Progress in Computer Science and Applied Logic
29

Patrick Schultz
David I. Spivak

Temporal Type Theory

A Topos-Theoretic Approach to Systems and Behavior

Birkhäuser

Progress in Computer Science and Applied Logic

Volume 29

More information about this series at http://www.springer.com/series/4814

Patrick Schultz • David I. Spivak

Temporal Type Theory

A Topos-Theoretic Approach to Systems and Behavior

Birkhäuser

Patrick Schultz
Massachusetts Institute of Technology
Cambridge, MA, USA

David I. Spivak
Massachusetts Institute of Technology
Cambridge, MA, USA

ISSN 2297-0576 ISSN 2297-0584 (electronic)
Progress in Computer Science and Applied Logic
ISBN 978-3-030-00703-4 ISBN 978-3-030-00704-1 (eBook)
https://doi.org/10.1007/978-3-030-00704-1

Library of Congress Control Number: 2018962871

Mathematics Subject Classification: 03B44, 03B45, 03G30, 18B25, 18F20, 93A30

This book is published under the trade name Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Acknowledgments

Many of the ideas presented here began to take form in 2015–2016 while working closely with Christina Vasilakopoulou, a postdoc in our group at the time. We benefitted greatly from—and thoroughly enjoyed—this collaboration.

We have learned a great deal from our collaborators Kevin Schweiker, Srivatsan Varadarajan, and especially Alberto Speranzon at Honeywell Labs as well as from our sponsors at NASA, including Alwyn Goodloe, Anthony Narkawicz, and Paul Minor. We also want to thank Aaron Ames, Brendan Fong, Tobias Fritz, Ben Sherman, and Rémy Tuyéras for helpful discussions. Thanks also to Ben Sherman and Rémy Tuyéras for comments on an early draft of this book.

The work presented here was supported by NASA grant NNH13ZEA001N as well as by AFOSR grants FA9550–14–1–0031 and FA9550–17–1–0058.

Contents

Chapter 1
Introduction

1.1 Overview

In this book we provide a new mathematical formalism for proving properties about the behavior of systems. A system is a collection of interacting components, each of which may have some internal implementation that is reflected in some external behavior. This external behavior is what other neighboring systems interact with, through a shared environment. Properties of a behavior can be established over a given duration (sometimes called frame or window) of time, and we propose a mathematical language for working with these behavioral properties.

1.1.1 Behavior Types

A *behavior type* B is the information of a set of "different things that can occur" over any length of time. For example, a movie is a behavior type: given any duration of time, there is a set of possible snippets of the movie that have said duration, and the snippets of the movie are what we would call its behaviors. In fact, "all possible movies" is another a behavior type, because to any duration of time, we could associate the set of all 24-frame/second sequences of photographs together with an overlay of sound. All possible music, all possible fights or boxing matches, all behaviors that an airplane or set of airplanes are capable of, etc.—each of these can be modeled as a behavior type.

We give a category-theoretic description of behavior types, using the language of sheaves. That is, to every time window, say of length ℓ, a behavior type B is responsible for providing a set $B(\ell)$ of all behaviors, called *length-ℓ sections of* B, that can possibly occur over this time window. And for every inclusion of one time window into another, the behavior type B is responsible for providing a *restriction map* that restricts each long behavior to the shorter time window.

P. Schultz, D. I. Spivak, *Temporal Type Theory*, Progress in Computer Science and Applied Logic 29, https://doi.org/10.1007/978-3-030-00704-1_1

We say a behavior type *has composable sections* when, for any two overlapping time windows, behaviors over the first that match behaviors over the second can be *glued* to form a behavior over the union time window.

Although many behavior types have composable sections, not all do. For example, the sheaf of monotonic functions to $\mathbb{R}$ has composable sections, because if for every small interval $[t, t+\epsilon]$, we have $f(t) \leq f(t+\epsilon)$ then for every interval $[t_1, t_2]$ whatsoever, we also have $f(t_1) \leq f(t_2)$. In contrast, the following sort of "roughly monotonic" behaviors are not composable:

$$\forall t_1 \forall t_2.\, (t_1 + 5 \leq t_2) \Rightarrow f(t_1) \leq f(t_2). \tag{1.1}$$

This formula says "if you wait at least 5 seconds between taking samples, you will find f to be increasing." Such a property cannot be determined except on intervals of length at least 5. Any behavior — if tested over a very short time window—will comply with this property, but if one glues two compliant behaviors that agree along an overlap, the result may not be compliant. This example has the property that it satisfies a composition property for pairs of intervals whose overlap is large enough. An example of a behavior type without any sort of composability is what we might call "functions of bounded difference," e.g., satisfying:

$$\forall t_1 \forall t_2.\, |f(t_1) - f(t_2)| \leq 5. \tag{1.2}$$

Composable sections or not, however, all the behavior types we consider will be *sheaves* in an appropriate setting, and morphisms of sheaves allow us to connect one behavior type to another. For example, there is a morphism from the sheaf corresponding to a movie to the sheaf corresponding to its sound-track. One might formalize the system–environment notion as a morphism $S \to E$ and say two systems S_1 and S_2 share an environment when given morphisms $S_1 \to E \leftarrow S_2$. Thus, it is through sheaf morphisms that different behaviors can interact. Two singers may be hearing the same drum beat, or two airplanes may share the same communication channel.

The behavior types and their morphisms form a category $\mathcal{B}$ of sheaves, and as such, a *(Grothendieck) topos*. Toposes are particularly nice categories, in that they enjoy many properties that make them in some sense similar to the category of sets. One way to make this precise is to say that toposes satisfy the Giraud axioms [AGV71]. But to more clearly draw the analogy with **Set**, let us suffice to say that toposes are regular categories, and that they have limits and colimits, effective equivalence relations, exponential objects, and a subobject classifier Ω.

The subobject classifier in particular is an important object when thinking about properties, e.g., properties of behavior. For **Set**, the subobject classifier is $\Omega_{\mathbf{Set}} = \{\text{True}, \text{False}\}$. Many properties take the form of "yes/no" questions; such questions distinguish elements of a set according to a given property. That is, for any set S and property P, there is a corresponding yes/no question $S \to \{\text{True}, \text{False}\}$ that classifies the subset of P-satisfying elements in S. For example, the property of an integer being odd is classified by a function `is_odd`: $\mathbb{Z} \to \{\text{True}, \text{False}\}$.

An analogous fact holds for behavior types. There is a behavior type $\Omega_{\mathcal{B}}$ which classifies behavior subtypes. Rather than "yes/no" questions, properties of behavior types are "compliant when" questions. For example, the property of a continuous function $f : \mathbb{R} \to \mathbb{R}$ being greater than 0 is classified in terms of the largest open subset $\{x \in \mathbb{R} \mid f(x) > 0\}$ on which it is compliant. Similarly, the property of f satisfying Eq. (1.2) is classified by the collection of all open intervals over which f has upper and lower bounds less than 5 apart.

Every topos has an associated internal language and higher-order logic. The symbols of the logic are the typical ones, $\top$, $\bot$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\neg$, $\forall$, and $\exists$, and the sort of reasoning steps allowed are also typical constructive logic. For example, given P and $P \Rightarrow (Q \wedge Q')$, one can derive Q. However, our topos $\mathcal{B}$ is not Boolean, so $P \vee \neg P$ does not hold in general, nor does $(\neg \forall x.\, Px) \Rightarrow \exists x.\, \neg Px$. Again, the reasoning is *constructive*, meaning that a proof of a statement gives a witness to its truth: to prove an existential, one must provide a witness for it and similarly for a disjunction.

One need not imagine behavior types as sheaves when doing this logic—it is all purely formal—and yet anything that can be proven within this logic reflects a specific truth about behavior types as sheaves. The separate-but-connected relationship between the logic and its semantics allows us to work with a highly abstract and complex topos using standard constructive logic, reasoning as though with sets and their elements. Thus for example, one could prove sound theorems about $\mathcal{B}$ in an undergraduate course on formal logic, without ever discussing categories, let alone toposes.

1.1.2 Goal: To Prove Properties of Systems

Our goal is to understand what can possibly occur when multiple components—each with their own type of behavior, interact in any given way. If we know something about the behavior of each component, if each component has a *behavior contract* or guarantee that specifies something about how it will behave among all its possibilities, then we may be able to guarantee something about how the entire system will behave. This is relevant to industries in which different suppliers provide different parts, each with its own behavioral guarantee. In such a setting, one still wants to draw conclusions about the system formed by arranging these components so that they interact in some specified way.

For example, a thermostat, a furnace, and a room comprise three components, each of which may be guaranteed to satisfy a certain behavior contract: the thermostat promises to sense temperature of the room and promises to send a signal to the furnace if the temperature is too low. The furnace promises to heat the room air at a certain rate when given the signal, and the room promises to lose heat at some maximum rate. We may want to prove a behavior contract for the whole thermostat–furnace–room system, e.g., that the temperature will remain within certain bounds.

In this book, we provide a formal system—a temporal type theory and a higher-order temporal logic—in which such proofs can be carried out.

In other words, we will provide a language for proving properties of interconnected dynamical systems, broadly construed. Dynamical systems are generally considered to come in three flavors: continuous, discrete, and hybrid, according to how time is being modeled. In the above example, the temperature of the room could be modeled by a continuous dynamical system, the thermostat by a discrete dynamical system, and the whole setup is a hybrid system. However, our notion of behavior type is much more general, serving as a sort of "big tent" into which other conceivable notions of behavior can be translated and subsequently compared. For example, there is no differential equation whose solution set is the roughly monotonic curves from Eq. (1.1), because two such trajectories f can "cross," but these trajectories constitute a perfectly good behavior type in $\mathcal{B}$. Behavior types also include infinite-dimensional systems, and as such may be a good language for considering adaptive control [ÅW13], though we do not pursue that here.

There are already many different temporal logics—including linear temporal, (metric, Halpern–Shoham) interval temporal, and signal temporal [RU12, AFH96, MN04, HS91]—for describing behavior. Some of these generalize others, but none is most general. While we do not prove it here, we believe and will give evidence for the assertion that our temporal logic can serve as a "big tent," in which all other such logics can embed.

Some of these temporal logics have very powerful model checkers that can produce a proof or counterexample in finite (though often exponential or doubly exponential) time, and we make no such claim: our logic is wildly undecidable. However, as a big tent, our formalism can embed these proofs from other logics and integrate them as a last step in a process.[1] Our work may also be useful to those who simply want a topos-theoretic and category-theory-friendly approach to understanding behavior.

1.2 Behavior Types as Sheaves

While the introduction has been informal so far, the work in this book is fairly technical. We assume that the reader has a good understanding of category theory, including basic familiarity with Grothendieck toposes and internal languages; see [Joh02, MM92]. Some familiarity with domain theory [Gie+03], and of frames and locales [PP12] would be useful but is not necessary.

Consider the usual topological space of real numbers, $\mathbb{R}$, which has a basis consisting of open intervals (r, s). One might be tempted to model a behavior type as a sheaf S on $\mathbb{R}$, but we did not make this choice for several reasons that we will

[1] It may be objected that these temporal logics are often Boolean whereas our topos $\mathcal{B}$ is not; in this case, one would simply embed the statements into the Boolean subtopos $\mathcal{B}_{\neg\neg} \subseteq \mathcal{B}$.

soon explain. Nonetheless, sheaves on $\mathbb{R}$ will be important for our work, and they give a good starting point for the discussion, so let's briefly consider the structures and properties of a sheaf B on $\mathbb{R}$.

For any open interval $(r, s) \subseteq \mathbb{R}$, there is a set $B(r, s)$ whose elements we call behaviors of type B over the interval. If $r \leq r' \leq s' \leq s$, there is a restriction map $B(r, s) \to B(r', s')$ expressing how a system behavior over the longer interval restricts to one over the shorter interval. Given two overlapping open intervals, say (r_1, r_3) and (r_2, r_4) where $r_1 < r_2 < r_3 < r_4$, and given behaviors $b' \in B(r_1, r_3)$ and $b'' \in B(r_2, r_4)$ such that $b'\big|_{(r_2,r_3)} = b''\big|_{(r_2,r_3)}$, there exists a unique behavior $b \in M(r_1, r_4)$ extending both: $b\big|_{(r_1,r_3)} = b'$ and $b\big|_{(r_2,r_4)} = b''$. This sort of "composition"-gluing condition can be succinctly written as a finite limit:

$$B(r_1, r_4) \cong B(r_1, r_3) \times_{B(r_2,r_3)} B(r_2, r_4) \tag{1.3}$$

A gluing condition—expressed as the limit of a certain diagram—holds more generally for any open covering of an open interval by other open intervals in $\mathbb{R}$. However, it suffices to add just one more kind, which we might call the "continuity" gluing condition; namely, those for the following sort of telescoping inclusion:

$$B(r, s) \cong \lim_{r<r'<s'<s} B(r', s'). \tag{1.4}$$

These two gluing conditions—composition gluing and continuity gluing—on a functor B allow us to identify it with a sheaf on $\mathbb{R}$. However, we do not take sheaves on $\mathbb{R}$ as our model of behavior types for two reasons: non-composability of behaviors and translation invariance of behavior types. We explain these next.

1.2.1 Non-composable Behaviors

One usually imagines behaviors as composable: given two behaviors that match on an overlapping subinterval, they can be joined to form one long behavior. This was expressed in condition (1.3), but it is not always what we want. For example, if we glue together two roughly monotonic curves as in Eq. (1.1), the result may fail to be roughly monotonic.

Another way to see this is in terms of the internal logic and its semantics. In any sheaf topos—though we will speak in terms of topological spaces—when a proposition is true over some open set, it must be true over every open subset. Such a property in a temporal setting is often called a *safety property*: if a system is to be compliant over an interval of time, it must be compliant over every subinterval. The semantics of safety properties is thus that of *falsifiability*: a system satisfies a proposition over an interval if and only if there is no subinterval on which the proposition is false. For example, a property like "if event a happens, then event b will happen 10 seconds later" is impossible to falsify on intervals of length less than

10 s, so we must consider it to be vacuously satisfied on such short intervals. Again, if an interval is too short to falsify a proposition, the proposition is deemed true on that interval.

Now we can more clearly see why the composition gluing condition creates a problem. It says that if a proposition is true on each element of an open cover, then it is true on the union. In $\mathbb{R}$, we have $(0, 4) \cup (2, 6) = (0, 6)$, so if a proposition is unfalsifiable on $(0, 4)$ and on $(2, 6)$ we must call it true on $(0, 6)$. Again, consider the formula Eq. (1.1). It gives an example of a proposition that is unfalsifiable on intervals whose length is strictly less than 5, and hence would be true on $(0, 4)$ and $(2, 6)$. Thus, it would be forced by the composition-gluing condition to be true on $(0, 6)$—and, by induction, on an interval of any length—which is not the semantics we want.

The way we solve the above problem is by enlarging our topological space. Consider the (non-Hausdorff) topological space $\mathbb{IR}$, called the *interval domain*, whose points are compact intervals $[d, u] \subseteq \mathbb{R}$. As with $\mathbb{R}$, the space $\mathbb{IR}$ has a basis of open sets indexed by pairs of real numbers $r < s$. Namely, for any $r < s$, the corresponding basic open is $(r, s) \coloneqq \{[d, u] \mid r < d \leq u < s\} \subseteq \mathbb{IR}$. Then, a subset $U \subseteq \mathbb{IR}$ is open if and only if it can be expressed as a union of these basic opens.

Note that $(0, 4) \cup (2, 6) \neq (0, 6)$ in this topology, because the left-hand side does not contain the point (compact interval) $[1, 5]$. We will see that $\mathbb{IR}$ is a domain, called the *interval domain*, meaning that it is a topological space—the topology defined in the previous paragraph is called the Scott topology for the domain—and its points have a natural poset structure that completely determines its frame of open sets. Likewise, $\mathbb{IR}$ is a sober topological space, so its frame of open sets completely determines its poset of points.

We use $\mathbb{IR}$ rather than $\mathbb{R}$ as our main space of interest because it allows us to capture non-composable behaviors.

1.2.2 Translation-Invariant Behavior Types

The second reason for not using $\mathbb{R}$ persists even when we pass to $\mathbb{IR}$, but is simpler to explain. Namely, both $\mathbb{R}$ and $\mathbb{IR}$ come equipped with a specific reference point in time, the origin. An important assumption in science is that experiments run today are still valid tomorrow, all other things being equal. The concepts being tested may depend on durations of time, but they are independent of the "date," say, the number of years since the big bang or the birth of an influential person. Some behaviors, such as holidays, are dependent on date, but we regard such time dependence as an additional feature—something to made explicit—rather than as the norm.

Another way to understand what we might call "the translation problem" is that there is no connection in $\mathbb{IR}$ between the interval $(0, 1)$ and the interval $(5, 6)$, unless one somehow builds in the translation action of $\mathbb{R}$. But, one would be tempted to have $\mathbb{R}$ also act on the objects of the topos: to every sheaf X and real number $r \in \mathbb{R}$,

there is a translated sheaf, say $X \triangleright r$, whose (a, b)-behaviors are the $(a + r, b + r)$-behaviors of X. This creates an interdependence between the real numbers—as an object in the topos—and the rest of the objects in the topos. We believe that this would have made the logical system much more complex.

To remedy the translation problem, we work in a translation-invariant setting: A behavior that can occur over one interval could also occur over any other. In this setting, the time-line itself becomes a behavior type, we call `Time`. Over a given interval of length ℓ, a behavior of type `Time` can be regarded as the behavior of a perfect clock: it starts at some time t_0 and ends at $t_0 + \ell$. If a specific behavior b is dependent on a choice of clock time, we make that dependence explicit in the sense that t : `Time` will occur in the formula for b.

To get a bit more technical, our base topos $\mathcal{B}$ is a quotient of $\mathsf{Shv}(\mathbb{IR})$, i.e., there is a geometric surjection $p_* \colon \mathsf{Shv}(\mathbb{IR}) \to \mathcal{B}$. One characterization of geometric surjections is that the inverse image part, in this case $p^* \colon \mathcal{B} \to \mathsf{Shv}(\mathbb{IR})$, is faithful. Another is that $\mathcal{B}$ is the category of coalgebras for the left-exact comonad $p^* p_*$ on $\mathsf{Shv}(\mathbb{IR})$. Intuitively, objects in $\mathcal{B}$ can be thought of as sheaves $X \in \mathsf{Shv}(\mathbb{IR})$ that are *translation-invariant*, i.e., for which one has coherent isomorphisms $X(a, b) \cong X(a + r, b + r)$ for every open interval (a, b) and real $r \in \mathbb{R}$. This is the translation action of $\mathbb{R}$ on $\mathbb{IR}$, mentioned above, and we denote by $\mathbb{IR}_{/\triangleright}$ the localization of $\mathbb{IR}$ at the collection of translation maps. $\mathbb{IR}_{/\triangleright}$ is no longer a space—more precisely, it is a category and not a poset—but it has a natural site structure and $\mathcal{B} \coloneqq \mathsf{Shv}(\mathbb{IR}_{/\triangleright})$ is the corresponding topos of sheaves.

In fact, $\mathcal{B}$ is an étendue, meaning that there is a specific object, namely `Time`, such that the slice topos $\mathcal{B}/\mathtt{Time} \cong \mathsf{Shv}(\mathbb{IR})$ is localic.

$$\mathsf{Shv}(\mathbb{IR}) \;\cong\; \mathcal{B}/\mathtt{Time} \;\underset{p_!}{\overset{p_*}{\underset{\longleftarrow}{\overset{\longrightarrow}{\longleftarrow p^* \longrightarrow}}}}\; \mathcal{B} \;\cong\; \mathsf{Shv}(\mathbb{IR}_{/\triangleright})$$

The word étendue means “extent” and indeed objects in $\mathbb{IR}_{/\triangleright}$ are extents—or durations—of time, over which behaviors can occur.

1.2.3 Four Relevant Toposes

All four toposes in the square below play a role in this work:

$$\begin{array}{ccc} \mathsf{Shv}(\mathbb{R}) & \rightarrowtail & \mathsf{Shv}(\mathbb{IR}) \\ \downarrow & & \downarrow{\scriptstyle p_*} \\ \mathcal{B}_\pi & \rightarrowtail & \mathcal{B} \end{array}$$

The toposes in the top row comprise temporal sheaves—those over a specified time-line—whereas those in the bottom row comprise behaviors that are translation-invariant. Sheaves in the left column have composable behaviors, whereas sheaves

in the right column can include more general, non-composable behaviors. We discussed $\mathsf{Shv}(\mathbb{R})$, $\mathsf{Shv}(\mathbb{IR})$, and $\mathcal{B} = \mathsf{Shv}(\mathbb{IR}_{/\triangleright})$ above.

The horizontal maps are geometric embeddings, i.e., they correspond to modalities, and the vertical maps are geometric surjections. The topos $\mathcal{B}_\pi$ and the left-hand and bottom maps are uniquely determined from the top and right-hand maps, by way of the surjection-embedding factorization systems on toposes. The π stands for "pointwise," which means that properties of sheaves in $\mathcal{B}_\pi$ can all be determined in neighborhoods of points in $\mathbb{R}$.

1.3 Temporal Type Theory

The main subject of this book is the definition of a type theory—and its associated logic—that has its semantics in the topos $\mathcal{B} = \mathsf{Shv}(\mathbb{IR}_{/\triangleright})$ discussed above.

The logic we present is higher-order logic, plus subtypes and quotient types. Higher-order logic—as well as its strong connection with topos theory—has been very well-studied [LS88, Fou77, BJ81, Awo16, Joh02], which is part of our motivation for using it. In fact, toposes support not just higher-order logic but also dependent type theories. This means that one can use an automated proof-assistant based on dependent type theory—such as Coq or Lean [CH88, Mou+15]—to validate proofs.

The primary goal of the type theory and logic we present is to support defining and reasoning about behavior types. As such, the logic is capable of expressing statements from standard temporal logics, such as linear temporal logic (LTL). The primary "temporal operator" from LTL is called *until*. The meaning of until—written $\mathcal{U}$ in the logic—is often presented by a formula such as:

$$(\phi_1 \,\mathcal{U}\, \phi_2)(t) \coloneqq \exists(r : \mathtt{Time}).\left[(t < r) \wedge \phi_2(r) \wedge \forall u.\,(t < u < r \Rightarrow \phi_1(u))\right]. \tag{1.5}$$

Here, the temporal propositions ϕ_1 and ϕ_2 are being represented as propositions that have an explicit dependence on time, i.e., as functions $\mathtt{Time} \to \mathtt{Prop}$. As one can attempt to read off from Eq. (1.5), $\phi_1 \,\mathcal{U}\, \phi_2$ constructs a new temporal proposition which is true if both the following hold: ϕ_2 is true sometime in the future, and ϕ_1 is true from now until that point.

In fact, Eq. (1.5) is an example of a formula in our logic. The type `Prop` is what makes the logic higher order, and the type `Time`—which is definable from the one atomic term of our theory, and whose semantics is the behavior type `Time` discussed in Sect. 1.2.2—allows for temporal statements like $\phi_1 \,\mathcal{U}\, \phi_2$ to be expressed. It is from this perspective that we find it reasonable to refer to our system as *temporal type theory*. See Sect. 8.6 for a more in-depth discussion of how our type theory and logic relates to existing temporal logics, in particular to LTL and metric temporal logic.

Every property we can discuss is a safety property, in the sense described in Sect. 1.2.1, so all of our connectives and quantifiers take safety properties to safety

properties. An example of the expressive power of the temporal type theory: we are able to internally define real-valued functions of time, as well as their derivatives, and prove that the derivative satisfies the Leibniz rule.

As a first test that our formal system is strong enough to be useful in practice, we use a simplified version of the safe separation problem for airplanes in the US National Airspace System (NAS). There one wants to avoid situations in which airplanes get too close to one another. To achieve this, the current system consists of an interaction between radar, a traffic collision avoidance system (or TCAS), pilot decision-making, and actuators and thrusters on the surface of the airplane. The position of each airplane roughly follows a differential equation, with time-varying parameters such as "climb at rate $r(t)$," supplied by the pilot. The pilot's decisions take into account the commands from air traffic control and the advice of the TCAS, which alerts the pilot to urgent situations and suggests corrective maneuvers. In turn, this information is determined by the relative position of the airplanes, completing the loop.

The NAS thus requires continuous interaction between many different types of behavior. Some of the components are modeled continuously, such as the motion of the plane, while others are modeled discretely, such as the TCAS alerts and suggestions. The pilot generally takes the advice given and carries it out after some delay. It is the combination of continuous, discrete, and delay behaviors that we believe is the essence of the safe separation problem. In Chap. 8, we will prove a version of the safe separation property—a version that is greatly simplified but that still involves the above essential elements—in which there is only one airplane that must obtain safe separation from the ground.

1.4 Related Work

Spivak et al. [SVS16] presents a topos similar to that described above, in Sect. 1.2. The authors (ourselves plus C. Vasilakopoulou) consider sheaves and presheaves on a certain category **Int** of intervals: sheaves for behavior types and presheaves for behavior contracts. Sheaves on **Int** can be identified with discrete Conduché fibrations—or unique factorization liftings—over the monoid $(\mathbb{R}_{\geq 0}, 0, +)$ of non-negative real numbers. This may have interesting connections with decomposition spaces [GKT15], but more closely related are applications to dynamical systems, pursued in [Law86, BF00, Fio00].

This book considers instead a subtopos $\mathcal{B} \subseteq \mathsf{Shv}(\mathbf{Int})$, obtained by sheafifying **Int**-sheaves or **Int**-presheaves with respect to the "continuity" gluing condition (1.4), and hence building in continuity at the ground floor. This is a natural step, as the category $\mathbf{Int}^{\mathrm{op}}$ is already a continuous category in the sense of Johnstone and Joyal, and $\mathcal{B}$ is equivalent to the topos of continuous functors $\mathbf{Int}^{\mathrm{op}} \to \mathbf{Set}$; see [JJ82] or Appendix B. The present work develops a type theory with a semantics in the sheaf topos $\mathcal{B}$, but which can be used independently of the sheaf semantics.

There are also connections with [JNW96] and following works, where dynamics is considered in terms of morphisms from a given category of paths. For example in [HTP03], an object in the path category is an interval "modeling a clock running on [an open interval] at unit rate," a perspective quite similar to our own.

Our general approach to dynamical systems also has something in common with that of the early cyberneticists, such as Ashby or Weiner; for example, our notion of behavior type is roughly what Ashby [Ash13] calls the "field" of a system. However, our work is more closely aligned with the relatively recent "behavioral approach," as advocated in [Wil07]. There, a dynamical system is defined to be a triple $(\mathbb{T}, \mathbb{W}, \mathcal{B})$, where $\mathbb{T}$ represents time, and which we fix to be $\mathbb{T} = \mathbb{R}$, where $\mathbb{W}$ is an arbitrary set of "signal values," and where $\mathcal{B} \subseteq \mathbb{W}^{\mathbb{T}}$ is the set of possible behaviors as a subset of all functions $\mathbb{T} \to \mathbb{W}$.

An object of the category $\mathcal{B}$ is closely related to such a dynamical system. The primary difference is that, instead of specifying only those possible "infinitely extended" behaviors $\mathbb{T} \to \mathbb{W}$, an object $B \in \mathcal{B}$ must specify a subset $B(a, b) \subseteq \mathbb{W}^{(a,b)}$ for every open interval $(a, b) \subseteq \mathbb{R}$, subject to the conditions that for any $f\colon (a, b) \to \mathbb{W}$:

- if $f \in B(a, b)$, then $a \leq a' < b' \leq b$ implies $f|_{(a',b')} \in B(a', b')$;
- if $f|_{(a',b')} \in B(a', b')$ for any $a < a' < b' < b$, then $f \in B(a, b)$;
- if $f \in B(a, b)$, then $[t \mapsto f(t - r)] \in B(a + r, b + r)$ for any $r \in \mathbb{R}$.

We find this to be more natural from a modeling perspective, since an infinitely extended behavior is by definition unobservable.

Behavior types $B \in \mathcal{B}$ are also more general, in that we can consider behavior types in which there exist behaviors $B(0, 5)$ which admit no extension to $B(0, 10)$, say. As an example of this phenomenon, we could define $B(a, b)$ to be the set of all differentiable functions $f\colon (a, b) \to \mathbb{R}$ satisfying the differential equation $f' = 1 + f^2$. Then, $\tan(t)\colon (-\frac{\pi}{2}, \frac{\pi}{2}) \to \mathbb{R}$ defines a behavior in $B(-\frac{\pi}{2}, \frac{\pi}{2})$ for which there is no extension to a behavior on any larger interval.

As another example, consider the behavior type B of possible trajectories of population levels of wolves and deer in some ecosystem. For the sake of analysis, we may want to consider the subobject $B' \hookrightarrow B$ consisting of only the desired behaviors, say where neither the wolves nor the deer go extinct. It is certainly possible for the ecosystem to be in a state in which, according to the model being used, there does not exist a future in which neither species goes extinct.

Finally, as mentioned earlier in this introduction, one of the goals of this work is to serve as a "big tent," so that many approaches to models of systems can be embedded into our category $\mathcal{B}$, and many temporal logics invented to analyze those models can be embedded into our axiomatics. In that sense, our goal is not to compete with other bodies of work, but to help weave them all together.

1.5 Notation, Conventions, and Background

Here, we lay out some of the basic notation and conventions used throughout the book. We then give a lightning review of sites, sheaves, and toposes.

1.5.1 Notation and Conventions for Categories

Notation 1.1 We say that a set S is *inhabited* if there exists an element $s \in S$. This is the constructive way to say that S is nonempty. Similarly, we say that a category is inhabited if it has an object.

For any $n \in \mathbb{N}$, we sometimes write n to denote the linearly ordered category $\bullet^1 \to \cdots \to \bullet^n$ with n objects.

All categories in this book are 1-categories unless otherwise stated. We denote composition of morphisms $A \xrightarrow{f} B \xrightarrow{g} C$ in the classical "Leibniz" order $g \circ f \colon A \to C$, and we denote identity morphisms as id. If $\mathcal{C}$ is a category, we may denote the hom-set between objects $c, c' \in \mathcal{C}$ either by $\mathcal{C}(c, c')$ or $\mathrm{Hom}_{\mathcal{C}}(c, c')$, or simply $[c, c']$ if $\mathcal{C}$ is known from context.

1.5.2 Definition of Sites, Sheaves, and Toposes

Whenever we speak of a topos in this book, we always mean a Grothendieck topos, which is the category of sheaves on a site. We quickly remind the reader of the definition of site and topos. Readers for whom the definition seems overly abstract may simply skim it on a first reading. The following is taken from [Joh02].

Definition 1.2 (Coverage, Site, Sheaf, Topos, and Geometric Morphism) Let $\mathcal{C}$ be a category. For any object $U \in \mathcal{C}$, a *family over* U is a subset of $\mathrm{Hom}_{\mathcal{C}}(-, U)$. Explicitly, a family consists of an object U, a set I, and for each $i \in I$ a morphism $f_i \colon U_i \to U$ for some $U_i \in \mathcal{C}$. Let $\Phi(\mathcal{C})$ denote the set of families in $\mathcal{C}$.

A *coverage* χ consists of a set T and a function $F \colon T \to \Phi(\mathcal{C})$ satisfying the following condition. Suppose given an object $U \in \mathcal{C}$, a morphism $g \colon V \to U$, and a $t \in T$, such that $F(t) = (f_i \colon U_i \to U \mid i \in I)$. Then, there exists a $t' \in T$ such that $F(t') = (h_j \colon V_j \to V \mid j \in J)$ and such that for each $j \in J$ there exists some $i \in I$ and some $g_{j,i} \colon V_j \to U_i$ making the following square commute:

$$\begin{array}{ccc} V_j & \xrightarrow{h_j} & V \\ {\scriptstyle g_{j,i}}\downarrow & & \downarrow{\scriptstyle g} \\ U_i & \xrightarrow[f_i]{} & U \end{array}$$

A *site* is a category C equipped with a coverage $\chi = (T, F)$. For any $t \in T$, we say $F(t) = (U, I, f)$ is a χ*-covering family over* U.

Suppose given a functor $X : C^{\mathrm{op}} \to \mathbf{Set}$, a family $(f_i : U_i \to U \mid i \in I)$, and an element $x_i \in X(U_i)$ for each $i \in I$. We say that the x_i are *compatible* with respect to the family if, for every pair of elements $i, i' \in I$, object $W \in C$, and commutative square:

$$\begin{array}{ccc} W & \xrightarrow{g_i} & U_i \\ {\scriptstyle g_{i'}}\downarrow & & \downarrow{\scriptstyle f_i} \\ U_{i'} & \xrightarrow[f_{i'}]{} & U \end{array}$$

the equation $X(g_i)(x_i) = X(g_{i'})(x_{i'})$ holds in $X(W)$.

A *sheaf* on a site $S = (C, \chi)$ is a functor $B : C^{\mathrm{op}} \to \mathbf{Set}$ such that for each χ-covering family $(f_i : U_i \to U \mid i \in I)$ and compatible family $\{b_i \mid i \in I\}$, there exists a unique $b \in B(U)$ such that $B(f_i)(b) = b_i$ for each $i \in I$. A *morphism of sheaves* $B \to B'$ is simply a natural transformation of functors, and this defines the category of sheaves on (C, χ). The category of sheaves on the site S is denoted $\mathsf{Shv}(S)$.

A *topos* is any category $\mathcal{E}$ for which there exists a site S and an equivalence of categories $\mathcal{E} \cong \mathsf{Shv}(S)$. If $\mathcal{E}'$ is another topos, a *geometric morphism* $f : \mathcal{E} \to \mathcal{E}'$ is a functor which has a left adjoint, such that the left adjoint preserves finite limits.

1.6 What to Expect from the Book

We hope the reader can learn useful things by reading this book. Of course, what is learned depends on the reader: her background, interests, level of effort, etc. Before giving an outline of the chapters, we will discuss some potential items of interest.

1.6.1 What the Reader Can Hope to Learn

Unfortunately, this book could not be written for a general audience. For example, the authors are well-aware that anyone unfamiliar with toposes probably had difficulty knowing how to think about Definition 1.2. There is certainly material in this book that is amenable to such readers, including Sect. 2.4, Chaps. 4, 5, 7 and 8.

However, as mentioned above in Sect. 1.2, the book is mainly written for readers who have seen toposes before. We would like to think that students of all levels can learn something about toposes by reading this book. Indeed, it can be considered as an extended example of a single topos $\mathcal{B}$ as well as its slices and subtoposes. Chapter 4 is a stand-alone chapter that is meant to introduce readers to type theory and logic, especially as they relate to toposes. For example, we provide

detailed type-theoretic accounts of real numbers and other numeric objects that have semantics in an arbitrary topos.

While some knowledge about Scott domains would certainly be useful at times, it is not necessary. Again, we hope that this book will provide insight into the subject of domain theory by offering an extended example. Semantically, we will work in a quotient of the domain $\mathbb{IR}$, but we will also have occasion to consider domains internal to a topos. For example, the derivative of a continuous function is an interval-valued function, and it is defined in a domain-theoretic style (see Definition 7.25).

Finally, the reader will also learn about temporal logic. We spend a bit of space to try and convey how existing temporal logics fit into our theory. But mostly, we hope that the reader will have interest in our own particular *higher-order temporal logic*, as it motivates everything in the book.

1.6.2 Contributions

Our main contribution is to offer a new temporal type theory (the first of its kind as far as we know), together with a novel topos-theoretic semantics. We believe that it can mediate between several existing formal systems for dealing with time, e.g., for describing cyber-physical systems.

Along the way to understanding Dedekind real numbers—and generalizations like proper and improper intervals—in our topos-theoretic semantics, we were led to a number of results which we believe might be of independent interest. When using the temporal type theory to reason about behaviors which are represented by continuous real-valued functions, we found the need to make use of Dedekind real number objects in various subtoposes as well as the standard Dedekind real number object. In the type theory, these appear as objects defined using versions of the standard Dedekind axioms which have been modified by modal operators. Studying these "modal Dedekind real number objects" was greatly simplified by considering generalizations which remove some of the standard Dedekind axioms. The reason is that these generalized numeric objects form *domains*, which one can think of as a particularly nice class of topological spaces which are intimately connected to order theory. We found that for particular kinds of subtopos—in particular closed, quasi-closed, and dense proper subtoposes—and for particular kinds of domains which are presented by sufficiently nice bases, we could give strong comparisons between domains in a subtopos and domains in the enclosing topos.

These general domain-theoretic results are collected in Appendix A—a section we tried to make readable, independently of the rest of the book—while the special cases about generalized Dedekind numeric objects are collected in Chap. 7. As an example application, the theory of differentiation that we develop in the temporal type theory in Sect. 7.3 makes essential use of generalized Dedekind numeric objects in several different subtoposes.

In addition, we contribute a few other new ideas to the literature. In Sect. 2.5, we prove a result about dense morphisms of posites. While this formally follows from the results in [Shu12], we have provided a new direct proof of the simpler posite case. In Sect. 4.1, we give an informal introduction to type theory and higher-order logic, a subject for which informal accounts seem to be lacking in the literature.

In Chap. 8, we generalize the usual notion of hybrid system from the control theory literature. We also show how to integrate several different temporal logics into our own.

Finally in Appendix B.2, we prove the existence and various properties—e.g., regarding interaction with the Ind-completion—of the (connected, discrete bifribration) orthogonal factorization system on **Cat**. Although this material is well-known, it seemed difficult to locate in the literature.

1.6.3 Chapter Outline

This book has eight chapters and two appendices. Chapter 1 has hopefully summarized and explained the goals of this work: to study temporal properties of a very general class of behavior types, using the language and tools of toposes. We said that each behavior type is modeled as a set of possible behaviors over each interval of time, namely as sheaves on some sort of time-line. However, we made two choices that keep this from being as simple as one might be tempted to expect. The first, and more important, is that matching behaviors on overlapping intervals need not be composable (Sect. 1.2.1); the second is that the set of possible behaviors over an interval should not depend on its position in the time-line (Sect. 1.2.2).

We formalize these two ideas in the following two chapters. In Chap. 2, we define the time-line we will use, which is called the interval domain and denoted $\mathbb{IR}$. The real line sits inside it as the length-0 intervals $\mathbb{R} \subseteq \mathbb{IR}$. We give a continuous bijection between $\mathbb{IR}$ and the upper half-plane in $\mathbb{R}^2$, which not only allows the reader to visualize $\mathbb{IR}$ but also proves quite useful for semantic purposes throughout the book. In this chapter, we also review the definition of posites, (0, 1)-sheaves, Scott domains, and give four equivalent definitions of the time-line as a posite.

In Chap. 3, we deal with translation invariance, roughly by taking the quotient of $\mathbb{IR}$ with respect to the translation action of the group $\mathbb{R}$. The result is a category $\mathbb{IR}_{/\triangleright}$, which is no longer a Scott domain, but instead a continuous category, in the sense of Johnstone and Joyal. The topos of sheaves on the corresponding site is $\mathcal{B}$, our main topic of study throughout the book.

We transition from the external viewpoint to the internal viewpoint in Chap. 4. This chapter is fairly independent of the rest of the book, standing as a review of some connections between toposes, type theory, and higher-order logic. In particular, we spend the first half of the chapter reviewing these notions at a high-enough level not to get bogged down in specifics, but allowing the book to be fairly self-contained in terms of its type theory and logic. We then provide a short section recalling the notion of modalities j—also known as Lawvere–Tierney topologies—and their relationship to subtoposes. We spend the remainder of the

chapter discussing real numbers and related numeric objects in subtoposes. In particular, we explain j-local arithmetic and inequalities.

The technical heart of the book is in Chaps. 5 and 6. In Chap. 5, we axiomatize the higher-order logic of our temporal type theory, which includes one atomic predicate—defining `Time`—and several axioms. We also discuss a few modalities that correspond to important subtoposes of $\mathcal{B}$. In Chap. 6, we prove the soundness of our axioms in $\mathcal{B}$. This requires explaining the semantics of the various numeric objects, such as the real numbers, as well as the semantics of the modalities.

In Chap. 7, we work with numeric objects relative to various modalities j. For example, using what we call the "point-wise" modality, we have access to the sheaf of real-valued functions on the usual real line, inside of $\mathcal{B}$. In particular, we compare these numeric types internally for differing modalities in Sect. 7.1 and give their $\mathcal{B}$-semantics in Sect. 7.2. This section relies on technical work from Appendix A. Perhaps most interestingly, in Sect. 7.3 we internally define the derivative of such a real-valued function with respect to $t : \mathtt{Time}$ and prove that this definition is linear and satisfies the Leibniz rule. We also prove externally that its $\mathcal{B}$-semantics is that of derivatives in the usual sense.

The main body of the book concludes with Chap. 8, where we discuss several applications of the work. For example, we give an embedding of discrete, continuous, and hybrid dynamical systems into our temporal type theory. We also explain delays, and our general perspective on behavior contracts for interconnected systems. One of our main inspirations for this work was a case study involving safe separation in the National Airspace System; this is discussed in Sect. 8.5. Finally in Sect. 8.6, we discuss the relationship between our higher-order temporal logic and some of the better known temporal logics from the literature.

The book also has two appendices. In Appendix A, we define a technical tool—which we call predomains—by which to reduce the complexity of domains and the morphisms between them. We explain how our constructions work relative to arbitrary modalities, i.e., within arbitrary subtoposes. Finally, in Appendix B we prove that $\mathbb{IR}_{/\triangleright}$ is a continuous category in the sense of Johnstone and Joyal.

Chapter 2
The Interval Domain

In this chapter, we will introduce the *interval domain* $\mathbb{IR}$, which is a topological space that represents the line of time in our work to come. The points of this space can be thought of as compact intervals $[a, b]$ in $\mathbb{R}$. The specialization order on points gives $\mathbb{IR}$ a non-trivial poset structure—in fact it is a domain—and as such it is far from Hausdorff.

The topos of sheaves on the space $\mathbb{IR}$ will play a very important role throughout the book, so we begin in Sect. 2.1 with a review of sheaves on topological spaces, or more generally on posets equipped with a coverage. In Sect. 2.2, we review the theory of domains, and we define $\mathbb{IR}$ in Sect. 2.3. In Sect. 2.4, we show how to view $\mathbb{IR}$ in terms of the usual Euclidean upper half-plane. In Section 2.5 we discuss Grothendieck posites, which allow us to prove the equivalence between four different formulations of the topos of sheaves on $\mathbb{IR}$.

There are a few places in this chapter where we refer to predomains, which are fairly technical and are the subject of Appendix A. However, none of that material is necessary to understand the present chapter. It will become more important for the technical results about arithmetic between real number objects in various subtoposes, which we will discuss later in Sect. 4.3. For the time being, we suggest the reader look briefly at the predomain material if the interest arises, but otherwise feel free to skim it or take it on faith.

2.1 Review of Posites and (0, 1)-Sheaves

The material in this section is largely taken from [nLabb]. For any poset $(P, \leq)$ and $p \in P$, the down-set $\downarrow p \coloneqq \{p' \in P \mid p' \leq p\}$ will play a fundamental role throughout this section. For a set S, let $P(S) \coloneqq \{V \mid V \subseteq S\}$ denote its power set. Given $V \subseteq S$, we write $\downarrow V$ to denote the set $\bigcup_{v \in V} \downarrow v$.

P. Schultz, D. I. Spivak, *Temporal Type Theory*, Progress in Computer Science and Applied Logic 29, https://doi.org/10.1007/978-3-030-00704-1_2