1+1}^d v_i \cdot n^{b_i-b_1} (a_i/a_1)^n \rightarrow 0$ as $n \rightarrow \infty$. Lastly we have $\frac{(n+1)^{b_1}}{n^{b_1}} \rightarrow 1$ as $n \rightarrow \infty$.
2206 All together

$$\lim_{n \rightarrow \infty} \frac{x_{n+1}}{x_n} = a_1 \cdot 1 \cdot \frac{v_1 + 0}{v_1 + 0} = a_1$$

2207 In particular, eventually x_n is positive, or eventually it is negative. There are 4 cases:

- 2208 • If $a_1 \in (0, 1)$ and x_n is positive eventually, then x_n is decreasing eventually.
- 2209 • If $a_1 \in (1, \infty)$ and x_n is positive eventually, then x_n is increasing eventually.
- 2210 • If $a_1 \in (0, 1)$ and x_n is negative eventually, then x_n is increasing eventually.
- 2211 • If $a_1 \in (1, \infty)$ and x_n is negative eventually, then x_n is decreasing eventually.

2212 *Case 2:* $a_1 = 1$. We proceed by induction on b_1 . If $b_1 = 0$, then necessarily $d_1 = 1$, and $P(n) = v_1$.
2213 Then we have by Case 1 that $x_n - P(n) = x_n - v_1$ is eventually monotone, and so also x_n is
2214 eventually monotone.

2215 For the inductive step, consider

$$\begin{aligned} y_n &= x_{n+1} - x_n \\ &= P(n+1) - P(n) + \sum_{i=d_1+1}^d v_i \cdot a_i^n \cdot (a_i(n+1)^{b_i} - n^{b_i}) \end{aligned}$$

2216 We can again write $\sum_{i=d_1+1}^d v_i \cdot a_i^n \cdot (a_i(n+1)^{b_i} - n^{b_i})$ as $\sum_{i=1}^{d'} v'_i \cdot n^{b'_i} \cdot (a'_i)^n$, with $a'_i < a_1 = 1$.
 2217 On the other hand $Q(n) = P(n+1) - P(n)$ is a polynomial with leading coefficient of degree $< b_1$.
 2218 Thus we may apply inductive hypothesis to

$$y_n = Q(n) + \sum_{i=1}^{d'} v'_i \cdot n^{b'_i} \cdot (a'_i)^n$$

2219 to conclude that y_n is eventually monotone. Thus, either $x_{n+1} - x_n = y_n \leq 0$ eventually, or
 2220 $x_{n+1} - x_n = y_n \geq 0$ eventually. Hence x_n is eventually monotone.

2221 □

2222 G.2 Sequential Cascade Construction

2223 The serial cascade can be realised in terms of the feedforward cascade \rightsquigarrow . Consider $i \in 1, 2$ and
 2224 $D_i = \langle X_i, U_i, f_i \rangle$. Define the *repeat* dynamics on X_1 to be the system $R_{X_1} = \langle X_1^2, U \times X_1, r \rangle$,
 2225 with r given by

$$r(\langle x_1, x_2 \rangle, \langle u, x_3 \rangle) = \langle x_2, x_3 \rangle \quad \forall x_1, x_2, x_3 \in X_1, u \in U$$

2226 Thus R_{X_1} can delay the propagation of the state of D_1 by one time step. Also, define the modified
 2227 dynamics $D'_2 = \langle X_2, U \times X_1^3, f'_2 \rangle$, with f'_2 given by

$$f'_2(x_2, \langle u, x_1, x_{1,old}, x_{1,new} \rangle) = f_2(x_2, \langle u, x_{1,old} \rangle)$$

2228 Note that R_{X_1} is equivalent to the usual repeat dynamics over X_1 , $\langle X_1^2, X_1, r_X \rangle$, but with input
 2229 function $(u, x) \mapsto x$.

2230 Now, the feed-forward cascade $D_1 \rightsquigarrow R_{X_1} \rightsquigarrow D'_2$ is well-defined, and has the following transitions:

$$\begin{aligned} f'(\langle x_1, x_{1,old}, x_{1,new}, x_2 \rangle, u) &= \langle x'_1, x'_{1,old}, x'_{1,new}, x'_2 \rangle \quad \text{where} \\ x'_1 &= f_1(x_1, u); \quad x'_{1,old} = x_{1,new}; \quad x'_{1,new} = x'_1; \\ x'_2 &= f'_2(x_2, \langle u, x'_1, x'_{1,old}, x'_{1,new} \rangle) = f_2(x_2, \langle u, x_{1,new} \rangle) \end{aligned}$$

2231 Now, suppose we have system $S = \langle X_1 \times X_2, U, f, (x_{1,0}, x_{2,0}), Y, h \rangle$ with dynamics $D_1 \times$
 2232 D_2 . Then there is a system S' with dynamics $D_1 \rightsquigarrow R_{X_1} \rightsquigarrow D'_2$ which realises S : take
 2233 $S' = \langle X_1^3 \times X_2, U, f', x'_0, Y, h' \rangle$ with $x'_0 = (x_{1,0}, x_{1,0}, x_{1,0}, x_{2,0})$, $h'(\langle x_{1,1}, x_{1,2}, x_{1,3}, x_2 \rangle, u) =$
 2234 $h(\langle x_{1,1}, x_2 \rangle, u)$ and take

$$\begin{aligned} \alpha : X_1 \times X_2 &\rightarrow \mathcal{P}_+(X_1^3 \times X_2) \\ \alpha(x_1, x_2) &\mapsto \{(x_1, x_{old}, x_1, x_2) : x_{old} \in X_1\} \end{aligned}$$

2235 Take $\iota : U \rightarrow U$ and $\zeta : Y \rightarrow Y$ to be the identities. We then have for all $(x_1, x_2) \in X_1 \times X_2$,
 2236 $u \in U$ and $x' \in \alpha((x_1, x_2))$:

$$\begin{aligned} f'(x', \iota(u)) &= f'(\langle x_1, x_{old}, x_1, x_2 \rangle, u) \\ &= \langle x'_1, x_1, x'_1, x'_2 \rangle \in \alpha((x'_1, x'_2)) \end{aligned}$$

2237 where $x'_1 = f_1(x_1, u)$ and $x'_2 = f_2(x_2, \langle u, x_1 \rangle)$, so that $(x'_1, x'_2) = f((x_1, x_2), u)$. Moreover
 2238 $x'_0 \in \alpha(x_0)$.

2239 Finally, we have

$$\zeta \circ h'(x', \iota(u)) = h'(x', u) = h'(\langle x_1, x_{old}, x_1, x_2 \rangle, u) = h(\langle x_1, x_2 \rangle, u) = h(x, u)$$

2240 so that indeed S' is a realisation of S . Note, that we did not need to introduce any new transitions
 2241 on X_1 or X_2 in order to carry out this construction. In particular, if D_1 and D_2 are linear recurrent
 2242 dynamics, then D_1, D'_2 are linear recurrent dynamics. Also R_{X_1} is a Finite Context Dynamics.

2243 G.3 Robust Flip-Flop realisations

2244 Recall the sLSTM parametrisation: the state space of a sLSTM is \mathbb{R}^3 , and the input
 2245 space is \mathbb{R}^d for some $d \geq 1$. The dynamics function of the form $(\langle c, n, h \rangle, u) \mapsto$
 2246 $\langle f_c(\langle c, n, h \rangle, u), f_n(\langle c, n, h \rangle, u), f_h(\langle c, n, h \rangle, u) \rangle$, where

$$\begin{aligned} f_c(\langle c, n, h \rangle, u) &= \psi(l_f(h, u)) \cdot c + \exp(l_i(h, u)) \cdot \varphi(l_z(h, u)) \\ f_n(\langle c, n, h \rangle, u) &= \psi(l_f(h, u)) \cdot n + \exp(l_i(h, u)) \\ f_h(\langle c, n, h \rangle, u) &= \sigma(l_o(h, u)) \cdot \frac{f_c(\langle c, n, h \rangle, x)}{f_n(\langle c, n, h \rangle, x)} \end{aligned}$$

2247 where each $l_s : s \in o, i, z, f$ is a function of the form $w_s^t \cdot u + r_s \cdot h + b_s$, for $w_s \in \mathbb{R}^d, r_s, b_s \in \mathbb{R}$,
 2248 ψ is either \exp or σ , and φ is \tanh .

2249 G.3.1 Strongly robust sLSTM FLIP-FLOP realization

2250 We present a construction for a one layer sLSTM FLIP-FLOP, which is strongly robust. The key idea
 2251 is to only use the h state to implement the dynamics. Then, we can use Theorem 42, and similar
 2252 arguments involving uniform continuity, to extend the construction to be strongly robust in the states
 2253 h, c, n and the input space u . We shall present the arguments in more detail here, to demonstrate
 2254 how robustness can be used to prove properties of systems, in particular how to extend robustness to
 2255 strong robustness.

2256 Let $\psi = \sigma$. Set $w_s = \mathbf{0}$ and $r_s = 0$ for $s = f, i, z$. Set $b_f = -3, b_z = 2, b_i = 0$. Then we have
 2257 $l_f \equiv -3, l_i \equiv 0, l_z \equiv 2$. Thus the updates simplify as

$$\begin{aligned} f_c(\langle c, n, h \rangle, u) &= \sigma(-3) \cdot c + \exp(0) \cdot \tanh(2) = \sigma(-3) \cdot c + \tanh(2) \\ f_n(\langle c, n, h \rangle, u) &= \sigma(-3) \cdot n + \exp(0) = \sigma(-3) \cdot n + 1 \\ f_h(\langle c, n, h \rangle, u) &= \sigma(l_o(h, u)) \cdot \frac{f_c(\langle c, n, h \rangle, x)}{f_n(\langle c, n, h \rangle, x)} = \sigma(l_o(h, u)) \cdot \tanh(2) \in [0, 1] \end{aligned}$$

2258 Finally, take $d = 1$ and $l_o(h, u) = u + 10h - 5$.

2259 For now, let us fix c as $c^* = \frac{\tanh(2)}{1 - \sigma(-3)} \approx 1.01202$ and n as $n^* = \frac{1}{1 - \sigma(-3)} \approx 1.049787$, i.e. the fix
 2260 points of the linear recurrences given by f_c and f_n . Then we have that

$$\sigma(-3) \cdot c^* + \tanh(2) = c^* \quad \text{and} \quad \sigma(-3) \cdot n^* + 1 = n^*$$

2261 Moreover, $\frac{f_c(\langle c^*, n^*, h \rangle, x)}{f_n(\langle c^*, n^*, h \rangle, x)} = \frac{c^*}{n^*} = \tanh 2$, so that the update for h simplifies as

$$f(h, u) := f_h(\langle c^*, n^*, h \rangle, u) = \sigma(u + 10h - 5) \cdot \tanh(2)$$

2262 We can set $U = \{u_{\text{set}}, u_{\text{reset}}, u_{\text{id}}\}$, with $u_{\text{set}} = 8, u_{\text{reset}} = -8$ and $u_{\text{id}} = 0$, and $H_{\text{low}} =$
 2263 $[-0.05, 0.2], H_{\text{high}} = [0.8, 1.05]$

2264 Now, for $h \in [0, 1]$ we have

$$\begin{aligned} f(\langle c, n, h \rangle, u_{\text{set}}) &= \sigma(8 + 10h - 5) \cdot \tanh(2) \\ &\geq \sigma(3) \cdot \tanh(2) \approx 0.9183 \end{aligned}$$

2265 Therefore $f(\langle c, n, h \rangle, u_{\text{set}}) \in [0.85, 1]$. Similarly

$$\begin{aligned} f(\langle c, n, h \rangle, u_{\text{reset}}) &= \sigma(-8 + 10h - 5) \cdot \tanh(2) \\ &\leq \sigma(-3) \cdot \tanh(2) \approx 0.04572 \end{aligned}$$

2266 Therefore $f(\langle c, n, h \rangle, u_{\text{reset}}) \in [0, 0.05]$. Now, for $h \leq 0.2$

$$\sigma(10h - 5) \cdot \leq \sigma(2 - 5) \approx 0.047426 < 0.05$$

2267 and so $f(\langle c, n, h \rangle, u_{\text{id}}) \in [0, 0.05]$. Also for $h \geq 0.8$

$$\sigma(10h - 5) \cdot \tanh(2) > 0.95 \cdot 0.9 = 0.855$$

2268 and so $f(\langle c, n, h \rangle, u_{\text{id}}) \in [0.8, 1]$. Thus we see that the dynamics

$$\left\langle H = H_{\text{low}} \cup H_{\text{high}}, U, f = (h, u) \mapsto f_h(\langle c^*, n^*, h \rangle, u) \right\rangle$$

2269 realise the FLIP-FLOP dynamics, and is η -finite and ϵ -robust, for $\epsilon = 0.05$. Furthermore, we can
 2270 modify the input space U , to make it *strongly* ϵ -robust.

2271 Consider $U' = [0, 10]$. $H \times U'$ is compact, and f is continuous on $H \times U'$, so by Theorem 41 it is
 2272 uniformly continuous on $H \times U'$. In particular, for $\epsilon' = \epsilon/2$, there exists $\delta > 0$ such that

$$\| (h, u) - (h', u') \| \leq \delta \implies \| f(h, u) - f(h', u') \| \leq \epsilon'$$

2273 for all $(x, u), (x', u') \in X' \times U'$. Thus, we may take $\delta' = \min(\delta, 1)$ and $U'' = [u_{\text{set}} \pm \delta'] \cup [u_{\text{reset}} \pm$
 2274 $\delta'] \cup [u_{\text{id}} \pm \delta']$. Now, consider $h \in H, u \in U''$ and $h' \in \mathbb{R}$ such that $\|h' - f(h, u)\| \leq \epsilon'$. We have
 2275 $\|u - u'\| \leq \delta'$ for some $u' \in \{u_{\text{set}}, u_{\text{reset}}, u_{\text{id}}\}$, and so

$$\|f(h, u) - f_h(h, u')\| \leq \epsilon'$$

2276 All together

$$\begin{aligned} \epsilon &= \epsilon' + \epsilon' \geq \|h' - f(h, u)\| + \|f(h, u) - f(h, u')\| \\ &\geq \|(h' - f(h, u)) + (f(h, u) - f(h, u'))\| \\ &= \|h' - f(h, u')\| \end{aligned}$$

2277 Since $(h, u') \in H \times U$ and $\langle H, U, f \rangle$ is ϵ -robust, we get that $h' \in H$. Hence f also gives a well
 2278 defined dynamics function $H \times U'' \rightarrow H$, which moreover is ϵ' -robust. Thus, we have $\langle H, U'', f \rangle$
 2279 is η -finite and strongly $\min(\epsilon', \delta')$ -robust. It also realizes FLIP-FLOP, since the input components
 2280 induce the same η -transitions as $\{u_{\text{set}}, u_{\text{reset}}, u_{\text{id}}\}$ by path-connectedness.

2281 Finally, we extend the dynamics to c and n . We can see f as parametrized by $\theta \in [c^* \pm 0.5], \rho \in$
 2282 $[n^* \pm 0.5]$, given by

$$f_{\theta, \rho} = \sigma(u + 10h - 5) \cdot \frac{\theta}{\rho}$$

2283 So, $f = f_{c^*, n^*}$. We see that $f_{\theta, \rho}$ is continuous in θ and ρ , and $[c^* \pm 0.5] \times [n^* \pm 0.5]$ is compact.
 2284 Thus by Theorem 42, there is some $\gamma > 0$ such that $f_{\theta, \rho}$ induces the same function $\overline{H} \times \overline{U''} \rightarrow \overline{H}$
 2285 as f_{c^*, n^*} . Also, similarly to how we extended U to U'' , we can choose γ such that the resulting
 2286 dynamics are always $\epsilon/4$ -robust

2287 Lets take $X = H \times C \times N$ where $C = [c^* \pm \gamma]$ and $N = [n^* \pm \gamma]$. We have that the sLSTM
 2288 dynamics gives a well-defined, *robust* dynamics function $X \times U \rightarrow X$: we already have that
 2289 the restriction of the dynamics to the h component is robust. For the c and n components, since
 2290 $\sigma(-3) < 1$, the state updates given by f_c and f_n (which are independent of u) are contractions
 2291 towards c^* and n^* respectively, with rate $\sigma(-3)$. Thus f_c sends $C = [c^* \pm \gamma]$ to $[c^* \pm \gamma \cdot \sigma(-3)]$
 2292 and f_n sends $N = [n^* \pm \gamma]$ to $[n^* \pm \gamma \cdot \sigma(-3)]$. All together, the sLSTM dynamics are strongly
 2293 $\min(\epsilon/4, \delta', \gamma(1 - \sigma(3)))$ -robust, and realize FLIP-FLOP.

2294 G.3.2 Strongly robust sLSTM repeat dynamics

2295 To realize any repeat semiautomata, as defined in Appendix G.2, it is sufficient to realize the two
 2296 state repeat semiautomaton $R_2 = \langle \{0, 1\}^2, \{0, 1\}, r \rangle$, with $r(\langle x_{\text{old}}, x_{\text{new}} \rangle, x) = \langle x_{\text{new}}, x \rangle$.

2297 Here, the construction is extremely similar to the FLIP-FLOP one. We first show a robust dynamics
 2298 on just the h cell, using $f(h, u) = \sigma(u + 10h - 5) \cdot \tanh(2)$ which realize R_2 . Then we can use the
 2299 same argument as before to extend it to strongly robust dynamics on all 3 cells.

2300 We can use the h cell to represent x_{new} , by simply reusing the previous strongly robust construction
 2301 for setting the high and low state, with dynamics function $f(h, u) = f_h(\langle c^*, n^*, h \rangle, u)$, state space
 2302 H and input space $[u_{\text{set}} \pm \delta'] \cup [u]$. We then have that for some $\gamma > 0$ for all $c \in [c^* \pm \gamma]$ and
 2303 $n \in [n^* \pm \gamma]$ the dynamics function $f_h(\langle c, n, h \rangle, u)$ still performs

2304 Define $X_{00} = [-0.01, 0.015]$, $X_{01} = [0.02, 0.05]$, $X_{10} = [0.95, 0.98]$, $X_{11} = [0.985, 1, 01]$ and
 2305 $u_0 = -8.1, u_1 = 8.1$. Note that $X = X_{00} \cup X_{01} \cup X_{10} \cup X_{11}$ has 4 η -components. Also, define

2306 $X_0 = X_{01} \cup X_{10}$ and $X_1 = X_{10} \cup X_{11}$. In our construction X_{ab} will correspond to the state of R_2
 2307 after the last two inputs were ab , $a, b \in \{0, 1\}$.

2308 We have

$$\begin{aligned} f(0.95, u_1) &= \sigma(8.1 + 9.5 - 5) \approx 0.999997 \\ f(1.01, u_1) &= \sigma(8.1 + 10.1 - 5) \approx 0.999998. \end{aligned}$$

2309 As σ is increasing, we therefore have $f(X_1, u_1) \subseteq [0.99999, 1] \subset X_{11}$. Similarly, we have

$$\begin{aligned} f(-0.01, u_1) &= \sigma(8.1 - 0.1 - 5) \approx 0.9526 \\ f(0.05, u_1) &= \sigma(8.1 + 0.5 - 5) \approx 0.9734. \end{aligned}$$

2310 Therefore $f(X_0, u_1) \subseteq [0.952, 0.974] \subset X_{10}$. Similarly for u_0 , we have

$$\begin{aligned} f(0.95, u_0) &= \sigma(-8.1 + 9.5 - 5) \approx 0.0265 \\ f(1.01, u_0) &= \sigma(-8.1 + 10.1 - 5) \approx 0.0474. \end{aligned}$$

2311 Therefore $f(X_1, u_0) \subseteq [0.025, 0.0475] \subset X_{01}$. Similarly

$$\begin{aligned} f(-0.01, u_0) &= \sigma(-8.1 - 0.1 - 5) \approx 0.000001 \\ f(0.05, u_0) &= \sigma(-8.1 + 0.5 - 5) \approx 0.000003. \end{aligned}$$

2312 Therefore $f(X_0, u_0) \subseteq [0, 0.000004] \subset X_{00}$. Thus $\langle X, \{u_0, u_1\}, f \rangle$ are well-defined dynamics and
 2313 the 4 η -components correspond to 4 possible values for the last 2 inputs. Hence clearly they can
 2314 realize R_2 . Moreover, the dynamics are strongly robust. The remainder of the argument is the same
 2315 as for the FLIP-FLOP construction.

2316 **G.3.3 Strongly robust Elman-RNN FLIP-FLOP construction**

2317 The following is a modification of a construction in [Knorozova and Ronca, 2024a]. Consider the
 2318 dynamics function

$$f(x, u) = \tanh(2 \cdot x + u)$$

2319 for $x, u \in \mathbb{R}$. We have that for all x, u , $f(x, u) \in [-1, 1]$. Define $X_{\text{low}} = [-1.1, \tanh(-1)]$, $X_{\text{high}} =$
 2320 $[\tanh(1), 1.1]$. Note that $\tanh(1) \approx 0.76159$, $\tanh(-1) \approx -0.76159$

2321 We have

$$\begin{aligned} f(-1.1, 4) &= \tanh(-2.2 + 4) \approx 0.9468 \\ f(1.1, 4) &= \tanh(2.2 + 4) \approx 0.999992 \end{aligned}$$

2322 As \tanh is increasing, we have $f([-1.1, 1.1], 4) \subseteq [0.9467, 0.999993] \subset X_{\text{high}}$. Similarly,
 2323 $f([-1.1, 1.1], -4) \subseteq [-0.999993, -0.9467] \subset X_{\text{low}}$. Moreover

$$\begin{aligned} f(\tanh(1), 0) &= \tanh(2 \cdot \tanh(1)) \approx 0.909 \\ f(1.1, 0) &= \tanh(2 \cdot 1.1) \approx 0.9757 \end{aligned}$$

2324 Thus, $f(X_{\text{high}}, 0) \subseteq [0.908, 0.9757] \subset X_{\text{high}}$. Similarly $f(X_{\text{low}}, 0) \subseteq [-0.9757, -0.909] \subset X_{\text{low}}$.
 2325 Thus we see that, taking $X = X_{\text{low}} \cup X_{\text{high}}$, $u_{\text{set}} = 4$, $u_{\text{reset}} = -4$, $u_{\text{id}} = 0$, the η -finite dynamics
 2326 $\langle X, \{u_{\text{set}}, u_{\text{reset}}, u_{\text{id}}\}, f \rangle$ are well-defined, and realize FLIP-FLOP. Also clearly they are robust.
 2327 Now, by the same argument as for the sLTSM FLIP-FLOP realisation, we can extend the input space,
 2328 using Theorem 16 and Theorem 42, to obtain a strongly robust construction.

2329 **Extended References**

- 2330 Jeffrey L. Elman. Finding structure in time. *Cogn. Sci.*, 14(2):179–211, 1990.
- 2331 Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):
2332 1735–1780, 1997.
- 2333 Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger
2334 Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for
2335 statistical machine translation. In *EMNLP*, 2014a.
- 2336 Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova,
2337 Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended
2338 long short-term memory. In *Advances in Neural Information Processing Systems 38: Annual
2339 Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- 2340 Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured
2341 state spaces. In *Proceedings of the Tenth International Conference on Learning Representations
2342 (ICLR)*, 2022.
- 2343 Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*,
2344 abs/2312.00752, 2023.
- 2345 Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: recurrent memory
2346 with optimal polynomial projections. In *Advances in Neural Information Processing Systems 33:
2347 Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- 2348 Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers
2349 with the delta rule over sequence length. In *Advances in Neural Information Processing Systems
2350 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver,
2351 BC, Canada, December 10 - 15, 2024*, 2024.
- 2352 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz
2353 Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- 2354 Nadezda Alexandrovna Knorozova and Alessandro Ronca. On the expressivity of recurrent neural
2355 cascades. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*,
2356 pages 10589–10596, 2024a.
- 2357 Nadezda Alexandrovna Knorozova and Alessandro Ronca. On the expressivity of recurrent neural
2358 cascades with identity. In *Proceedings of the 21st International Conference on Principles of
2359 Knowledge Representation and Reasoning (KR)*, 2024b.
- 2360 Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision
2361 RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for
2362 Computational Linguistics (ACL)*, pages 740–745, 2018.
- 2363 William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. A
2364 formal hierarchy of RNN architectures. In *ACL*, 2020.
- 2365 Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages
2366 can transformers express? A survey. *Trans. Assoc. Comput. Linguistics*, 12:543–561, 2024.
- 2367 William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought.
2368 In *The Twelfth International Conference on Learning Representations*, 2024.
- 2369 Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Trans. Assoc.
2370 Comput. Linguist.*, 8, 2020.
- 2371 Satwik Bhattamishra, Michael Hahn, Phil Blunsom, and Varun Kanade. Separations in the repre-
2372 sentational capabilities of transformers and recurrent architectures. In *The Thirty-eighth Annual
2373 Conference on Neural Information Processing Systems*, 2024.

- 2374 Yash Sarrof, Yana Veitsman, and Michael Hahn. The expressive capacity of State Space Models: A
2375 formal language perspective. In *Proceedings of the Thirty-Eighth Annual Conference on Neural*
2376 *Information Processing Systems (NeurIPS)*, 2024.
- 2377 Riccardo Grazi, Julien Siems, Arber Zela, Jörg K. H. Franke, Frank Hutter, and Massimiliano Pontil.
2378 Unlocking state-tracking in linear RNNs through negative eigenvalues. In *Proceedings of the*
2379 *Thirteenth International Conference on Learning Representations (ICLR)*, 2025.
- 2380 Juris Hartmanis and R. E. Stearns. *Algebraic structure theory of sequential machines*. Prentice-Hall
2381 international series in applied mathematics. Prentice-Hall, Englewood Cliffs, N.J, 1966.
- 2382 Abraham Ginzburg. *Algebraic Theory of Automata*. Academic Press, 1968.
- 2383 Michael A. Arbib. *Theories of abstract automata*. Prentice-Hall series in automatic computation.
2384 Prentice-Hall, Englewood Cliffs, N.J, 1969.
- 2385 Pál Dömösi and Chrystopher L Nehaniv. *Algebraic theory of automata networks: An introduction*.
2386 SIAM, 2005.
- 2387 Kenneth Krohn and John Rhodes. Algebraic theory of machines. I. Prime decomposition theorem for
2388 finite semigroups and machines. *Trans. Am. Math. Soc.*, 116, 1965.
- 2389 Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In
2390 *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of
2391 *Proceedings of Machine Learning Research*, pages 5819–5827. PMLR, 2018.
- 2392 Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers
2393 and their ability to learn sparse boolean functions. In *Proceedings of the 61st Annual Meeting of*
2394 *the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 5767–5791,
2395 2023.
- 2396 Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Exposing attention
2397 glitches with flip-flop language modeling. In *Thirty-seventh Conference on Neural Information*
2398 *Processing Systems*, 2023. URL <https://openreview.net/forum?id=VzmpXQAn6E>.
- 2399 Gregoire Deletang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt,
2400 Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A Ortega. Neural networks and
2401 the chomsky hierarchy. In *The Eleventh International Conference on Learning Representations*,
2402 2023. URL <https://openreview.net/forum?id=WbxHAzkeQcn>.
- 2403 Kyunghyun Cho, Bart Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Y. Bengio.
2404 Learning phrase representations using rnn encoder-decoder for statistical machine translation. 06
2405 2014b. doi: 10.3115/v1/D14-1179.
- 2406 Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Y. Bengio. Empirical evaluation of gated
2407 recurrent neural networks on sequence modeling. 12 2014.
- 2408 William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in State-Space Models.
2409 In *Proceedings of the Forty-first International Conference on Machine Learning (ICML)*, 2024.
- 2410 Stephen. Willard. *General Topology*. Dover Books on Mathematics. Dover Publications, Newburyport,
2411 1st edition, 2012.
- 2412 Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and*
2413 *Control*, 8(2):190–194, 1965.
- 2414 Nadezda Alexandrovna Knorozova and Alessandro Ronca. On the expressivity of recurrent neural
2415 cascades. *CoRR*, abs/2312.09048, 2023.
- 2416 George Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals*
2417 *Syst.*, 5(4):455, 1992.
- 2418 Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are
2419 universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- 2420 Stephen P. Boyd and Lieven. Vandenberghe. *Convex optimization*. Cambridge University Press,
2421 Cambridge, 2006 - 2004.