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.

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 spec-
ifications.

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 \( \omega \)) [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.
Flexible variables may appear primed ($x'$) or unprimed ($x$) 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 $\square[A]_{\nu}$, where $A$ is an action and $\nu$ is a function on the system state. Intuitively, the formula $\square[A]_{\nu}$ means “it is always the case that either the action $A$ happens or $\nu$ 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 $\forall \forall x.P$.

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 ($\forall \forall$). Naïvely, flexible quantification would be

$$\theta, \rho \models \forall \forall x.P \iff \text{for every } d \in D^n, \quad \theta, \rho \cup (x \mapsto d) \models P$$

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

$$\rho'(m) = \rho(m) \quad \text{when } m \leq n$$

$$\rho'(m) = \rho(m - 1) \quad \text{when } m > n.$$
2. there is no bounded increasing sequence $t_0,t_1,t_2,...$ such that for all $i$, $f(t_i) \neq f(t_{i+1})$.

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 $\lambda$-calculus. It may be viewed as an extension to multi-sorted first-order logic that adds features for quantifying over function types and
\[ \text{next}(\tau, S) \triangleq 0 \quad \text{when } \forall t \in \mathbb{R}_{\geq 0}, \forall x \in S, \tau(0)(x) = \tau(t)(x) \]

\[ \text{next}(\tau, S) \triangleq \sup \{ r \mid \forall 0 \leq k \leq r, \forall x \in S, \tau(0)(x) = \tau(k)(x) \} \quad \text{otherwise} \]

\[ \theta, \tau \models_R \Box [A]_{x_1, \ldots, x_n} \iff r = 0 \text{ or } \theta, \tau(0), \tau(r) \models A \]

where \( r = \text{next}(\tau, \{ x_i | 0 \leq i \leq n \}) \)

\[ \theta, \tau \models_R T_1 \land T_2 \iff \theta, \tau \models_R T_1 \text{ and } \theta, \tau \models_R T_2 \]

\[ \theta, \tau \models_R \neg T \iff \theta, \tau \not\models_R T \]

\[ \theta, \tau \models_R \forall x. T \iff \text{iff for every } v \in D \text{ we have } (\theta, x \mapsto v), \tau \models_R T \]

\[ \theta, \tau \models_R \exists x. T \iff \text{iff for every } v \in D^{\mathbb{R}^+} \text{ we have } \theta, (x \mapsto (\tau(r), x \mapsto v(r))) \models_R T \]

\[ \theta, \tau \models_R \Box T \iff \text{iff for every } k \in \mathbb{R}_{\geq 0} \text{ such that } \theta, \tau[k..] \models_R T \]

Figure 3: Continuous-time Semantics of TLA

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 topos.

**Definition 3.** A topos is a cartesian closed category \( \mathcal{E} \) possessing all finite limits and a subobject classifier, i.e. an object \( \Omega \) and a monic arrow \( \text{true} : 1 \to \Omega \) such that \( \forall \) monic \( m : S \to B \)

\[
\exists ! \phi_m : B \to \Omega \text{ such that } \int_m \downarrow \text{true is a pullback.}
\]

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)

\[
\Gamma \vdash P, Q : \sigma \to \text{Prop} \quad \Gamma, x : \sigma \mid \Theta, Px \models Q_x \quad \Gamma, x : \sigma \mid \Theta, Qx \models Px \\
\Gamma \mid \Theta \vdash P =_{\sigma \to \text{Prop}} Q
\]

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 \( P : \mathcal{C}^{\text{op}} \to \text{HeyAlg} \) be a functor from a cartesian closed \( \mathcal{C} \)
into the category of Heyting algebras such that:

1. \( \forall X, Y : \text{Obj} \mathcal{C} \) there are monotone \( \exists^X_Y : \text{Hom}_{\mathcal{PreOrd}}(P(X \times Y), P(Y)) \) such that for \( \pi : X \times Y \to Y \) the projection \( \exists^X_Y \vdash P(\pi) \vdash \forall^Y_X \) and satisfying the Beck-Chevalley condition

\[
P(X \times Y') P(\text{id}_{X \times Y}) P(X \times Y)
\]

commutes as does the similar \( \exists^X_Y \) diagram;

2. \((\text{Forget} \cdot P) : \mathcal{C}^{op} \to \text{Set}\) 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 \((A, \Box) : \text{Obj}(\text{MAlg})\) 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 \(P : \mathcal{C}^{op} \to \text{MAlg}\) be a functor from a small cartesian closed category \(\mathcal{C}\) into the category of Modal algebras \(\text{MAlg}\) otherwise satisfying the axioms of a hyperdoctrine.

The hyperdoctrine semantics fully generalizes the topos semantics, as every topos \(T\) induces a (intuitionistic) hyperdoctrine

\[
(T, \text{Hom}_T(-, \Omega)).
\]

However, these are not the only hyperdoctrines of interest. Specifically, the only fact about \(\Omega\) 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 \(\text{Hom}_\mathcal{E}(-, H)\) with a Heyting algebra structure so that \((\mathcal{E}, \text{Hom}_\mathcal{E}(-, H))\) 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 topos. A geometric morphism \(f : \mathcal{E} \to \mathcal{F}\) is an adjunction \(\mathcal{E} \xleftarrow{f^*} \xrightarrow{f_*} \mathcal{F}\) such that the left adjoint \(f^*\), known as the inverse image, preserves finite limits. If every object \(X : \text{Obj}(\mathcal{E})\) is a subquotient of an object of the inverse image \(f^*\), so that there exists \(Y : \text{Obj}(\mathcal{F})\) and diagram \(f^*(Y) \leftrightarrow S \to X\), then \(f\) is localic.
Geometric morphisms are a source of internal complete Heyting algebras.

**Proposition 2.** Let \( f : E \to F \) a geometric morphism. Then \( f_*(\Omega_E) \) is a complete Heyting algebra internal to \( F \).

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

**Lemma 1** ([8] C1.3). In any topos \( E \), the subobject classifier \( \Omega_E \) is the initial complete Heyting algebra object. That is, for all complete Heyting algebras \( H \) internal to \( E \), there is a unique map of complete Heyting algebras \( i : \Omega_E \to H \). Moreover, the right adjoint of \( \tau \) 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 \( E \), let \( i \vdash \tau \) the canonical adjunction \( i : \Omega_E \vdash H : \tau \). The composite \( i \circ \tau \) is an S4 modality on \( H \).

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

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 \( f : D \to C \) be a functor of small categories. If \( f \) is faithful, then the induced geometric morphism \( Psh(D) \to Psh(C) \) is localic.

Thus we obtain a modal hyperdoctrine on \( (Psh(K), \text{Hom}_{Psh(K)}(\_, f_*(\Omega_{Psh(|K|)}))) \). In particular, as \( |K| \) is a groupoid, \( E = Psh(|K|) \) is a Boolean topos, so \( f_*(\Omega_E) \) is not only a complete Heyting algebra internal to \( F = Psh(K) \), 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. Pre-
orders are inadequate to this task. Luck-
ily, the geometric morphism construc-
tion described in Section 3 is not spe-
cific to Kripke’s inclusion of a discrete set
into a preorder. Any faithful functor be-
tween small categories whose domain is
a groupoid induces a model of classical
higher-order modal logic.

Our model is enabled by the following
elementry semantics: in the continuous
semantics, stuttering invariance is pre-
cisely closure under the action of stutters.

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

By \( S \) we denote the group of stutters

\[
S = \{ f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \mid f \text{ is a stutter} \}, \cdot, id_{\mathbb{R}_{\geq 0}}
\]

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

Non-Zeno functions on a set form a
\( S \)-set where the action of \( S \) is pre-
composition. Stuttering invariant subsets
of that set are then, exactly, sub \( S \)-sets.
As such, the category of \( S \)-sets \( (Psh(BS)) \)
seems to be closely connected to our prob-
lem. Since \( S \) is a group \( BS \) is a groupoid
and the presheaf topos \( Psh(BS) \) is boolean.
Therefore, it is a tempting target for the
semantics of a higher-order TLA. We al-
ready know this will not work on its own
though, as a topos is not enough to in-
terpret the modalities. The most important
modality for our purposes is \( \Box \). 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 bi-
continuous functions, \( \Box \) has to do with the
translation of time.

To that end, we introduce a generaliza-
tion of stutters, which we call “falters,”
which can include translation as well as di-
lation.

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

By \( F \) we denote the monoid of falters (under
function composition).

There is a natural morphism of monoids
\( \iota : S \rightarrow F \) given by inclusion, inducing
a faithful functor \( \iota : BS \rightarrow BF \). As men-
tioned in Section 3, such a faithful functor
induces a localic geometric morphism on
the associated presheaf categories \( \iota^* \dashv \iota_* : Psh(BS) \leftrightarrow Psh(BF) \). Our proposed model
for a higher-order TLA is the hyperdoc-
trine induced by this geometric morphism.

We will now elaborate some details of
this model. We consider \( F \)-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

\[ \text{Flex} : \text{Set} \to \mathcal{P}sh(\mathcal{B} \mathcal{F}) \]
\[ \text{Flex}(S) = \{ \{ f : \mathbb{R}_{\geq 0} \to S \mid f \text{ non zero} \} \} \]

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

\[ \text{Rigid} : \text{Set} \to \mathcal{P}sh(\mathcal{B} \mathcal{F}) \]
\[ \text{Rigid}(S) = (S, ((\_ , x) \mapsto x)) \]

There is a natural inclusion morphism from \( \text{Rigid} \to \text{Flex} \), which is (for every set) monic. However, \( \text{Rigid}(S) \) is not the only subobject of \( \text{Flex}(S) \). 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 \( \mathcal{P}sh(\mathcal{B} \mathcal{S}) \). As \( \mathcal{S} \) is a group, it has only two ideals, \( \emptyset \) and \( \mathcal{S} \). Thus, \( \Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})} \) 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 \( \mathcal{B} \mathcal{S} \) and \( \mathcal{B} \mathcal{F} \) have singleton objects, this can be computed pointwise. Given \( \mathcal{F} : \text{Set}^{\mathcal{B} \mathcal{S}} \), we compute

\[ \lim \left( \bullet_{\mathcal{S}} \downarrow \mathcal{F} \xrightarrow{\mathcal{F}_*} \mathcal{B} \mathcal{S} \xrightarrow{\mathcal{F}_*} \text{Set} \right) \]

which amounts to equalizing away the stutter action

\[ \Pi_{\mathcal{S} \times \mathcal{F}}(\bullet_{\mathcal{S}}) \twoheadrightarrow \Pi_{\mathcal{F}}(\bullet_{\mathcal{S}}) \twoheadrightarrow \Pi_{\mathcal{S} \times \mathcal{F}}(\bullet_{\mathcal{S}}) \]

On \( \mathcal{P}sh(\mathcal{B} \mathcal{S}) \)'s subobject classifier, this is

\[ \text{Prop} \triangleq \iota_*(\Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})}) \]
\[ = \{ \{ p : \mathcal{F} \to 2 \mid \forall m \in \mathcal{S}, n \in \mathcal{F}, p(n) = p(nm) \} \]
\[ , \quad (n, p) \mapsto ((r') \mapsto p(n \cdot r')) \]
\[ \cong (\mathcal{P}(\mathbb{R}_{\geq 0}), (n, O) \mapsto \text{im}^{-1}(n)(O)) \]

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 \( \mathcal{P}sh(\mathcal{B} \mathcal{F}) \) is the collection of falter ideals

\[ \Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{F})} = \{ I \subseteq \mathcal{F} \mid \forall i \in I \forall f \in \mathcal{F}, i \cdot f \in I \} \]

but these are just all upward-closed subsets of \( \mathbb{R}_{\geq 0} \), so \( \Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{F})} \cong (\mathcal{P}_1(\mathbb{R}_{\geq 0}), (n, O) \mapsto \text{im}^{-1}(n)(O)) \). As subobject classifier in \( \mathcal{P}sh(\mathcal{B} \mathcal{F}) \), \( \Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{F})} \) is initial in complete Heyting algebras internal to \( \mathcal{F} \), so the obvious equivariant inclusion \( i_{\Omega} : \Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{F})} \hookrightarrow \iota_*(\Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})}) \) is essentially unique. The right adjoint \( \tau_{\Omega} : \iota_*(\Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})}) \to \Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{F})} \), which classifies \( 1 \to \iota_*(\Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})}) \), is, then, the upward closure \( \uparrow (\_ : \mathcal{P}(\mathbb{R}) \to \mathcal{P}_1(\mathbb{R})) \). The adjunction \( \square := i_{\Omega} \circ \tau_{\Omega} : \text{End}(\iota_*(\Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})})) \) provides a left exact comonad on the complete internal Heyting algebra \( \iota_*(\Omega_{\mathcal{P}sh(\mathcal{B} \mathcal{S})}) \).

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

\[ \square(\_ : \text{Prop} \to \mathcal{P}sh(\mathcal{B} \mathcal{F}) \text{ Prop}) \]
\[ \square(S) = \{ r \in \mathbb{R}_{\geq 0} \mid \forall r' \geq r, r' \in S \} \]
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 \((Psh(BΣ), \text{Hom}(−, τ⋆(ΩBS)))\) 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 cartesian 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


A Rules of Higher-order logic

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

\[
\begin{align*}
M & \Rightarrow N \triangleq (\Rightarrow) M \ N \\
\neg M & \triangleq M \Rightarrow \bot \\
\forall(x : T).M & \triangleq \forall_T(\lambda(x : T).M) \\
M \land \ N & \triangleq \forall(p : \text{Prop}).(M \Rightarrow N \Rightarrow p) \Rightarrow p \\
\bot & \triangleq \forall(p : \text{Prop}), p \\
M \lor \ N & \triangleq \forall(p : \text{Prop}).(M \Rightarrow p) \Rightarrow (N \Rightarrow p) \Rightarrow p \\
\top & \triangleq \forall(p : \text{Prop}), p \Rightarrow p \\
\exists(x : T).M & \triangleq \forall(p : \text{Prop}).(\forall(x : T).M \Rightarrow p) \Rightarrow p \\
\end{align*}
\]

\[
\begin{array}{ccc}
\Gamma \vdash M \equiv N : T & \Gamma \vdash M \equiv N & \Gamma \vdash N \equiv O & \ (x : T) \in \Gamma \\
\hline
\Gamma \vdash N \equiv M : T & \Gamma \vdash M \equiv O & \Gamma \vdash x \equiv x : T \\
\end{array}
\]

\[
\begin{array}{ccc}
\Gamma \vdash M_1 \equiv M_2 : S \rightarrow T & \Gamma \vdash N_1 \equiv N_2 : S & \Gamma, x : T \vdash M \equiv N : S \\
\hline
\Gamma \vdash M_1 \ N_1 \equiv M_2 \ N_2 : T & \Gamma \vdash N_1 \equiv N_2 : T & \Gamma, x : T \vdash M \ x \equiv N \ x : S \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \vdash (\lambda(x : T).M) \equiv M[N/x] : S \\
\hline
\Gamma \vdash (\Rightarrow) \equiv (\Rightarrow) : \text{Prop} \rightarrow \text{Prop} \rightarrow \text{Prop} \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \mid \emptyset \vdash \text{wf} \\
\hline
\Gamma \mid \emptyset, M \vdash \text{wf} \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \mid \emptyset \vdash \text{true} \\
\hline
\Gamma \mid \Theta \vdash \text{true} \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \mid \emptyset \vdash M \equiv N : \text{Prop} \\
\hline
\Gamma \mid \emptyset \vdash M \Rightarrow N \ \text{true} \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \mid \emptyset \vdash M \equiv N : \text{Prop} \\
\hline
\Gamma \mid \emptyset \vdash M \Rightarrow N \ \text{true} \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \mid \emptyset \vdash \forall_T M \ \text{true} \\
\hline
\Gamma \mid \emptyset \vdash \forall_T M \ \text{true} \\
\end{array}
\]

\[
\begin{array}{c}
\Gamma \mid \emptyset \vdash M \ N \ \text{true} \\
\hline
\Gamma \mid \emptyset \vdash \forall_T M \ \text{true} \\
\end{array}
\]

Figure 4: Intuitionistic Higher-order Logic