Topos Semantics for a Higher-order Temporal Logic of Actions

Philip Johnson-Freyd, Jon Aytac, and Geoffrey Hulette

Sandia National Laboratories

TLA is a popular temporal logic for writing stuttering-invariant specifications of digital systems. However, TLA lacks higher-order features useful for specifying modern software written in higher-order programming languages. We use categorical techniques to recast a real-time semantics for TLA in terms of the actions of a group of time dilations, or "stutters," and an extension by a monoid incorporating delays, or "falters." Via the geometric morphism of the associated presheaf topoi induced by the inclusion of stutters into falters, we construct the first model of a higher-order TLA.

Philip Johnson-Freyd: pajohn@sandia.gov
Jon Aytac: jmaytac@sandia.gov
Geoffrey Hulette: ghulett@sandia.gov,
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (NNSA) under contract DE-NA0003525. This work was funded by NNSA's Advanced Simulation and Computing (ASC) Program. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

1 Introduction

The Temporal Logic of Actions (TLA) is a temporal logic commonly used for specifying digital computer systems [11, 13]. TLA formulae are linear temporal properties *invariant under "stuttering."* Stuttering invariant specifications written as TLA formulae are easily composed, using nothing more than conjunction, with no implicit assumptions about synchronization. Stuttering invariance also leads to a simple but powerful notion of "refinement," that is, showing that a detailed specification implements an abstract one.

In [11] Lamport presents TLA as a first-order logic, but, in specifications, higher-order features are often desirable. For example, one would often like to prove a rule of inference that works over *all* propositions or *all* predicates. Lamport must introduce special syntax (e.g., for fairness) where in a higher-order context these language features could be replaced with simple functions on propositions. Moreover, programmers today often work in higher-order programming languages and the powerful abstraction features in these languages (e.g., a generalized "map" function) are not easily expressed in TLA specifications.

As a step towards the goal of defining a higher-order TLA, we present *a* model in which it could be interpreted. In standard linear temporal logics, which do not feature stuttering invariance, higher-order features can be modeled in the so-called "Topos of Trees" (i.e., presheaves over ω) [5, 15]. Another impressive line of work on "Temporal Types" takes a topos theoretic approach based on translation invariant sheaves (using the additive structure of $\mathbb{R}$) [16]. Unfortunately, these models cannot capture TLA's stuttering invariance.

Our categorical model of higher-order TLA meets several desiderata, motivated by the observations above:

1. it should provide a model of higher-order classical S4 (TLA is a special case of this modal logic);
2. it should have a "temporal" interpretation which accounts for stuttering invariance;
3. it should correspond with an equivalent notion of validity, in the first-order subset, to the standard semantics of TLA.

We believe our model to be the first that is suitable for a higher-order TLA. It is constructed as follows. First, we switch perspective, from the standard discrete-time semantics of TLA to an alternative real-time semantics found in the literature [9] and reviewed in Section 2. Then, recalling that models for higher-order modal logic can be generated by geometric morphisms between topoi (Section 3), we construct our model by recasting the real-valued semantics by way of such a geometric morphism (Section 4). Our key insight was to consider stutterings as a group, leading to a generalization of stuttering, which we call "faltering."

2 The Temporal Logic of Action

Like Pnueli's Linear Temporal Logic (LTL) [14], TLA adopts the perspective of linear time: formulae classify sets of (linear) infinite traces of a system evolving through time. Also like LTL, TLA has temporal modalities "always" ($\Box$) and "eventually" ($\Diamond$). However, unlike LTL, TLA has no "next" ($\circ$) modality. Instead, TLA has a notion of "actions" that describe instantaneous changes in the system state, but which also allow "stuttering steps" in which the trace evolves in time but the state remains unchanged. Thus, unlike LTL, TLA formulae are always "stuttering invariant," that is, they cannot differentiate traces by how long they stutter.

Syntactically, TLA has two classes of formulae (Figure 1): actions, which denote instantaneous changes to the system state, and temporal formulae, which are predicates on traces.

Actions are normal first-order logic formulae except in the handling of terms. Variables appearing in terms can be "rigid" (written in italics), indicating that they do not change over time, or "flexible" (written in bold face), indicating that they may.

$$
\begin{aligned}
E \in \text{Terms} &::= x \mid \mathbf{x} \mid \mathbf{x}' \mid f(E_1, \ldots, E_n) & \exists x.T &\triangleq \neg\forall x.\neg T \\
A \in \text{Actions} &::= R(E_1, \ldots, E_n) \mid E_1 = E_2 \mid \forall x.A \mid A_1 \wedge A_2 \mid \neg A & \boldsymbol{\exists}\mathbf{x}.P &\triangleq \neg\boldsymbol{\forall}\mathbf{x}.\neg P \\
P \in \text{Propositions} &::= \Box P \mid \neg P \mid P_1 \wedge P_2 \mid \Box[A]_{<\mathbf{x_1}, \ldots, \mathbf{x_n}>} \mid \forall x.P \mid \boldsymbol{\forall}\mathbf{x}.P & T_1 \vee T_2 &\triangleq \neg(T_1 \wedge T_2) \\
T \in \text{Formulae} &::= A \mid P & T_1 \Rightarrow T_2 &\triangleq \neg T_1 \vee T_2
\end{aligned}
$$

Figure 1: Syntax and Syntactic Sugar of TLA

Flexible variables may appear primed ($\mathbf{x}'$) or unprimed ($\mathbf{x}$) denoting the variable's value in the next or current state, respectively.

Temporal formulae are comprised of the usual propositional connectives and temporal quantifiers, along with a special operator $\Box[A]_v$, where A is an action and v is a function on the system state). Intuitively, the formula $\Box[A]_v$ means "it is always the case that either the action A happens or v does not change." TLA is also equipped with ordinary (first-order) quantifiers over rigid variables $\forall x.P$ as well as "temporal" quantifiers over flexible variables $\boldsymbol{\forall}\mathbf{x}.P$.

Lamport's semantics for TLA (Figure 2) interprets temporal formulae using a discrete model of time. Traces are modeled as functions from natural numbers to a "state," where states are assignments of values for each flexible variable.

Lamport's semantics are unusual in the handling of the flexible quantifier ($\boldsymbol{\forall}$). Naïvely, flexible quantification would be

$$
\theta, \rho \models \boldsymbol{\forall}\mathbf{x}.P \quad \text{iff} \quad \text{for every } d \in \mathcal{D}^{\mathbb{N}}, \\
\theta, \rho \uplus (\mathbf{x} \mapsto d) \models P
$$

Definition 1 (Discrete Stuttering Equivalence)**.** Given any set S, the stuttering equivalence relation $\approx$ on behaviours $S^{\mathbb{N}}$ is the least equivalence relation such that for every $\rho \in S^{\mathbb{N}}$ and $n \in \mathbb{N}$ we have $\rho \approx \rho'$ when ρ' is given by

$$
\begin{aligned}
\rho'(m) &= \rho(m) && \text{when } m \leq n \\
\rho'(m) &= \rho(m-1) && \text{when } m > n.
\end{aligned}
$$

Unfortunately, in this semantics, the definition of flexible quantification must explicitly "bake in" stuttering invariance and this makes flexible quantification behave quite differently from the ordinary semantics of modal logic.

Proposition 1 (Stuttering Equivalence of TLA)**.** For any P, θ, ρ, ρ' such that $\rho \approx \rho'$

$$
\theta, \rho \models P \text{ iff } \theta, \rho' \models P.
$$

Kaminski and Yariv [9] proposed an alternative semantics for TLA based on a continuous notion of time. In this setting traces are interpreted as "non-Zeno" functions from the non-negative real numbers.

Definition 2 (Non-Zeno function)**.** A non-Zeno function over a set S is a function f from non-negative real numbers to S such that

1. for every $t \in \mathbb{R}_{\geq 0}$ there exists a positive ϵ such that for all t' where $t \leq t' < t + \epsilon$ we have $f(t) = f(t')$ and

$$
\begin{array}{lll}
\theta, \sigma, \sigma' \models A_1 \wedge A_2 & \text{iff } (\theta, \sigma, \sigma' \models A_1) \text{ and } (\theta, \sigma, \sigma' \models A_1) & \\
\theta, \sigma, \sigma' \models \neg A & \text{iff } \theta, \sigma, \sigma' \not\models A & [\![x]\!](\theta, \sigma, \sigma') = \theta(x) \\
\theta, \sigma, \sigma' \models E_1 = E_2 & \text{iff } [\![E_1]\!](\theta, \sigma, \sigma') = [\![E_2]\!](\theta, \sigma, \sigma') & [\![\mathbf{x}]\!](\theta, \sigma, \sigma') = \sigma(x) \\
\theta, \sigma, \sigma' \models \forall x.A & \text{iff for every } v \in \mathcal{D}\ (\theta \uplus x \mapsto v), \sigma, \sigma' \models A & [\![\mathbf{x}']\!](\theta, \sigma, \sigma') = \sigma'(x) \\
\theta, \sigma, \sigma' \models R(E_1, \ldots, E_n) & \text{iff } \mathcal{R}(R)([\![E_1]\!](\theta, \sigma, \sigma'), \ldots, [\![E_n]\!](\theta, \sigma, \sigma')) &
\end{array}
$$

$$
[\![f(E_1, \ldots, E_n)]\!](\theta, \sigma, \sigma') = \mathcal{F}(f)([\![E_1]\!](\theta, \sigma, \sigma'), \ldots, [\![E_1]\!](\theta, \sigma, \sigma'))
$$

$$
\begin{array}{ll}
\theta, \rho \models P_1 \wedge P_2 & \text{iff } \theta, \rho \models P_1 \text{ and } \theta, \rho \models P_2 \\
\theta, \rho \models \neg P & \text{iff } \theta, \rho \not\models P \\
\theta, \rho \models \Box P & \text{iff for every } n \in \mathbb{N}\ \theta, \rho[n, \ldots] \models P \\
\theta, \rho \models \Box[A]_{<\mathbf{x_1}, \ldots, \mathbf{x_m}>} & \text{iff for each } n \in \mathbb{N} \text{ either } \theta, \rho[n], \rho[n+1] \models A \text{ or } \forall i \in [1, m].\rho[n](\mathbf{x_i}) \\
\theta, \rho \models \forall x.P & \text{iff for every } v \in \mathcal{D}\ (\theta \uplus x \mapsto v), \rho \models P \\
\theta, \rho \models \boldsymbol{\forall}\mathbf{x}.P & \text{iff for every } d \in \mathcal{D}^{\mathbf{N}} \text{ and } \rho' \approx \rho, \theta, \rho' \uplus (\mathbf{x} \mapsto d) \models P
\end{array}
$$

Figure 2: Discrete Time Semantics of TLA

2. there is no bounded increasing sequence $t_0, t_1, t_2, \ldots$ such that forall i, $f(t_i) \neq f(t_{i+1})$.

These two conditions ensure that a non-Zeno function does not change too quickly: the first condition guarantees that each state is held for positive time, while the second ensures that only a finite number of states are visited in any finite length of time. We (ab)use the notation $S^{\mathbb{R}^+}$ to refer to the set of non-Zeno functions over S.

Stuttering invariance of a set of such non-Zeno functions is modeled as closure under pre-composition by homeomorphisms on $\mathbb{R}_{\geq 0}$ (with the standard topology). The alternative continuous semantics (Figure 3) yields exactly the same notion of truth as Lamport's original semantics, while avoiding the need to "bake in" stuttering invariance in its definitions.

This continuous semantics clarifies many aspects of TLA. It explains stuttering invariance as invariance under time dilation. Furthermore, it presents rigid and flexible variables uniformly, allowing them to be viewed as coming from two different types. Categorically, this means rigid and flexible quantification should correspond to quantification over different objects.

3 Semantics of Higher-order Logic

Higher-order Logic (HOL) (see [2]) combines a (possibly intuitionistic) logic with the simply-typed λ-calculus. It may be viewed as an extension to multi-sorted first-order logic that adds features for quantifying over function types and

$$
\begin{aligned}
&next(\tau, S) \triangleq 0 && \text{when } \forall t \in \mathbb{R}_{\geq 0}, \forall x \in S, \tau(0)(x) = \tau(t)(x)\\
&next(\tau, S) \triangleq \mathbf{sup}\{r \mid \forall 0 \leq k \leq r, \forall x \in S, \tau(0)(x) = \tau(k)(x)\} && \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
\theta, \tau &\models_{\mathbb{R}} \Box[A]_{\mathbf{x}_1,\ldots,\mathbf{x}_n} && \text{iff } r = 0 \text{ or } \theta, \tau(0), \tau(r) \models A\\
& && \quad \text{where } r = next(\tau, \{\mathbf{x}_i | 0 \leq i \leq n\})\\
\theta, \tau &\models_{\mathbb{R}} T_1 \wedge T_2 && \text{iff } \theta, \tau \models_{\mathbb{R}} T_1 \text{ and } \theta, \tau \models_{\mathbb{R}} T_2\\
\theta, \tau &\models_{\mathbb{R}} \neg T && \text{iff } \theta, \tau \not\models_{\mathbb{R}} T\\
\theta, \tau &\models_{\mathbb{R}} \forall x.T && \text{iff for every } v \in \mathcal{D} \text{ we have } (\theta, x \mapsto v), \tau \models_{\mathbb{R}} T\\
\theta, \tau &\models_{\mathbb{R}} \boldsymbol{\forall}\mathbf{x}.T && \text{iff for every } v \in \mathcal{D}^{\mathbb{R}^+} \text{ we have } \theta, (x \mapsto (\tau(r), \mathbf{x} \mapsto v(r))) \models_{\mathbb{R}} T\\
\theta, \tau &\models_{\mathbb{R}} \Box T && \text{iff for every } k \in \mathbb{R}_{\geq 0} \text{ such that } \theta, \tau[k..] \models_{\mathbb{R}} T
\end{aligned}
$$

Figure 3: Continuous-time Semantics of TLA

propositions.

Modal variants of higher-order logic are usually formed simply by adding additional modal operators exactly as one would in a propositional logic.

There are many semantics for higher-order logic. In the "standard" semantics, types are interpreted as sets, function types are interpreted as the set of all functions between their constituents, and propositions are interpreted as booleans. This model is incomplete, however.

A more general class of model is found in topoi.

Definition 3. A topos is a cartesion closed category $\mathcal{E}$ possesing all finite limits and a subobject classifier, i.e. an object Ω and a monic arrow $\mathsf{true} : 1 \to \Omega$ such that $\forall$ monic $m : S \to B$ $\exists! \phi_m : B \to \Omega$ such that
$$
\begin{tikzcd}
S \arrow[r, "!"] \arrow[d, hook, "m"'] & 1 \arrow[d, "\mathsf{true}"] \\
B \arrow[r, dashed, "\phi_m"'] & \Omega
\end{tikzcd}
$$
is a pullback.

In the naïve topos semantics, types (also, contexts) are interpreted as objects, terms are interpreted as morphisms, function types are interpreted by way of the inner hom, and the proposition type is interpreted as the subobject classifier.

However, this topos semantics is still too strong—it justifies additional laws which are not derivable from the natural deduction rules in Figure 4. In particular, the topos semantics imposes upon higher-order logic the additional property of extensionality of entailment (see [4] 5.3.7)

$$
\frac{\Gamma \vdash P, Q : \sigma \to \mathrm{Prop} \qquad \Gamma, x : \sigma \mid \Theta, Px \vdash Qx \qquad \Gamma, x : \sigma \mid \Theta, Qx \vdash Px}{\Gamma \mid \Theta \vdash P =_{\sigma \to \mathrm{Prop}} Q}
$$
.

A class of categorical models for higher-order logic with more examples is obtained by weakening the structure involved in the subobject classifier.

Definition 4 (Hyperdoctrine)**.** Let $\mathsf{P} : \mathsf{C}^{op} \to \mathsf{HeyAlg}$ be a functor from a cartesion closed C

into the category of Heyting algbras such that:

1. $\forall X, Y : \mathrm{Obj}\mathsf{C}$ there are monotone $\exists^X_Y, \forall^X_Y : \mathrm{Hom}_{\mathsf{PreOrd}}(\mathsf{P}(X \times Y), \mathsf{P}(Y))$ such that for $\pi : X \times Y \to Y$ the projection $\exists^X_Y \dashv \mathsf{P}(\pi) \dashv \forall^X_Y$ and satisfying the Beck-Chevalley condition
$$\forall f : Y \to Y' \quad \begin{tikzcd} \mathsf{P}(X \times Y') \arrow[r, "\mathsf{P}(\mathrm{id}_X \times f)"] \arrow[d, "\forall^X_{Y'}"'] & \mathsf{P}(X \times Y) \arrow[d, "\forall^X_Y"'] \\ \mathsf{P}Y' \arrow[r, "\mathsf{P}f"'] & \mathsf{P}Y \end{tikzcd}$$
commutes as does the similar $\exists^X_Y$ diagram;

2. $(\mathsf{Forget} \cdot \mathsf{P}) : \mathsf{C}^{op} \to \mathsf{Set}$ is representable.

Hyperdoctrines provide a setting for a sound and complete semantics for HOL by modeling contexts using the underlying cartesian closed category structure, with the Heyting algebra of propositions over those contexts given by the functor, and the quantifiers induced by the adjoints.[1] Moreover, by replacing the category of Heyting algebras with the category of Boolean algebras, we gain a notion of "classical hyperdoctrine," which provides a sound and complete semantics for classical higher-order logic. Finally, using an even stronger category of "modal algebras" yields a model of S4 modal higher-order logic.

Definition 5. A ***modal algebra*** is a pair $(A, \Box) : \mathrm{Obj}(\mathsf{MAlg})$ where A is a Heyting algebra and $\Box$ is a left exact comonad on A.

A ***modal algebra morphism*** $f : (A, \Box) \to (B, \Box')$ is a morphism of the underlying Heyting algebras which commutes with the modalities in the sense that $f \cdot \Box = \Box' \cdot f$.

[1]Completeness, as is often the case, holds for the class of models by constructing an appropriate syntactic object initial in the category of hyperdoctrines as in [10].

Definition 6 (Modal Hyperdoctrine)**.** Let $\mathsf{P} : \mathsf{C}^{op} \to \mathsf{MAlg}$ be a functor from a small cartesian closed category C into the category of Modal algbras MAlg otherwise satisfying the axioms of a hyperdoctrine.

The hyperdoctrine semantics fully generalizes the topos semantics, as every topos T induces a (intuitionistic) hyperdoctrine
$$(T, \mathrm{Hom}_T(-, \Omega)). \tag{1}$$

However, these are not the only hyperdoctrines of interest. Specifically, the only fact about Ω in equation 1 required for the resulting structure to be a (intuitionistic) hyperdoctrine is that it forms an internal complete Heyting algebra in T.

Given any topos $\mathcal{E}$ and internal complete Heyting algebra H in $\mathcal{E}$, there is a natural way of equipping $\mathrm{Hom}_{\mathcal{E}}(-, H)$ with a Heyting algebra structure so that $(\mathcal{E}, \mathrm{Hom}_{\mathcal{E}}(-, H))$ forms a hyperdoctrine.

If H is an internal complete boolean or modal algebra in T, then the resulting hyperdoctrine will be classical or modal, respectively [1].

In this topos-theoretic setting, we can apply a simple recipe for constructing a topos together with internal complete modal algebras. Recall

Definition 7. Let $\mathcal{E}, \mathcal{F}$ be topoi. A ***geometric morphism*** $\mathsf{f} : \mathcal{E} \to \mathcal{F}$ is an adjunction $\begin{tikzcd} \mathcal{E} \arrow[r, shift left, "\mathsf{f}_\star"] \arrow[r, phantom, "\top"] & \mathcal{F} \arrow[l, shift left, "\mathsf{f}^\star"] \end{tikzcd}$ such that the left adjoint $\mathsf{f}^\star$, known as the inverse image, preserves finite limits. If every object $X : \mathrm{Obj}(\mathcal{E})$ is a subquotient of an object of the inverse image $\mathsf{f}^\star$, so that there exists $Y : \mathrm{Obj}(\mathcal{F})$ and diagram $f^\star(Y) \leftarrowtail S \twoheadrightarrow X$, then f is ***localic***.

Geometric morphisms are a source of internal complete Heyting algebras.

Proposition 2. Let $f : \mathcal{E} \to \mathcal{F}$ a geometric morphism. Then $f_\star(\Omega_\mathcal{E})$ is a complete Heyting algebra internal to $\mathcal{F}$.

Geometric morphisms are also a source of adjoint pairs of maps of complete Heyting algebras.

Lemma 1 ([8] C1.3)**.** In any topos $\mathcal{E}$, the subobject classifier $\Omega_\mathcal{E}$ is the initial complete Heyting algebra object. That is, for all complete Heyting algebras H internal to $\mathcal{E}$, there is a unique map of complete Heyting algebras $i : \Omega_\mathcal{E} \to H$. Moreover, the right adjoint of τ is the classifying map of the top element $\top_H : 1 \to H$.

This adjoint pair of maps defines a useful comonad.

Lemma 2 ([1])**.** Given a complete Heyting algebra H internal to topos $\mathcal{E}$, let $i \vdash \tau$ the canonical adjunction $i : \Omega_\mathcal{E} \rightleftarrows H : \tau$. The composite $i \circ \tau$ is an S4 modality on H.

If we have *two* topoi, $\mathcal{E}$ and $\mathcal{F}$, and a geometric morphism $f : \mathcal{E} \to \mathcal{F}$ then the image of the subobject classifier of $\mathcal{E}$ in $\mathcal{F}$ is an internal complete modal algebra in $\mathcal{F}$.

An illustrative example is given by a topos-theoretic view of Kripke semantics. Let K be a preorder, interpreted as a collection of "possible worlds," together with an accessibility relation. By $|K|$ we mean the discrete category with the same underlying objects as K.

The inclusion $|K| \to K$ induces a geometric morphism $f : Psh(|K|) \to Psh(K)$.

Lemma 3 ([7], prop. 3.1)**.** Let $f : \mathsf{D} \to \mathsf{C}$ be a functor of small categories. If f is faithful, then the induced geometric morphism $Psh(\mathsf{D}) \to Psh(\mathsf{C})$ is localic.

Thus we obtain a modal hyperdoctrine on $(Psh(K), \mathrm{Hom}_{Psh(K)}(-, f_\star(\Omega_{Psh(|K|)}))$. In particular, as $|K|$ is a groupoid, $\mathcal{E} = Psh(|K|)$ is a Boolean topos, so $f_\star(\Omega_\mathcal{E})$ is not only a complete Heyting algebra internal to $\mathcal{F} = Psh(K)$, it is an internal Boolean algebra! The resulting logic is classical, even though $Psh(K)$ is very much not a boolean topos in general (it is, instead, a Kripke model of an intuitionistic logic). The internal logic of this modal hyperdoctrine is, in the first-order fragment, exactly what we would get from the Kripke semantics over K. And thus we have a simple presentation of a higher-order version of that semantics.

4 The Model

Now we are ready to construct a candidate model for a Higher-order TLA.

Why not simply use the topos-theoretic Kripke semantics, described in Section 3, applied to the discrete semantics? This approach will fail because TLA's discrete semantics is *not* an ordinary Kripke semantics, since flexible quantification is not ordinary Kripke quantification (see Section 2). Even the continuous semantics is not adequately captured in the ordinary, preorder-based, Kripke view since Kripke does not account for stuttering.

We must build a model that includes

stuttering invariance from the get go. Preorders are inadequate to this task. Luckily, the geometric morphism construction described in Section 3 is not specific to Kripke's inclusion of a discrete set into a preorder. *Any* faithful functor between small categories whose domain is a groupoid induces a model of classical higher-order modal logic.

Our model is enabled by the following elementry observation: in the continuous semantics, stuttering invariance is precisely closure under the action the group of stutters.

Definition 8 (Stutter)**.** A stutter is a continuous function $\mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with continuous inverse.

By $\mathcal{S}$ we denote the group of stutters

$$\mathcal{S} = (\{f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \mid f \text{ is a stutter}\}, \cdot, id_{\mathbb{R}_{\geq 0}})$$

.

We will adopt the convention of viewing any monoid G as the category $\mathsf{B}G$ with one object and one monoid's worth of morphisms. This way the category of G-sets and G-set morphisms for a group G (more generally, for any monoid) is just $Psh(\mathsf{B}G)$.

Non-Zeno functions on a set form a $\mathcal{S}$-set where the action of $\mathcal{S}$ is precomposition. Stuttering invariant subsets of that set are then, exactly, sub $\mathcal{S}$-sets. As such, the category of $\mathcal{S}$-sets ($Psh(\mathsf{B}\mathcal{S})$) seems to be closely connected to our problem. Since $\mathcal{S}$ is a group $\mathsf{B}\mathcal{S}$ is a groupoid and the presheaf topos $Psh(\mathsf{B}\mathcal{S})$ is boolean. Therefore, it is a tempting target for the semantics of a higher-order TLA. We already know this will not work on its own though, as a topos is not enough to interpret the modalities. The most important modality for our purposes is $\square$. A behavior (viewed as a non-Zeno function) is always a member of some set of behaviors if, given any initial delay in which the behavior is not observed, the remainder is in that set. Thus, while stuttering invariance has to do with closure under dilation of time by bicontinuous functions, $\square$ has to do with the translation of time.

To that end, we introduce a generalization of stutters, which we call "falters," which can include translation as well as dilation.

Definition 9. A falter is a monotone function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that the function $x \mapsto f(x) - f(0)$ is a stutter.

By $\mathcal{F}$ we denote the monoid of falters (under function composition).

There is a natural morphism of monoids $\iota : \mathcal{S} \to \mathcal{F}$ given by inclusion, inducing a faithful functor $\iota : \mathsf{B}\mathcal{S} \to \mathsf{B}\mathcal{F}$. As mentioned in Section 3, such a faithful functor induces a localic geometric morphism on the associated presheaf categories $\iota^\star \dashv \iota_\star : Psh(\mathsf{B}\mathcal{S}) \rightleftarrows Psh(\mathsf{B}\mathcal{F})$. Our proposed model for a higher-order TLA is the hyperdoctrine induced by this geometric morphism.

We will now elaborate some details of this model. We consider $\mathcal{F}$-sets to be "temporal types" as these are the types about which we can talk in our model. The type of flexible variables over some base set are computed according to the functor

$$\mathrm{Flex} : \mathrm{Set} \to Psh(\mathsf{B}\mathcal{F})$$
$$\mathrm{Flex}(S) = (\{f : \mathbb{R}_{\geq 0} \to S \mid f \text{ non zeno}\}, \cdot)$$

While the type of rigid variables over a base set is computed according to the functor

$$\mathrm{Rigid} : \mathrm{Set} \to Psh(\mathsf{B}\mathcal{F})$$
$$\mathrm{Rigid}(S) = (S, ((\_, x) \mapsto x))$$

There is a natural inclusion morphism from `Rigid` $\to$ `Flex`, which is (for every set) monic. However, `Rigid`(S) is not the only subobject of `Flex`(S). Any stuttering- and translation-closed subset of behaviors will be interpretable as a temporal set. Of course, these are not the only temporal types: the inner hom between types of flexible variables, for instance, corresponds to temporal processes rather than flexible variables over functions of the underlying sets.

In Section 3 we reviewed the fact that a modal hyperdoctrine may be represented by applying the inverse image part of the geometric morphism to the subobject classifier in $Psh(\mathsf{B}\mathcal{S})$. As $\mathcal{S}$ is a group, it has only two ideals, $\emptyset$ and $\mathcal{S}$. Thus, $\Omega_{Psh(\mathsf{B}\mathcal{S})}$ is the set 2 with the trivial $\mathcal{S}$-action.

As presheaf categories have all (co)limits, the inverse image part of the geometric morphism may be computed as a right Kan extension. As our categories $\mathsf{B}\mathcal{S}$ and $\mathsf{B}\mathcal{F}$ have singleton objects, this can be computed pointwise. Given $\mathsf{F} : \mathsf{Set}^{\mathsf{B}\mathcal{S}}$, we compute $\lim\left(\bullet_{\mathcal{S}} \downarrow \iota \xrightarrow{\pi^{\bullet_{\mathcal{S}}}} \mathsf{B}\mathcal{S} \xrightarrow{\mathsf{F}} \mathsf{Set}\right)$, which amounts to equalizing away the stutter action

$$\textstyle\prod_{\mathcal{S}\backslash\mathcal{F}} \mathsf{F}(\bullet_{\mathcal{S}}) \rightarrowtail \prod_{\mathcal{F}} \mathsf{F}(\bullet_{\mathcal{S}}) \rightrightarrows \prod_{\mathcal{S}\times\mathcal{F}} \mathsf{F}(\bullet_{\mathcal{S}}) \ .$$

On $Psh(\mathsf{B}\mathcal{S})$'s subobject classifier, this is

$$\begin{aligned}\mathrm{Prop} &\triangleq \iota_\star(\Omega_{Psh(\mathsf{B}\mathcal{S})}) \\ &= (\{p : \mathcal{F} \to 2 \mid \forall m \in \mathcal{S}, n \in \mathcal{F}, p(n) = p(nm)\} \\ &\quad , \quad (n, p) \mapsto (r' \mapsto p(n \cdot n'))) \\ &\cong (\mathcal{P}(\mathbb{R}_{\geq 0}), (n, O) \mapsto \mathrm{im}^{-1}(n)(O))\end{aligned}$$

Consequently (and pleasingly), in our model, a proposition corresponds to the set of times when that proposition is true.

All the usual connectives coming from the boolean algebra structure are computed pointwise. All that remains is to compute the modal structure. The subobject classifier in $Psh(\mathsf{B}\mathcal{F})$ is the collection of falter ideals

$$\Omega_{Psh(\mathsf{B}\mathcal{F})} = \{I \subseteq \mathcal{F} \mid \forall i \in I \; \forall f \in \mathcal{F}.\ i \cdot f \in I\},$$

but these are just all upward-closed subsets of $\mathbb{R}_{\geq 0}$, so $\Omega_{Psh(\mathsf{B}\mathcal{F})} \cong \langle \mathcal{P}_{\uparrow}(\mathbb{R}_{\geq 0}), (n, O) \mapsto \mathrm{im}^{-1}(n)(O)\rangle$. As subobject classifier in $Psh(\mathsf{B}\mathcal{F})$, $\Omega_{Psh(\mathsf{B}\mathcal{F})}$ is initial in complete Heyting algebras internal to $\mathcal{F}$, so the obvious equivariant inclusion $i_\Omega : \Omega_{Psh(\mathsf{B}\mathcal{F})} \hookrightarrow \iota_\star(\Omega_{Psh(\mathsf{B}\mathcal{S})})$ is essentially unique. The right adjoint $\tau_\Omega : \iota_\star(\Omega_{Psh(\mathsf{B}\mathcal{S})}) \to \Omega_{Psh(\mathsf{B}\mathcal{F})}$, which classifies $1 \to \iota_\star(\Omega_{Psh(\mathsf{B}\mathcal{S})})$, is, then, the upward closure $\uparrow(-) : \mathcal{P}(\mathbb{R}) \to \mathcal{P}_{\uparrow}(\mathbb{R})$. The adjunction $\Box := i_\Omega \circ \tau_\Omega : \mathrm{End}(\iota_\star \Omega_{Psh(\mathsf{B}\mathcal{S})})$ provides a left exact comonad on the complete internal Heyting algebra $\iota_\star(\Omega_{Psh(\mathsf{B}\mathcal{S})})$.

The resulting modal structure is quite natural – it reduces to ensuring that a proposition holds at all future times

$$\Box(-) : \mathrm{Prop} \to_{Psh(\mathsf{B}\mathcal{F})} \mathrm{Prop}$$
$$\Box(S) = \{r \in \mathbb{R}_{\geq 0} \mid \forall r' \geq r, r' \in S\}.$$

As such, our categorical model is precisely a higher order generalization of the continuous-time semantics presented in Section 2.

Theorem 1. The modal hyperdoctrine $(Psh(\mathsf{B}\mathcal{F}), \mathrm{Hom}(-, \iota_\star(\Omega_{\mathsf{B}\mathcal{S}})))$ admits a sound interpretation of higher-order classical S4. Moreover, restricting to the first-order fragment, this model corresponds to the model of the Temporal Logic of Actions in Figure 3 and agrees for validity with the standard semantics (Figure 2).

5 Conclusion

We have found a categorical setting in which to model a higher-order version of TLA, providing a way of assigning meaning to statements in this logic. This a first step towards a useful higher-order temporal logic for digital systems. In particular, the model we have described will allow us to formulate proof rules and verify that they are sound with respect to our model. We imagine that other models for such a proof theory may also be of interest.

Our model construction started by switching from the discrete-time semantics for TLA that was originally formulated by Lamport to a real-time semantics. This was essential, since stuttering invariance does not correspond to closure under a group action in the discrete case. In Lamport's semantics, stuttering forms a monoid (at best) rather than a group, and closure under the action of that monoid fails to fully account for stuttering invariance. Nonetheless, a categorical semantics of higher-order TLA based on discrete-time stuttering invariance remains an intriguing challenge.

We plan to continue our work on a higher-order TLA, with the goal of using it as the basis of a proof assistant and toolchain for practical engineering purposes. Yet significant challenges remain, such as developing the required syntax, proof theory, and so on. Moreover, it remains to be seen how extending TLA with higher-order features can be put into useful practice. A potential use case would be to specify a variant of PlusCal [12], a programming language that translates to TLA, then extending it with handy higher-order features such as closures or objects.

Our goal in this paper was to find *a* model satisfying our desiderata. It remains to state what, exactly, "higher-order TLA" is and to specify its class of models. In the present paper we focused on giving an account of the temporal types, neglecting the underlying non-temporal sets. A detailed and generalized account of the categorical properties of TLA's action lifting construction will necessarily be needed in future work. All that said, the particular form of the model we found is intriguing. Because the underlying category of our hyperdoctrine is a topos, and not just cartesion closed, it has all finite limits. As such, it is a promising setting for developing an account of specification composition using pullbacks [3, 6].

References

[1] S. Awodey, K. Kishida, and H.-C. Kotzsch. Topos Semantics for Higher-Order Modal Logic. *ArXiv*, February 2014.

[2] Alonzo Church. A formulation of the simple theory of types. *The journal of symbolic logic*, 5(2):56–68, 1940.

[3] Joseph A Goguen. A categorical manifesto. *Mathematical structures in computer science*, 1(01):49–67, 1991.

[4] Bart Jacobs. *Categorical logic and type theory*, volume 141. Elsevier, 1999.

[5] Wolfgang Jeltsch. Towards a common categorical semantics for linear-time temporal logic and functional reactive programming. *Electron. Notes Theor. C. S.*, 286:229–242, September 2012.

[6] Philip Johnson-Freyd, Geoffrey C. Hulette, and Zena M. Ariola. Verification by way of refinement: A case study in the use of coq and TLA in the design of a safety critical system. In *FMICS-AVoCS*, volume 9933 of *LNCS*, pages 205–213. Springer, 2016.

[7] Peter T Johnstone. Factorization theorems for geometric morphisms, i. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 22(1):3–17, 1981.

[8] Peter T Johnstone. *Sketches of an elephant: A topos theory compendium-volume 1*. Oxford University Press, 2002.

[9] Michael Kaminski and Yael Yariv. A real-time semantics of temporal logic of actions. *J. Log. Comput.*, 13(6):921–937, 2003.

[10] Joachim Lambek and Philip J Scott. Reflections on the categorical foundations of mathematics. In *Foundational Theories of Classical and Constructive Mathematics*, pages 171–186. Springer, 2011.

[11] Leslie Lamport. The Temporal Logic of Actions. *ACM Trans. Program. Lang. Syst.*, 16(3):872–923, May 1994.

[12] Leslie Lamport. The PlusCal algorithm language. In *International Colloquium on Theoretical Aspects of Computing*, pages 36–60. Springer, 2009.

[13] Chris Newcombe. Why Amazon chose TLA+. In Yamine Ait Ameur and Klaus-Dieter Schewe, editors, *Abstract State Machines, Alloy, B, TLA, VDM, and Z*, pages 25–39, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[14] Amir Pnueli. The temporal logic of programs. In *Symposium on Foundations of Computer Science*, pages 46–57. IEEE, 1977.

[15] Colin Riba and Guilhem Jaber. Modal Logic of Transition Systems in the Topos of Trees. working paper, October 2017.

[16] Patrick Schultz and David I. Spivak. Temporal type theory: A topos-theoretic approach to systems and behavior, 2017.

A Rules of Higher-order logic

$$T, S \in \text{Types} ::= \ldots \mid T \to S \mid \text{Prop} \qquad M, N, O \in \text{Terms} \quad ::= \ldots \mid x \mid \lambda(x : T).M \mid M\ N \mid (\Rightarrow) \mid \forall_T$$

$$M \Rightarrow N \triangleq (\Rightarrow)\ M\ N \qquad \neg M \triangleq M \Rightarrow \bot$$

$$\forall(x : T).M \triangleq \forall_T(\lambda(x : T).M) \qquad M \wedge N \triangleq \forall(p : \text{Prop}).(M \Rightarrow N \Rightarrow p) \Rightarrow p$$

$$\bot \triangleq \forall(p : \text{Prop}).p \qquad M \vee N \triangleq \forall(p : \text{Prop}).(M \Rightarrow p) \Rightarrow (N \Rightarrow p) \Rightarrow p$$

$$\top \triangleq \forall(p : \text{Prop}).p \Rightarrow p \qquad \exists(x : T).M \triangleq \forall(p : \text{Prop}).(\forall(x : T).M \Rightarrow p) \Rightarrow p$$

$$\frac{\Gamma \vdash M \equiv N : T}{\Gamma \vdash N \equiv M : T} \qquad \frac{\Gamma \vdash M \equiv N \qquad \Gamma \vdash N \equiv O}{\Gamma \vdash M \equiv O} \qquad \frac{(x : T) \in \Gamma}{\Gamma \vdash x \equiv x : T}$$

$$\frac{\Gamma \vdash M_1 \equiv M_2 : S \to T \qquad \Gamma \vdash N_1 \equiv N_2 : S}{\Gamma \vdash M_1\ N_1 \equiv M_2\ N_2 : T} \qquad \frac{\Gamma, x : T \vdash M \equiv N : S}{\Gamma \vdash \lambda(x : T).M \equiv \lambda(x : T).N : T \to S}$$

$$\frac{\Gamma, x : T \vdash M \equiv M : S \qquad \Gamma \vdash N \equiv N : T}{\Gamma \vdash (\lambda(x : T).M)\ N \equiv M[N/x] : S} \qquad \frac{\Gamma, x : T \vdash M\ x \equiv N\ x : S}{\Gamma \vdash M \equiv N : T \to S}$$

$$\frac{}{\Gamma \vdash (\Rightarrow) \equiv (\Rightarrow) : \text{Prop} \to \text{Prop} \to \text{Prop}} \qquad \frac{}{\Gamma \vdash \forall_T \equiv \forall_T : (T \to \text{Prop}) \to \text{Prop}}$$

$$\frac{}{\Gamma \mid \emptyset \vdash \mathbf{wf}} \qquad \frac{\Gamma \mid \Theta \vdash \mathbf{wf} \qquad \Gamma \vdash M \equiv M : \text{Prop}}{\Gamma \mid \Theta, M \vdash \mathbf{wf}} \qquad \frac{M \in \Theta \qquad \Gamma \mid \Theta \vdash \mathbf{wf}}{\Gamma \mid \Theta \vdash M\ \mathbf{true}}$$

$$\frac{\Gamma \mid \Theta \vdash M\ \mathbf{true} \qquad \Gamma \vdash M \equiv N : \text{Prop}}{\Gamma \mid \Theta \vdash N\ \mathbf{true}} \qquad \frac{\Gamma \mid \Theta \vdash M \Rightarrow N\ \mathbf{true} \qquad \Gamma \mid \Theta \vdash M\ \mathbf{true}}{\Gamma \mid \Theta \vdash N\ \mathbf{true}} \qquad \frac{\Gamma \mid \Theta, M \vdash N\ \mathbf{true}}{\Gamma \mid \Theta \vdash M \Rightarrow N}$$

$$\frac{\Gamma \mid \Theta \vdash \forall_T\ M\ \mathbf{true} \qquad \Gamma \vdash N \equiv N : T}{\Gamma \mid \Theta \vdash M\ N\ \mathbf{true}} \qquad \frac{\Gamma \vdash M \equiv M : T \to \text{Prop} \qquad \Gamma \mid \Theta \vdash \mathbf{wf} \qquad \Gamma, x : T \mid \Theta \vdash M\ x\ \mathbf{true}}{\Gamma \mid \Theta \vdash \forall_T\ M\ \mathbf{true}}$$

Figure 4: Intuitionistic Higher-order Logic