Toward a Logic for Neural Networks

Hornischer, L.

Publication date
2019
Document Version
Author accepted manuscript
Published in
The Logica Yearbook 2018

Toward a Logic for Neural Networks

LEVIN HORNISCHER[1]

Abstract: Neural networks and related computing systems suffer from the notorious *black box problem*: despite their success, we lack a general framework or language to reason about the behavior of these systems. We need a logic with a mathematical semantics for this. In this paper, we sketch a first such logic: a mathematical structure with a logic that describes the behavior of possibly non-deterministic, discrete dynamical systems (which include neural networks).
The mathematical structure is based on *domain theory*. Domains solved the 'black box problem' of (classical) computers by providing a denotational semantics for computer programs. Can it analogously be used for neural networks? We show that, under precise conditions, the possible behaviors of a system form a domain—relating domain theoretic concepts to properties of the system.
This mathematical structure can interpret the well-behaved logic HYPE which thus can be used to reason about both the long-term behavior and the history of the system.

Keywords: Neural networks, dynamical systems, black box problem, domain theory, logic, HYPE, unbounded nondeterminism

1 Introduction

Among computing systems, neural networks gained great prominence in the recent years due to their success in image recognition, natural language processing, and, more generally, in learning from great amounts of data. However, despite their success, they suffer from the notorious *black box problem*: we don't fully understand how they do what they do. Making neural networks and modern artificial intelligence more transparent has been dubbed 'explainable AI'. While explainable AI recently made progress in understanding *specific* applications of neural networks, we still lack a general framework or language to reason about the behavior of these systems.

[1]For inspiring discussions, I'm grateful to Samson Abramsky, Franz Berto, Jon Michael Dunn, Michiel van Lambalgen, Hannes Leitgeb, and the audience at Logica 2018.

That is, we need a logic together with a mathematical semantics that is able to describe the behavior of these computing systems. In this paper, we want to sketch a first such logic.

In section 2, we define the class of computing systems that we'll consider. In section 3, we define the set of possible behaviors of such a system and a partial order on this set. In section 4, we present precise conditions on the system ensuring the behavior poset to be an algebraic domain in the sense of domain theory. In section 5, we then show how this domain can be transformed in a structure that interprets the recently developed logic HYPE. We conclude in section 6.

The aim of the paper is to convey the main idea of this 'logic plus semantics' for neural nets. Thus, we (necessarily have to) focus on the intuition behind it, leaving the details for elsewhere (Hornischer, 2018). We indicate possible further development of this idea and potentially fruitful connections to other fields (e.g., automata theory, coalgebra, homology, or general relativity).

2 Dynamical systems and trajectories

We define the notion of a possibly non-deterministic discrete dynamical system that we'll be working with, and we show that both neural network computation and learning are such systems.

In the most general sense, a *possibly non-deterministic discrete dynamical system* is a pair (S, f) where S is a non-empty set, called the *state space*, and $f : S \to \mathcal{P}(S) \setminus \{\emptyset\}$ is a function mapping each state s to a non-empty set of states, called the *successor states* of s. Intuitively, S is the set of states that the system can be in, and f is the local rule describing which states the system might evolve to given the current state. (Thus, these systems can be seen as *transition systems* known in computer science.) If there always is exactly one successor state, the system is called *deterministic*.[2] A (finite or infinite) *trajectory* is a (finite or infinite) sequence $\langle s_0, s_1, \ldots \rangle$ of states in S such that for each $i \geq 0$, s_{i+1} is a successor state of s_i. We denote the empty trajectory by $\perp$. Intuitively, a trajectory is a possible evolution of the system (a path through the state space). For brevity we call a possibly non-deterministic discrete dynamical systems a *(non-) deterministic system*.

[2] If we want to be even more general, we could allow the local update rule to change over time, that is, define a system as $(S, \{f_n\}_{n \in \mathbb{N}})$ where each f_n maps states to non-empty sets of states. However, for convenience we stick to the simpler definition.

This definition is very general. Depending on the kind of application, dynamical systems are usually assumed to have additional structure: that the state space carries a topology and the dynamics is continuous or that the state carries a measure and the dynamics is measure preserving. However, we need to work with the general definition if we want to capture all the different kinds of neural networks and related computing systems. It is surprising that, as we'll see, even though this general notion of a dynamical system has very little structure, a lot of structure will emerge when considering equivalence classes of trajectories.

Let's see that computation in neural networks can be regarded as (non-) deterministic system. Roughly, a *neural network* is a collection of neurons that are linked via synapses. Each neuron can have an activation that describes whether and how the neuron is firing. Each synapse connecting neuron i to neuron j carries a weight describing how much activation can pass from i to j. A state of the network describes the activation of all the neurons at a given time. The activation propagates through the synapses and determines the successor state: if neurons $i_1, \ldots, i_k$ are connected by synapses to neuron j, then the activation of j at time $n+1$ is computed from the activation of $i_1, \ldots, i_k$ at time n. Thus, a neural network is a deterministic system (S, f) where S is the set of all possible states of activation of the neurons and f describes the propagation of activation. Of course, much more can be said, e.g., about the different kinds of neural nets (feed forward, recurrent, etc.) or about learning (changing of weights).[3] But, again to remain as general as needed, we don't need further details for now. In fact, many (if not most) computing systems can be seen as (non-) deterministic systems (Turing machines, automata, etc.).

On this perspective, neural networks are deterministic systems—so why did we include non-deterministic systems? For two reasons. First, since usually continuous (rather than binary) activation values are allowed, the state space is infinite (of size continuum). Thus, to understand the behavior of the neural net, it is useful to partition the state space into finitely many cells: e.g., clustering observationally equivalent states together. The dynamics induced on these cells may be non-deterministic: a cell can contain two states whose successor states are in two different cells. Second, instead of understanding the computation of a trained neural network, we may also wish to describe the training of a neural network. That is, instead of sequences

[3]Details can be found in the many textbooks on neural networks. Albeit older, Rojas (1996) has a good presentation of the mathematical basics.

of neuron-activation given by the propagation rule we consider sequences of network-weights given by a learning algorithm—and learning might be non-deterministic.

3 Behavioral equivalence and order

If we want to analyze the behavior of systems, we should have a notion of what a possible behavior of a system is. A trajectory of a system is an instance of a possible behavior, but two trajectories might exhibit the same behavior. Thus, a possible behavior is a trajectory modulo behavioral equivalence. So we need to define when two trajectories t and t' are behaviorally equivalent. Intuitively, t and t' should agree on the computationally important information. That is, in the case of neural networks,

(i) t and t' yield the same result: there is a (minimal) point where the two trajectories meet and continue the same path.

(ii) t and t' gather the same information along the way: they described the same cycles and visited the same stable states before reaching the meeting point.

The second point is motivated by the rule of thumb from dynamical systems theory that in the stable states and cycles of a system lies the information computed by the system. For example, a Hopfield neural network retrieves memorized pictures in its stable states, and the limit cycle in a Hodgkin-Huxley model of a neuron describes the spiking pattern of the neuron.

Let's formalize this intuitive idea of behavioral equivalence. Concerning (i), call a pair $(i, j) \in \mathbb{N}$ a *locally minimal coincidence pair* of two trajectories t and t' if $t(i) = t'(j)$ and, whenever defined,

$$t(i-1) \neq t'(j-1) \qquad t(i-1) \neq t'(j) \qquad t(i) \neq t'(j-1).^4$$

Say t and t' have the *same tail* starting in (i, j) (denoted $t(i) \ldots = t'(j) \ldots$), if for all n, $t(i+n)$ is defined iff $t'(j+n)$ is defined and in both cases they are equal. Then (i) says that there is a locally minimal coincidence pair (i, j) starting in which t and t' have the same tail.

[4]For a globally minimal coincidence pair we don't quantify only about the direct precursors but over all precursors. This doesn't make sense for non-deterministic systems, and for deterministic systems one can show that the local and the global formulation of the resulting notion of behavioral equivalence are equivalent.

Concerning (ii), we need to additionally formalize that $t(0)\dots t(i)$ and $t'(0)\dots t'(j)$ are *cycle equivalent* and *stable state equivalent*. We do this first explicitly and then axiomatically.

If s' is a successor state of s, call the transition from s to s' a *cycle edge* if there is a trajectory from s' back to s. Say two finite trajectories t and t' are *cycle equivalent* if the cycle edges occurring on t are precisely the cycle edges occurring on t', and the respective number of occurrences are the same, too. The *occurrence profile* of a stable state s (i.e., s is a successor of itself) on a finite trajectory t is a finite (possibly empty) sequence of non-negative integers $\langle n_1,\dots,n_k\rangle$ where there are k-many blocks of uninterrupted repetitions of s on t such that $n_i \geq 1$ is the number of repetitions of s in the i-th block.[5] Say two finite trajectories t and t' are *stable state equivalent* if the stable states occurring on t are precisely the stable states occurring on t', and the stable state have the same occurrence profile on t and t', respectively.

In fact, we'll rely only on a few of the properties of these definitions. Thus, for transparency and to allow our results to be applicable to a range of possible explicit definitions, we introduce cycle- and stable state equivalence axiomatically. We call equivalence relations $\approx_c$ and $\approx_s$ on finite trajectories *cycle equivalence* and *stable state equivalence*, respectively, if they satisfy the following axioms. (For readability we omit the qualifier 'and the trajectory is possible in the system'.)

1. Both $ta \approx_c t'a$ iff $tat'' \approx_c t'at''$, and $ta \approx_s t'a$ iff $tat'' \approx_s t'at''$.
2. If $a \neq b$, then $taa \not\approx_s t'ba$, $ta \not\approx_s taat'$, and $ta \not\approx_c tabt'a$.
3. If $aa_1\dots a_na$ occurs on t but no a_i on t', then $t \not\approx_c t'$.
4. If there is no stable state in t and t', then $ta \approx_s t'a$.
5. If there is no limit cycles in system, then $\approx_c$ is trivial.
6. If $t \approx_c t'$, then for every occurrence of $aa_1\dots a_na$ on t, there is an occurrence of some a_i on t'.
7. If $t \approx_s t'$, then for every occurrence of stable state a on t, there is an occurrence of a on t'.

[5] For example, the occurrence profile of stable state a on trajectory $aabab$ is $\langle 2,1\rangle$.

The axioms are satisfied by the brute force definition. More elegant definitions might be found by considering homology on (the simplex graph of) the system considered as graph.[6]

For the remainder of the paper, we fix two relations $\approx_c$ and $\approx_s$ satisfying the above axioms. We summarize.

Definition 1 (Behavioral equivalence) *Given a (non-) deterministic system, two trajectories t and t' are* behaviorally equivalent *($t \equiv t'$) if $t = \bot = t'$ or there is a locally minimal coincidence pair (i, j) of (t, t') such that $t(i) \ldots = t'(j) \ldots$, $t(0) \ldots t(i) \approx_c t'(0) \ldots t'(j)$, and $t(0) \ldots t(i) \approx_s t'(0) \ldots t'(j)$. We write $\mathbb{T} := \{[t] : t \text{ is a trajectory}\}$ for the set of equivalence classes of trajectories.*

Thus, $\mathbb{T}$ is the set of possible behaviors of the given (non-) deterministic system. We next observe that there is a natural order on $\mathbb{T}$: $[t] \leq [t']$ if any trajectory equivalent to t can be extended to one equivalent to t', that is,

$$[t] \leq [t'] \text{ iff } \forall t_0 \in [t]\ \exists t_1 \in [t'] : t \preceq t'$$

(iff, intuitively, behavior $[t]$ can be extended by the system into behavior $[t']$). It follows from the axioms on $\approx_c$ and $\approx_s$ that $(\mathbb{T}, \leq)$ is a partial order (it trivially is a preorder).

In the next sections, we'll investigate the order of behaviors $(\mathbb{T}, \leq)$.

4 Long-term behavior and history as a domain

We'll provide precise conditions on the system such that the order of behaviors $(\mathbb{T}, \leq)$, capturing the long-term behavior of the system, and its dual, capturing the history of the system, are algebraic domains.

We start with a very brief recap of domain-theory. Domain theory was originated by Dana Scott and others in the late 1960's (Scott, 1970). Since then, it developed into a rich logico-mathematical framework to understand computation (Abramsky & Jung, 1994). Roughly, domain theory studies particular partial orders that can be regarded as "information orderings": the elements represent pieces of information or partially known objects. As one moves up in the order, one gains more information (about the objects). The important formal concepts are the following. Let $(P, \leq)$ be a partial order.

[6] Generally speaking, a homological perspective on neural networks is fruitful: see e.g. Reimann et al. (2017).

A subset $D \subseteq P$ is *directed* if $D \neq \emptyset$ and any two elements of D have an upper bound in D. Moreover, P is a *directed-complete* partial order (dcpo) if every directed subset of P has a least upper bound. Thus, intuitively, chains of ever increasing information converge to a limit. If P is a dcpo, we say that x is *way below* y ($x \ll y$) if if any directed set whose least upper bound is above y already contains an element above x. An element $x \in P$ is *compact* if $x \ll x$, so x is, in a sense, finitary. An *algebraic domain* is a dcpo where, for every element x, the compact elements below x form a directed set whose least upper bound is x. Algebraic domains are particularly well behaved since their important properties are already determined by the compact elements.

Most importantly, domain theory developed denotational semantics for computer programs (of various programming languages). So it provides mathematical objects explaining what a program is computing. Can it also be used to analogously analyze the behavior of computing systems? Yes, as we'll now show: for appropriate systems, the set of behaviors is an algebraic domain.

Theorem 1 (Upward trajectory domain) *Let $(\mathbb{T}, \leq)$ be the set of behaviors of a (non-) deterministic system. $(\mathbb{T}, \leq)$ is an algebraic domain if the state space doesn't contain the subsystems in figure 1 (arrows indicate paths and not necessarily direct successors).*[7] *We then refer to $(\mathbb{T}, \leq)$ as* upward trajectory domain. *(The converse holds under some more assumptions on $\approx_c$ and $\approx_s$.)*

Intuitively, what the forbidden subsystems have in common is that there are infinitely many choices: In the left-most one, there are infinitely many ways to go from a_1 to b. In the middle one, there are infinitely many ways of changing from the a-path to the b-path. In the right-most system, there are infinitely many choices between doing the b-cycle or the c-cycle. Thus,

[7] The following side conditions are imposed: In the left-most subsystem, (i) all indicated states are distinct. (ii) There is no state that is both reachable from all a_n's and reaches some a_n. And (iii) there is an infinite trajectory t through all a_n's and a finite trajectory t' ending in b such that $[t']$ is an upper bound of $[t \restriction 1] \leq [t \restriction 2] \leq \ldots$.

In the middle subsystem, (i) none of the a_n's is identical to another one, but arbitrarily many b_n's might be identical and some b_n's might be equal to some a_n. (ii) If t is a trajectory through all a_n's, then $[t \restriction 1] \leq [t \restriction 2] \leq \ldots$ doesn't have a finite upper bound. And (iii) there is an infinite trajectory t through all a_n's and an infinite trajectory t' through all b_n's such that $[t']$ is an upper bound of $[t \restriction 1] \leq [t \restriction 2] \leq \ldots$ and $t' \not\equiv t$.

In the right-most subsystem, either $a \neq b$ or $a \neq c$ (or both).

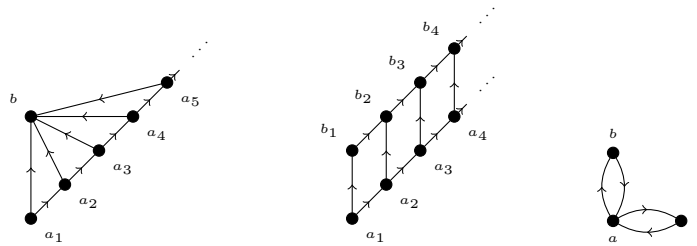

Figure 1: Forbidden subsystems (upward)

if this 'infinite non-determinism'[8] is excluded, the set of behaviors is an algebraic domain.

The upward trajectory domain describes the long-term behavior of the system: the further one moves up in the order starting at a finite $[t]$, the further the continuation of behavior of $[t]$ one considers. Since it is a domain, it can then be analyzed with the tools of domain theory (Abramsky & Jung, 1994). For example, all finite behaviors are compact, whence algebraicity entails that the global (i.e., infinite) behavior of the system can be completely described from observing finite behavior.[9] Moreover, domain theoretic concepts can be related to properties of the system. The theorem indicates such a connection between algebraicity and 'finite nondeterminisim.' One could conjecture other connections: $(\mathbb{T}, \leq)$ is a Scott domain or a bifinite domain if additionally the left-most and, respectively, the middle subsystem of figure 2 is excluded.

While the upward trajectory domain $(\mathbb{T}, \leq)$ describes the structure of the long-term behavior of the system – that is, the *effects* of certain states –, we're also often interested in the *causes* of a state. Then we need to look at the history of the system: the order-dual $\leq^{\mathsf{op}}$ of $\leq$. We can again find conditions on the system guaranteeing this partial order to be an algebraic domain, too.

Theorem 2 (Downward trajectory domain) *For a (non-) deterministic sys-*

[8]This is not the exactly the same as the well-known term 'unbounded non-determinism' which usually means that some state contains infinitely many successor states (as, e.g., in the diamond of figure 2).

[9]Formally, an algebraic domain is isomorphic to the ideal completion of its compact elements.

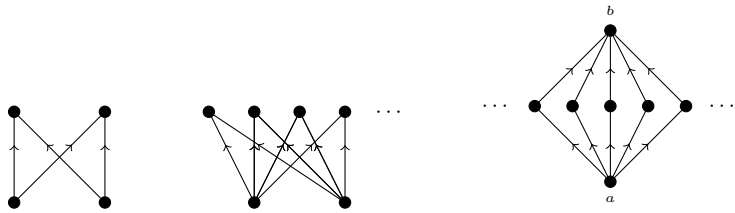

Figure 2: Forbidden subsystems related to bounded completeness, bifiniteness, and interval-compactness.

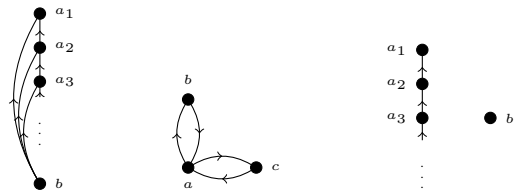

Figure 3: Forbidden subsystem (downward)

tem, $(\mathbb{T}, \leq^{\mathsf{op}})$ *is an algebraic domain if the state space doesn't contain the subsystems in figure 3 (again, arrows indicate paths and not necessarily direct successors).*[10] *We then refer to* $(\mathbb{T}, \leq)$ *as* downward trajectory domain.

We'll next consider how to combine the upward and downward order into a new structure that can interpret the logic HYPE.

5 The combined trajectory domain and a logic

As mentioned, when we investigate a system, we're both interested in the limit-behavior (effects) and the history (causes). Thus, in particular, when we reason about the limit-behavior of the system, we thus should also take into account from where the limit was reached. This suggests to replace

[10]The following side conditions are imposed: In the right-most subsystem, there are trajectories $t', t_1, t_2, \ldots$ ending in $b, a_1, a_2, \ldots$, respectively, such that $[t_1] > [t_2] > \ldots$ and, for all n, $[t_n] \not\leq [t']$.

(infinite) limit-behaviors by new elements representing the different possibilities from where these limit-behaviors can be reached. We achieve this by essentially mirroring $\mathbb{T}$ at the infinite trajectories. To be precise, let t be a trajectory. Then the set

$$[\![t]\!] := \big\{[t'] : t' \text{ infinite trajectory and } t \preceq t'\big\}$$

records the possible limit behavior of t. If t is infinite, $[\![t]\!] = \{[t]\}$, whence we identify it with $[t]$. To get the refined limit states that keep track of the origin from where they were reached, we add the new elements $([\![t]\!], [t])$ to $\mathbb{T}$ for all finite $[t]$. Qua refined limits, the new elements are above the respective old ones: If $[t'] \in \mathbb{T}$, then $[t'] \leq ([\![t]\!], [t])$ iff $[t']$ and $[t]$ share a common limit-behavior, that is, there is $[t_m] \in [\![t]\!]$ such that $[t'] \leq [t_m]$. The new elements are ordered among each other as follows: the further away the origin from which the limit (represented by the new element) was reached, the further up is the new element. That is, $([\![t]\!], [t]) \leq ([\![t']\!], [t'])$ iff $[t'] \leq [t]$. For reasons of symmetry, we can also think of the old elements $[t]$ as really being of the form $([t], [\![t]\!])$. We summarize.

Definition 2 (Combined trajectory domain) *For a (non-) deterministic system, define $\mathbb{T}_c$ as the union of $\{\bot, \top\}$ and*

$$\begin{aligned} L &:= \big\{([t], [\![t]\!]) : \bot \neq t \textit{ finite trajectory}\big\}, \\ U &:= \big\{([\![t]\!], [t]) : \bot \neq t \textit{ finite trajectory}\big\}, \\ M &:= \big\{([\![t]\!], [\![t]\!]) : t \textit{ infinite trajectory}\big\}, \end{aligned}$$

called the lower half*,* upper half*, and the* limit trajectories *respectively. Define the order $\leq_c$ on $\mathbb{T}_c$ as follows ($\leq$ is the order on $\mathbb{T}$):*

$$\begin{aligned} &([t], [\![t]\!]) \leq_c ([t'], [\![t']\!]) \quad \textit{if} \quad [t] \leq [t'] \\ &([\![t]\!], [t]) \leq_c ([\![t']\!], [t']) \quad \textit{if} \quad [t] \geq [t'] \\ &([t], [\![t]\!]) \leq_c ([\![t']\!], [t']) \quad \textit{if} \quad \exists [t_m] \in [\![t']\!] : [t] \leq [t_m], \end{aligned}$$

and $\bot$ and $\top$ are the $\leq_c$-least and biggest element, respectively. The order for limit trajectories $[t]$ is given via the above by the identification $[\![t]\!] = [t]$. We call $(\mathbb{T}_c, \leq_c)$ the combined trajectory domain *of the system.*

An involution poset $(P, \leq, ')$ is a poset $(P, \leq)$ with a function $\cdot' : P \to P$ such that $x'' = x$ and $x \leq y$ implies $y' \leq x'$. We observe that it is an involution to switch perspectives from regarding a trajectory as finite trajectory to regarding some of its limit behavior as being reached from that trajectory.

Proposition 1 *Let $(\mathbb{T}_c, \leq_c)$ be the combined trajectory domain of a (non-) deterministic system. Define a mapping $\cdot^* : \mathbb{T}_c \to \mathbb{T}_c$ by*

$$\big((t, [\![t]\!])\big)^* := ([\![t]\!], t) \quad \textit{and} \quad \big(([\![t]\!], t)\big)^* := (t, [\![t]\!]).$$

Then $\cdot^*$ is an involution. Moreover, if τ is finite, $\tau \leq_c \tau^*$, and if τ is infinite, $\tau^* = \tau$.

We now can get to the logic. Leitgeb (2018) recently developed a general logical system called HYPE. Although not originally intended, we show that this logic can, in fact, be interpreted by dynamical systems.[11]

We first recap HYPE (Leitgeb, 2018). Fix a propositional language with variables $p_1, p_2, \ldots$ and connectives $\neg, \wedge, \vee, \rightarrow$. Let Lit be the set of literals (negated or non-negated propositional variables). Given a literal l, its dual is denoted $\bar{l}$.[12] A valuation on a nonempty set M is a function $V : M \to \mathcal{P}(\mathsf{Lit})$. A HYPE *model* $\mathfrak{M}$ (for our fixed propositional language) is a quadrupel $(M, V, \circ, \perp)$ where M is a nonempty set, V a valuation on M, and the following axioms hold—for the detailed formulation of the axioms see Leitgeb (2018).

- $\circ$ is a partial binary function from $M \times M$ to M (the *fusion function*) such that V is $\circ$-monotone and $\circ$ is reflexive, commutative, and weakly associative.
- $\perp$ is a binary symmetric relation on M (the *incompatibility relation*) such that literal incompatibility entails $\perp$, and $s \perp s'$ entails $s \circ s'' \perp s'$, whenever defined.
- For every $s \in M$ there is a unique $s^* \in M$ (the *star image of* s) such that $V(s^*) = \{\bar{v} : v \notin V(s)\}$, $s^{**} = s$, $s \not\perp s^*$, and s^* is the $\circ$-largest state compatible with s.

We define $s \leq s'$ if $s \circ s'$ exists and $s' = s \circ s'$. Formula satisfaction and logical consequence in HYPE models are defined as follows Leitgeb (2018).

- $s \models v$ iff $l \in V(s)$ (where l is a literal).
- $s \models \neg\varphi$ iff for all s', if $s' \models \varphi$, then $s \perp s'$.

[11] A hint that such an interpretation is possible is the HYPE model consisting of fixed-points of the untyped truth-prediacte (Leitgeb, 2018)—and fixed-point operators are a general form of computation.

[12] So if $l = p$, then $\bar{l} = \neg p$, and if $l = \neg p$, then $\bar{l} = p$.

(A1) $\vdash \top$

(A2) $\vdash A \to A$

(A3) $\vdash A \to (B \to A)$

(A4) $\vdash A \to (B \to C) \to ((A \to B) \to (A \to C))$

(A5) $\vdash A \wedge B \to A$

(A6) $\vdash A \wedge B \to B$

(A7) $\vdash A \to A \vee B$

(A8) $\vdash B \to A \vee B$

(A9) $\vdash A \to (B \to A \wedge B)$

(A10) $\vdash (A \to C) \to ((B \to C) \to (A \vee B \to C))$

(A11) $\vdash A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$

(A12) $\vdash A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$

(A13) $\vdash A \leftrightarrow \neg\neg A$

(A14) $\vdash \neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$

(A15) $\vdash \neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$

(A16) $\dfrac{\vdash A \to B}{\vdash \neg B \to \neg A}$

(A17) $A, A \to B \vdash B$

Figure 4: The system HYPE as presented by Leitgeb (2018).

- $s \models \varphi \wedge \psi$ and $s \models \varphi \vee \psi$ as usual.
- $s \models \varphi \to \psi$ iff for all s', if $s' \models \varphi$ and $s \circ s'$ is defined, then $s \circ s' \models \psi$.

Given a set Γ of formulas and a formula ψ, $\Gamma \models \psi$ iff for all HYPE models $\mathfrak{M}$ and states s of $\mathfrak{M}$, if $s \models \varphi$ for all $\varphi \in \Gamma$, then $s \models \psi$.

Leitgeb (2018) provides a sound and complete logic for HYPE models with this semantics. We'll refer to this logic as HYPE which we repeat for convenience in figure 4. The logic HYPE is well-understood and well-behaved. It not only is sound and complete (via a canonical model construction) with respect to the HYPE models, it also has the deduction theorem, the disjunction property, the finite model property, and is decidable. Moreover, it contains first-degree entailment, conservatively extends intuitionistic logic, and the structures of HYPE models are well-known from ordinary mathematics. (For all these results see Leitgeb (2018).)

To show that the combined trajectory domain carries the structure of a HYPE model, we'll make use of the fact that it has an involution (proposition 1).

Proposition 2 (Involution posets as HYPE models) *Let $(M, \leq, ^*)$ be a non-empty involution poset. Let $V : M \to \mathcal{P}(\mathsf{Lit})$ be a valuation such that*

(i) V is monotone ($s \leq s'$ implies $V(s) \subseteq V(s')$),

(ii) If $l \in V(s)$ and $\bar{l} \in V(s')$, then $s \not\leq s'^*$, and

(iii) $V(s^*) = \{\bar{l} : l \notin V(s)\}$.

Then $(M, V, \circ, \bot)$ is a HYPE *model where $s \circ s'$ is the least $\leq$-upper bound of $\{s, s'\}$, if it exits, and $s \perp s'$ iff $s \not\leq s'^*$ (so $\perp$ is orthogonality in the involution poset).*

So it remains to find appropriate valuations on the combined trajectory domain $(\mathbb{T}_c, \leq_c)$. We do so by lifting a valuation on the state space to a valuation of trajectories as follows.

We start with a valuation $W : S \to \mathcal{P}(\mathsf{Lit})$ of the state space of the system. This captures the idea that we can measure the system with respect to the properties $p_1, \neg p_1, p_2, \neg p_2, \ldots$. That is, any state of the system carries the information of which properties certainly obtain and which certainly do not obtain at that current state of the system. So $[\![l]\!]_W := \{s : l \in W(s)\}$ describes an area of the state space where l holds (or is made true).[13]

We say that W is *separated* if for all p, there is no state that is reachable from some state in $[\![p]\!]_W$ and from some state in $[\![\neg p]\!]_W$.[14] We say that W is *bifurcating* if for all p and trajectories t, if t is undecided on p, then both p and $\neg p$ can be forced. That is, if no trajectory equivalent to t contains a p- or $\neg p$-state, then t can be extended to trajectories t_0 and t_1 such that t_0 is equivalent to a trajectory containing a p-state and t_1 is equivalent to a trajectory containing a $\neg p$-state.

How do we get from a valuation W of the state space to the valuation V of trajectories? The most straightforward idea is to take the valuation of a (lower half) trajectory as collecting all the properties of the states that it has visited. For a limit behavior we collect all the properties that can occur in that limit-behavior.

$$V_W\big([t], [\![t]\!]\big) := \bigcup_{t_0 \in [t]} \bigcup_{n \in \mathbb{N}} W\big(t_0(n)\big)$$

$$V_W\big([\![t]\!], [t]\big) := \bigcup_{[t_0] \in [\![t]\!]} \bigcup_{n \in \mathbb{N}} W\big(t_0(n)\big).$$

We call the valuation $V_W : \mathbb{T}_c \to \mathcal{P}(\mathsf{Lit})$ defined in this way the *valuation induced by W*. We now have the promised result.

Theorem 3 (Logic of $\mathbb{T}_c$) *Let $(\mathbb{T}_c, \leq_c)$ be the combined trajectory domain of a (non-) deterministic system. Let $W : S \to \mathcal{P}(\mathsf{Lit})$ be a separated and*

[13] If the state space S carries a topology, it is natural to demand $[\![l]\!]_W$ to be open—this captures the idea that l is verifiable by finitary measurements (Vickers, 1989).

[14] In other words, if from a state s it is possible that the system develops into a state that is also reached from a $\neg p$ state, then p doesn't "certainly" obtain at s. Rather, p cannot be conclusively decided at state s.

bifurcating valuation of the state space. Then $(\mathbb{T}_c, V_W, \circ, \bot)$ *is a* HYPE *model where* V_W *the valuation induced by* W*,* $\tau \circ \tau' = \tau \vee \tau'$*, if it exits, and* $\tau \bot \tau' \Leftrightarrow \tau \not\leq \tau'^*$.

What do the logical operations mean? For example, a finite trajectory makes $\neg\varphi$ true iff φ is false in the limit (as reached from this trajectory). And $\varphi \to \psi$ is true at a finite trajectory $[t]$ if any future state that is reachable through a φ-area of the state space is also reachable through a ψ-area. (This is equivalent to any infinite $t_m \in [\![t]\!]$ making the classical conditional $\neg\varphi \vee \psi$ true.) An interesting open question is whether HYPE is also complete for this trajectory domain interpretation (and, if not, which extension of HYPE is).

6 Conclusion and further developments

We started with the aim of developing a logic together with a mathematical semantics to reason about both the long-term behavior and the history of a system. We provided a first such instance with the trajectory domain as mathematical structure and HYPE as logic.

We end with three areas of further development. First, since the general notion of a (non-) deterministic system that we've worked with is a transition system, it's natural to exploit their coalgebraic character. In addition to the homological considerations mentioned above, this might also provide more structural definitions of behavioral equivalence (bisimulation). More generally, it seems worth further exploring the link between the two classical subjects of computer science – transition systems and domain theory – that our results provide.

Second, in the spirit of explainable AI, we might wish to show that for every computing system there is a 'transparent' system with the same behavior. In our framework, this would translate into the following—so to speak the *Hauptvermutung* of the project. For every appropriate (non-) deterministic system S, there is another 'transparent' system S' whose trajectory domain is isomorphic (or otherwise closely related) to that of S. (Note the analogy to DFA minimization in automata theory.)

Third, the order structures found in the trajectory domains are, surprisingly, similar to order structures found in the causality order of spacetimes (in general relativity). This can be seen as follows. Define $\mathbb{T}_{\mathsf{fin}}$ as the finite and nonempty elements of $\mathbb{T}$ end exclude the subsystems of theorems 1 and 2, and exclude the 'infinite diamond' of figure 2. Then $(\mathbb{T}_{\mathsf{fin}}, \leq)$ is a causal set in the sense of the causal set approach to quantum grav-

ity (Bombelli, Lee, Meyer, & Sorkin, 1987). Do the excluded subsystems have physical meaning under this interpretation? Moreover, $(\mathbb{T}_{\mathsf{fin}}, \leq)$ then also is a globally hyperbolic poset in the sense of Martin and Panangaden (2006) and thus in the same category as the causality order of globally hyperbolic spacetimes. Does this relate cosmic censorship (valid on globally hyperbolic spacetimes) to bounded non-determinisim (valid when diamond excluded)?

References

Abramsky, S., & Jung, A. (1994). Domain theory. In S. Abramsky, D. M. Gabbay, & T. S. E. Maibaum (Eds.), *Handbook of Logic in Computer Science.* Oxford: Clarendon Press.

Bombelli, L., Lee, J., Meyer, D., & Sorkin, R. D. (1987). Space-time as a causal set. *Physical Review Letters*, *59*(5), 521-524.

Hornischer, L. (2018). *Trajectory domains: describing the behavior of computing systems.* (Unpublished manuscript.)

Leitgeb, H. (2018). Hype: A system of hyperintensional logic (with an application to semantic paradoxes). *Journal of Philosophical Logic*. doi: 10.1007/s10992-018-9467-0

Martin, K., & Panangaden, P. (2006). A domain of space-time intervals in general relativity. *Communications in Mathematical Physics*, *267*, 563-586.

Reimann, M. W., Nolte, M., Scolamiero, M., Turner, K., Perin, R., Chindemi, G., … Markram, H. (2017). Cliques of neurons bound into cavities provide a missing link between structure and function. *Frontiers in Computational Neuroscience*, *11*, 48. doi: 10.3389/fncom.2017.00048

Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Berlin: Springer.

Scott, D. (1970). *Outline of a Mathematical Theory of Computation* (Tech. Rep. No. PRG02). Oxford University Computing Laboratory.

Vickers, S. (1989). *Topology via Logic*. Cambridge: Cambridge University Press.

Levin Hornischer
University of Amsterdam, Institute for Logic, Language and Computation
The Netherlands
E-mail: `l.a.hornischer@uva.nl`