A More Complete TLA

Stephan Merz

Institut für Informatik, Universität München
merz@informatik.uni-muenchen.de

Abstract. This paper defines a generalization of Lamport's Temporal Logic of Actions. We prove that our logic is stuttering-invariant and give an axiomatization of its propositional fragment. We also show that standard TLA is as expressive as our extension once quantification over flexible propositions is added.

1 Background

Temporal logics are routinely used for the specification and analysis of reactive systems. However, Lamport [10] has identified a shortcoming of standard linear-time temporal logic (LTL): because it is based on a global notion of "next state", it does not allow to relate specifications written at different levels of abstraction. He has therefore maintained that specifications should be invariant under "stuttering", that is, finite repetitions of identical states, and has proposed the Temporal Logic of Actions (TLA) [12, 13, 6]. Characteristically, TLA formulas contain the "next-time" operator only in a restricted form and can therefore not distinguish between stuttering-equivalent behaviors. Several case studies have established TLA as a useful formalism for describing systems; on the theoretical side, researchers have studied questions such as the description of real-time and hybrid systems [3, 11], the representation of assumption-commitment reasoning [4, 5], and the expressiveness of propositional TLA [18]. Moreover, Lamport has developed a formal specification language TLA+ based on TLA.

Although TLA has been found to be expressively complete for stuttering-invariant ω-regular languages [18], this does not necessarily imply that specifications can be expressed in a natural way. In fact, the syntactic restrictions imposed by Lamport that ensure invariance under stuttering occasionally make it hard to express seemingly simple properties. For example, whereas the requirement "eventually P will be true, and Q will hold at some later state" is expressed by the formula $\Diamond(P \wedge \Diamond Q)$, as in standard LTL, the analogous requirement "eventually action A will be performed, some time later followed by action B" is not expressed as easily. Eventual occurrence of action A is expressed by the formula $\Diamond\langle A\rangle_v$, where A describes the action as a relation on pairs of states, and v is (roughly speaking) the tuple of all state components of interest. One might therefore expect to express the informal requirement above by a formula such as $\Diamond\big\langle A \wedge \Diamond\langle B\rangle_v\big\rangle_v$, but TLA does not allow temporal formulas to occur inside an action formula (i.e., inside angle brackets). In some cases one can identify a state formula pA that is true iff action A has happened sometime in the past: for example, A might represent a request for a resource, and pA could be defined from the system's logfile. In those cases, we can express our requirement by the formula $\Diamond\langle pA \wedge B\rangle_v$. This

J. Wing, J. Woodcock, J. Davies (Eds.): FM'99, Vol. II, LNCS 1709, pp. 1226–1244, 1999.

formula requires that eventually action B occurs with pA being true—hence A must have occurred before. Observe, however, that the "point of reference" has changed with respect to the informal statement of the requirement, and that action A is no longer mentioned directly. If no suitable formula pA exists, we can "create" one using TLA's quantification over state variables, and write[1]

$$\exists\!\!\exists\, pA : \neg pA \wedge \Box[pA' \equiv (pA \vee A)]_v \wedge \Diamond\langle pA \wedge B\rangle_v$$

This formula defines pA to become true at the first occurrence of action A and then remain true forever; it is an example for a so-called *history variable* [2]. Although the formula can be shown to capture the informal requirement, it is certainly not natural.

Another concern that has not been resolved in a satisfactory way is the question of proof systems, even for propositional TLA. Lamport [12] states a relative completeness result for first-order TLA, subject to expressiveness assumptions similar to those for Hoare logics, for specifications in so-called "normal form". Formulas that deviate from "normal form" specifications arise naturally when specifications are composed [4]. Abadi [1] has proposed an axiomatization of an earlier version of TLA, but it is not clear whether his proof system can be adapted to the present-day TLA. This is in contrast to standard propositional temporal logic (PTL) whose axiomatization has been well understood since a landmark paper by Gabbay et al [8]. Complete axiomatizations are perhaps of rather academic interest; nevertheless they supply important information about the principles that underly a given logic, and they can form the basis of practical verification systems. For example, an accepted axiomatization would have helped us with the mechanization of TLA in the generic interactive theorem prover Isabelle [15].

In this paper we argue that the two shortcomings of TLA identified above are in fact related: we define the logic GTLA, which is a variant of TLA, but has a more liberal syntax. For example, $\Diamond\big\langle A \wedge \Diamond\langle B\rangle_v\big\rangle_v$ is a GTLA formula. We prove that GTLA, like TLA, is invariant under stuttering and provide a sound and complete axiomatization, via two different presentations. Finally, we show that TLA and GTLA are equally expressive once we add quantification over flexible propositions, preserving stuttering invariance. More precisely, while TLA is a sublogic of GTLA, every GTLA formula (possibly containing quantifiers) can be effectively translated to a quantified TLA formula. We argue that GTLA is better suited for verification than TLA. The added flexibility in expressiveness, which comes at no extra cost, may prove useful for writing specifications.

The plan of the paper is as follows: section 2 defines GTLA and contains the proof of stuttering invariance. Sections 3 and 4 introduce the first, heterogeneous version of an axiomatization for GTLA; an alternative, homogeneous presentation is derived in section 5. Section 6 compares the expressiveness of TLA and GTLA. Section 7 concludes the paper. Throughout, we restrict ourselves to propositional (or quantified propositional) logics, although the logic is easily extended to a first-order language.

[1] The formula becomes even more complex if A and B are allowed to occur simultaneously.

2 A Generalized TLA

We define the syntax and semantics of propositional GTLA and prove that all formulas are invariant under stuttering.

2.1 Syntax and Semantics

Assume given a denumerable set $\mathcal{V}$ of atomic propositions.

Definition 1. *Formulas and pre-formulas of GTLA are inductively defined as follows.*

1. *Every atomic proposition $v \in \mathcal{V}$ is a formula.*
2. *If F, G are formulas then $\neg F$, $F \Rightarrow G$, and $\Box F$ are formulas.*
3. *If P is a pre-formula and $v \in \mathcal{V}$ then $\Box[P]_v$ is a formula.*
4. *If F is a formula then F and $\circ F$ are pre-formulas.*
5. *If P, Q are pre-formulas then $\neg P$ and $P \Rightarrow Q$ are pre-formulas.*

The pre-formulas of GTLA generalize the transition formulas (actions) of TLA. In fact, propositional TLA can be defined similarly, except that clause (4) above should then be changed to

4'. If $v \in \mathcal{V}$ is an atomic proposition then v and $\circ v$ are pre-formulas.

We will use symbols such as F, G for formulas, P, Q for pre-formulas, and A, B for either formulas or pre-formulas. Note that, as in TLA, we consider $\Box$ and $\Box[\_]_v$ to be different operators, for each $v \in \mathcal{V}$.

In the following we assume standard abbreviations such as **true**, $\wedge$, $\vee$, $\equiv$, and $\not\equiv$ (equivalence, non-equivalence) for both formulas and pre-formulas. For compatibility with standard TLA syntax, we sometimes write v' instead of $\circ v$ when v is an atomic proposition. For a finite set $V = \{v_1, \ldots, v_n\} \subseteq \mathcal{V}$ of atomic propositions we let $\Box[P]_V$ denote the formula $\Box[P]_{v_1} \wedge \ldots \wedge \Box[P]_{v_n}$; in particular, $\Box[P]_\emptyset$ equals **true**. Stretching the notation even further, we write $\Box[P]_F$ (where F is any formula) for[2] $\Box[P \vee (\circ F \equiv F)]_{At(F)}$ where $At(F) \subseteq \mathcal{V}$ denotes the set of atomic propositions that occur in F. We write $\Diamond F$ for the formula $\neg\Box\neg F$ and $\Diamond\langle P\rangle_v$ for $\neg\Box[\neg P]_v$. Consequently, $\Diamond\langle P\rangle_{\{v_1,\ldots,v_n\}}$ denotes $\Diamond\langle P\rangle_{v_1} \vee \ldots \vee \Diamond\langle P\rangle_{v_n}$, and $\Diamond\langle P\rangle_F$ abbreviates $\Diamond\langle P \wedge (\circ F \not\equiv F)\rangle_{At(F)}$. Finally, we let $[P]_F$ and $\langle P\rangle_F$ abbreviate the pre-formulas $P \vee (\circ F \equiv F)$ and $P \wedge (\circ F \not\equiv F)$, respectively.

A *state* is a boolean valuation $s : \mathcal{V} \longrightarrow \{\text{tt}, \text{ff}\}$ of the atomic propositions. A *behavior* $\sigma = s_0 s_1 \ldots$ is an infinite sequence of states. For any $i \geq 0$, we denote by $\sigma|_i$ the suffix of σ starting at state s_i, that is, the sequence $s_i s_{i+1} \ldots$. We now define what it means for a (pre-)formula to hold of a behavior σ, written $\sigma \models F$ or $\sigma \approx\!\!\!\!| \; P$.

Definition 2. *The semantics of (pre-)formulas is given by the relation $|\!\approx$, which is inductively defined as follows:*

[2] This notation introduces an ambiguity when $F \equiv v$ is an atomic proposition. However, both possible interpretations are equivalent under the semantics of definition 2 below.

$\sigma \approx\!\!\models v$ *iff* $s_0(v) = \mathrm{tt}$ *(for $v \in \mathcal{V}$).*
$\sigma \approx\!\!\models \neg A$ *iff* *$\sigma \approx\!\!\models A$ does not hold.*
$\sigma \approx\!\!\models A \Rightarrow B$ *iff* *$\sigma \approx\!\!\models A$ implies $\sigma \approx\!\!\models B$.*
$\sigma \approx\!\!\models \Box F$ *iff* *$\sigma|_i \approx\!\!\models F$ holds for all $i \geq 0$.*
$\sigma \approx\!\!\models \Box[P]_v$ *iff* *for all $i \geq 0$, $s_i(v) = s_{i+1}(v)$ or $\sigma|_i \approx\!\!\models P$.*
$\sigma \approx\!\!\models \circ F$ *iff* *$\sigma|_1 \approx\!\!\models F$.*
For a formula F, we usually write $\sigma \models F$ instead of $\sigma \approx\!\!\models F$.

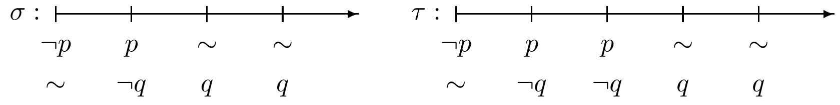
We say that a formula F is *valid over a behavior* σ iff $\sigma|_n \models F$ holds for all $n \geq 0$. Formula F *follows from* a set $\mathcal{F}$ of formulas (written $\mathcal{F} \models F$) iff F is valid over all behaviors over which all formulas $G \in \mathcal{F}$ are valid. Finally, F is *valid* (written $\models F$) iff it is valid over all behaviors, which is equivalent to saying that it follows from $\emptyset$.

Note that we have chosen the definition of *floating validity*, which is the traditional definition for modal logics, rather than the alternative *anchored validity*, which Lamport [12] uses. It is well known that either choice leads to the same set of valid formulas, although the consequence relation is different. We prefer floating validity because it is usually easier to axiomatize.

We say that a (pre-)formula is tautological if it results from a propositional tautology A of classical logic by consistently replacing atomic subformulas of A by formulas or pre-formulas. It is easy to see that every tautological formula is valid.

2.2 Stuttering Invariance

Definition 1 allows the $\Box$ operator to be applied only to formulas. For example, $\Box\circ v$ is not a pre-formula, although $\circ\Box v$ is. Had we allowed pre-formulas to freely contain outermost boxes, we would not obtain invariance under stuttering: consider, for example, $\Box[\Box(p \Rightarrow \circ q)]_p$, which is not a GTLA formula, and the behaviors σ and τ, where τ differs from σ only in the repetition of a single state, as illustrated by the following diagram (where $\sim$ means "don't care"):



Assuming the last state to repeat indefinitely, $\Box[\Box(p \Rightarrow \circ q)]_p$ clearly holds of σ, but not of τ.

We now formally define stuttering equivalence and prove that GTLA formulas do not distinguish between stuttering equivalent behaviors.

Definition 3 (stuttering equivalence). *Let $V \subseteq \mathcal{V}$ be a set of atomic propositions.*

1. *Two states s, t are called V-*similar*, written $s \simeq_V t$ iff $s(v) = t(v)$ for all $v \in V$.*
2. *V-*stuttering equivalence*, again written $\simeq_V$, is the smallest equivalence relation on behaviors that identifies $\rho \circ \langle s \rangle \circ \sigma$ and $\rho \circ \langle tu \rangle \circ \sigma$, for any finite sequence of states ρ, infinite sequence of states σ, and pairwise V-similar states s, t, u.*
3. *Stuttering equivalence (written $\simeq$) is $\mathcal{V}$-stuttering equivalence.*

It follows that $\sigma \simeq_V \tau$ implies $\sigma \simeq_W \tau$ whenever $W \subseteq V$ holds. In particular, stuttering equivalence is the finest relation among all $\simeq_V$. Let us list some elementary facts about stuttering equivalent behaviors.

Proposition 4. *Assume that $\sigma \simeq_V \tau$ holds for behaviors $\sigma = s_0 s_1 \ldots$ and $\tau = t_0 t_1 \ldots$.*

1. *$t_0 \simeq_V s_0$.*
2. *For every $n \geq 0$ there is some $m \geq 0$ such that $\sigma|_n \simeq_V \tau|_m$ and $\sigma|_{n+1} \simeq_V \tau|_{m+1}$.*

Theorem 5 (stuttering invariance). *For any GTLA formula F and any behaviors σ, τ such that $\sigma \simeq_{At(F)} \tau$, we have $\sigma \models F$ iff $\tau \models F$.*

Proof. We simultaneously prove the following assertions by induction on the structure of (pre-)formulas, for all behaviors $\sigma = s_0 s_1 \ldots$ and $\tau = t_0 t_1 \ldots$.

1. If $\sigma \simeq_{At(F)} \tau$ then $\sigma \models F$ iff $\tau \models F$.
2. If $\sigma \simeq_{At(P)} \tau$ and $\sigma|_1 \simeq_{At(P)} \tau|_1$ then $\sigma \approx\!\!\!\models P$ iff $\tau \approx\!\!\!\models P$.

We first consider the different cases in the definition of formulas F.

$F \in \mathcal{V}$: The assertion follows from proposition 4.1, since $s_0(F) = t_0(F)$.

$\neg F$: immediate from the induction hypothesis.

$F \Rightarrow G$: Since $At(F) \subseteq At(F \Rightarrow G)$ and $At(G) \subseteq At(F \Rightarrow G)$, the assumption $\sigma \simeq_{At(F \Rightarrow G)} \tau$ implies both $\sigma \simeq_{At(F)} \tau$ and $\sigma \simeq_{At(G)} \tau$. This observation, together with the induction hypothesis, implies the assertion.

$\Box F$: By symmetry of $\simeq_{At(\Box F)}$, it is enough to prove "if". So assume that $\tau \models \Box F$, and let $n \geq 0$ be arbitrary. Proposition 4.2 implies that there exists some $m \geq 0$ such that $\sigma|_n \simeq_{At(\Box F)} \tau|_m$. From $\tau \models \Box F$ we conclude $\tau|_m \models F$, and therefore $\sigma|_n \models F$ by induction hypothesis, since $At(\Box F) = At(F)$.

$\Box[P]_v$: Again, we need only prove the "if" part. Assume that $\tau \models \Box[P]_v$, and let $n \geq 0$ be arbitrary. Choose $m \geq 0$ such that $\sigma|_n \simeq_{At(\Box[P]_v)} \tau|_m$ and also $\sigma|_{n+1} \simeq_{At(\Box[P]_v)} \tau|_{m+1}$; proposition 4.2 ensures that m exists. Proposition 4.1 implies that $s_n(v) = t_m(v)$ and $s_{n+1}(v) = t_{m+1}(v)$. If $t_m(v) = t_{m+1}(v)$, it follows that $s_n(v) = s_{n+1}(v)$, and we are done. Otherwise, by the assumption $\tau \models \Box[P]_v$ it follows that $\tau|_m \approx\!\!\!\models P$, and the induction hypothesis (for assertion 2) gives $\sigma|_n \approx\!\!\!\models P$ because $\simeq_{At(\Box[P]_v)} \subseteq \simeq_{At(P)}$.

Turning to assertion 2, we consider the cases in the definition of pre-formulas:

P a formula : immediate from the induction hypothesis for assertion 1.

$\circ F$: The assumption that $\sigma|_1 \simeq_{At(\circ F)} \tau|_1$ and the induction hypothesis for assertion 1 imply $\sigma|_1 \models F$ iff $\tau|_1 \models F$, and therefore $\sigma \approx\!\!\!\models \circ F$ iff $\tau \approx\!\!\!\models \circ F$.

$\neg P, P \Rightarrow Q$: analogous to the corresponding cases for formulas. $\because$

$$
\begin{array}{ll@{\qquad}ll}
\text{(ax0)} & \vdash F \text{ whenever } F \text{ is tautological} & \text{(pax0)} & \mathrel{|\!\sim} P \text{ whenever } P \text{ is tautological} \\
\text{(ax1)} & \vdash \Box F \Rightarrow F & \text{(pax1)} & \mathrel{|\!\sim} \circ\neg F \equiv \neg\circ F \\
\text{(ax2)} & \vdash \Box F \Rightarrow \Box[\Box F]_v & \text{(pax2)} & \mathrel{|\!\sim} \circ(F \Rightarrow G) \Rightarrow (\circ F \Rightarrow \circ G) \\
\text{(ax3)} & \vdash \Box[F \Rightarrow \circ F]_F \Rightarrow (F \Rightarrow \Box F) & \text{(pax3)} & \mathrel{|\!\sim} \Box F \Rightarrow \circ\Box F \\
\text{(ax4)} & \vdash \Box[P \Rightarrow Q]_v \Rightarrow (\Box[P]_v \Rightarrow \Box[Q]_v) & \text{(pax4)} & \mathrel{|\!\sim} \Box[P]_v \equiv [P]_v \wedge \circ\Box[P]_v \\
\text{(ax5)} & \vdash \Box[v' \not\equiv v]_v & \text{(pax5)} & \mathrel{|\!\sim} \circ\Box F \Rightarrow \Box[\circ F]_v
\end{array}
$$

$$
\text{(mp)}\ \frac{\vdash F \qquad \vdash F \Rightarrow G}{\vdash G} \qquad\qquad \text{(pmp)}\ \frac{\mathrel{|\!\sim} P \qquad \mathrel{|\!\sim} P \Rightarrow Q}{\mathrel{|\!\sim} Q}
$$

$$
\text{(sq)}\ \frac{\mathrel{|\!\sim} P}{\vdash \Box[P]_v} \qquad\qquad \text{(pre)}\ \frac{\vdash F}{\mathrel{|\!\sim} F} \qquad \text{(nex)}\ \frac{\vdash F}{\mathrel{|\!\sim} \circ F}
$$

Fig. 1. The proof system Σ_{GTLA}.

3 An Axiomatization of GTLA

We now present a proof system Σ_{GTLA} for GTLA and prove its adequacy. Σ_{GTLA} is based on two provability relations $\vdash$ and $\mathrel{|\!\sim}$ for formulas and pre-formulas; we therefore call Σ_{GTLA} a heterogeneous proof system. An alternative "homogeneous" proof system will be given in section 5. Figure 1 contains the axioms and rules that define $\vdash$ and $\mathrel{|\!\sim}$. We extend $\vdash$ to a relation between sets of formulas and formulas by defining $\mathcal{F} \vdash F$ iff $\vdash F$ can be established from the axioms and rules of Σ_{GTLA} if additionally $\vdash G$ is assumed for all formulas $G \in \mathcal{F}$, and similarly define $\mathcal{F} \mathrel{|\!\sim} P$. Because we are ultimately only interested in the relation $\vdash$ for formulas, we do not allow pre-formulas to occur in the set $\mathcal{F}$ of hypotheses.

Many of the axioms and rules of Σ_{GTLA} are familiar from propositional linear-time temporal logic [8, 9]. First observe that both $\vdash$ and $\mathrel{|\!\sim}$ contain full propositional calculus. Axiom (ax3) is a "stuttering-invariant" version of the induction axiom. Its formulation relies essentially on the GTLA syntax that allows temporal formulas in the scope of the $\Box[\_]_v$ operator. Axiom (ax5) effectively asserts that the pre-formula P in $\Box[P]_v$ is evaluated only when v changes value. Axiom (pax1) expresses that time is linear. We cannot state an induction principle for formulas of the form $\Box[P]_v$ because $\circ P$ or $\circ[P]_v$ are not even pre-formulas. For this reason, (pax4) is stronger than its counterparts (ax1) and (pax3). Axiom (pax5) asserts a form of commutativity for the $\circ$ and $\Box$ operators. The rules (sq) and (nex) reflect the floating definition of validity. The necessitation rule

$$
\text{(alw)}\ \frac{\vdash F}{\vdash \Box F}
$$

is easily derived in Σ_{GTLA}. Note also that the axioms (ax2), (ax4), (pax4), (pax5) and the rule (sq) are easily generalized to versions where the "index" v is replaced by a finite set V of atomic propositions, or by a GTLA formula.

Theorem 6 (Soundness). *For any set $\mathcal{F}$ of formulas, $\mathcal{F} \vdash F$ implies $\mathcal{F} \models F$.*

Proof. The proof is by induction on the assumed derivation of F from $\mathcal{F}$, also proving that $\mathcal{F} \mathrel{|\!\!\sim} P$ implies that $\sigma|_n \mathrel{|\!\!\approx} P$ holds for every $n \geq 0$ and every behavior σ such that all formulas in $\mathcal{F}$ are valid over σ. We only consider a few cases.

(ax3) It suffices to prove $\sigma \models \Box[F \Rightarrow \circ F]_F \Rightarrow (F \Rightarrow \Box F)$, for any formula F and any behavior $\sigma = s_0 s_1 \ldots$. So suppose $\sigma \models \Box[F \Rightarrow \circ F]_F$ and $\sigma \models F$. We prove $\sigma|_n \models F$ for every $n \geq 0$, by induction on n. The base case being trivial, assume that $\sigma|_n \models F$. If $s_n \simeq_{At(F)} s_{n+1}$, we have $\sigma|_n \simeq_{At(F)} \sigma|_{n+1}$, and theorem 5 ensures that $\sigma|_{n+1} \models F$. Otherwise, there is some $v \in At(F)$ such that $s_n(v) \neq s_{n+1}(v)$, and the assumption $\sigma \models \Box[F \Rightarrow \circ F]_F$ implies that $\sigma|_n \mathrel{|\!\!\approx} (F \Rightarrow \circ F) \vee (\circ F \equiv F)$, hence again $\sigma|_{n+1} \models F$.

(pax5) Suppose $\sigma \mathrel{|\!\!\approx} \circ\Box F$, that is, $\sigma|_{n+1} \models F$, for every $n \geq 0$. We prove that $\sigma \models \Box[\circ F]_v$. Let $m \geq 0$ be arbitrary. The assumption ensures that $\sigma|_{m+1} \models F$, and therefore $\sigma|_m \mathrel{|\!\!\approx} \circ F$. This suffices.

(sq) Assume that $\mathcal{F} \mathrel{|\!\!\sim} P$, that σ is some behavior such that all formulas in $\mathcal{F}$ are valid over σ, and that $n \geq 0$. We need to prove that $\sigma|_n \models \Box[P]_v$. So let $m \geq 0$ be arbitrary. By induction hypothesis, we know that $\sigma|_{n+m} \mathrel{|\!\!\approx} P$, and therefore $(\sigma|_n)|_m \mathrel{|\!\!\approx} P$. This suffices. ∵

We also have a version of the deduction theorem for Σ_{GTLA}, as stated in the following theorem.

Theorem 7. *For any set $\mathcal{F}$ of formulas, any formulas F, G, and any pre-formula P we have $\mathcal{F} \cup \{F\} \vdash G$ iff $\mathcal{F} \vdash \Box F \Rightarrow G$ and $\mathcal{F} \cup \{F\} \mathrel{|\!\!\sim} P$ iff $\mathcal{F} \mathrel{|\!\!\sim} \Box F \Rightarrow P$.*

Proof. "if": Assume $\mathcal{F} \vdash \Box F \Rightarrow G$. A fortiori, we have $\mathcal{F} \cup \{F\} \vdash \Box F \Rightarrow G$. The derived rule (alw) implies that $\mathcal{F} \cup \{F\} \vdash \Box F$, and therefore we have $\mathcal{F} \cup \{F\} \vdash G$ by (mp). The second assertion is proven similarly.

"only if": The proof is by induction on the assumed derivations of $\mathcal{F} \cup \{F\} \vdash G$ and $\mathcal{F} \cup \{F\} \mathrel{|\!\!\sim} P$ (simultaneously for all F and P).

- If G is an axiom or $G \in \mathcal{F}$, we have $\mathcal{F} \vdash G$, and $\mathcal{F} \vdash \Box F \Rightarrow G$ follows by propositional reasoning. The same argument applies for the second assertion when P is an axiom.
- If G is F, then $\mathcal{F} \vdash \Box F \Rightarrow F$ is an instance of (ax1).
- If G results from an application of (mp) to previously derived formulas $H \Rightarrow G$ and H, then the induction hypothesis implies $\mathcal{F} \vdash \Box F \Rightarrow (H \Rightarrow G)$ as well as $\mathcal{F} \vdash \Box F \Rightarrow H$, from which we conclude $\mathcal{F} \vdash \Box F \Rightarrow G$ by propositional reasoning. The same argument holds for (pmp).
- Assume that G results from an application of (sq), say, $G \equiv \Box[Q]_v$. By induction hypothesis, we have $\mathcal{F} \mathrel{|\!\!\sim} \Box F \Rightarrow Q$, and we continue as follows:

(1)	$\mathrel{|\!\!\sim} \Box F \Rightarrow Q$	(ind.hyp.)
(2)	$\vdash \Box[\Box F \Rightarrow Q]_v$	(sq)(1)
(3)	$\vdash \Box[\Box F \Rightarrow Q]_v \Rightarrow (\Box[\Box F]_v \Rightarrow \Box[Q]_v)$	(ax4)
(4)	$\vdash \Box F \Rightarrow \Box[\Box F]_v$	(ax2)
(5)	$\vdash \Box F \Rightarrow \Box[Q]_v$	(prop)(2)(3)(4)

- If G results from an application of (pre), then by induction hypothesis we have $\mathcal{F} \vdash \Box F \Rightarrow G$, and therefore also $\mathcal{F} \mathrel{|\!\!\sim} \Box F \Rightarrow G$, by (pre).
- If $G \equiv \circ H$ results from an application of (nex), then the induction hypothesis yields $\mathcal{F} \vdash \Box F \Rightarrow H$. Rule (nex) shows $\mathcal{F} \mathrel{|\!\!\sim} \circ(\Box F \Rightarrow H)$, and we obtain $\mathcal{F} \mathrel{|\!\!\sim} \circ\Box F \Rightarrow \circ H$ by (pax2) and (pmp). The conclusion $\mathcal{F} \mathrel{|\!\!\sim} \Box F \Rightarrow \circ H$ follows with the help of (pax3). $\because$

The following are some derived theorems of Σ_{GTLA}, which will be used later. Derivations of these theorems can be found in the full version of this paper [14].

$$\begin{array}{llll}
\text{(T1)} & \vdash \Box F \equiv \Box\Box F & \text{(T2)} & \vdash \Box[P]_v \equiv \Box\Box[P]_v \\
\text{(T3)} & \vdash \Box[[P]_v]_v \equiv \Box[P]_v & \text{(T4)} & \vdash \Box[P]_v \Rightarrow \Box[[P]_v]_w \\
\text{(T5)} & \vdash \Box[[P]_w]_v \Rightarrow \Box[[P]_v]_w & \text{(T6)} & \vdash \Box F \Rightarrow \Box[\circ F]_v \\
\text{(T7)} & \mathrel{|\!\!\sim} \Box F \equiv F \wedge \circ\Box F & \text{(T8)} & \mathrel{|\!\!\sim} \circ(F \wedge G) \equiv \circ F \wedge \circ G
\end{array}$$

By rule (pre), every provable formula is also provable as a pre-formula. An important result for Σ_{GTLA} shows that the converse is also true. This can be shown by a careful analysis of the derivations in Σ_{GTLA}; the full proof is given in [14].

Theorem 8. *For any set $\mathcal{F}$ of formulas and any formula F:*

$$\mathcal{F} \vdash F \quad \textit{iff} \quad \mathcal{F} \mathrel{|\!\!\sim} F \quad \textit{iff} \quad \mathcal{F} \mathrel{|\!\!\sim} \circ F$$

4 Completeness of Σ_{GTLA}

We will now prove the completeness of Σ_{GTLA}. Let us first note that GTLA, just as PTL, is not compact:

Example 9. Let $\mathcal{F} = \{\Box[v_i \Rightarrow v'_{i+1}]_{v_i}, \Box(v_i \Rightarrow w) : i \geq 0\}$. It is easy to see that $\mathcal{F} \models v_0 \Rightarrow \Box w$, but we can clearly not derive $\mathcal{F} \vdash v_0 \Rightarrow \Box w$, because this would require the infinitary invariant $\Box \bigvee_{i \geq 0} v_i$.

We can therefore only hope for completeness when $\mathcal{F}$ is a finite set, and by theorem 7 it is enough to show that $\models F$ implies $\vdash F$.

Our completeness proof follows the standard approach [9] of constructing a model for a finite and consistent set of formulas. To do so, we have to assemble information about pre-formulas as well as formulas. Nevertheless, the critical step in the proof is to show that all the essential information is contained in the formulas used for the construction; this is due to the fact that the assumptions in a derivation $\mathcal{F} \vdash F$ do not contain pre-formulas. For a set $\mathcal{G}$ of formulas and pre-formulas, we denote by $\mathcal{G}^F$ the set of all formulas contained in $\mathcal{G}$. We also use $\mathcal{G}$ to denote the conjunction of all (pre-) formulas in $\mathcal{G}$; it will always be clear from the context whether we refer to the set or the (pre-)formula.

A set $\mathcal{G}$ is called *inconsistent* if $\mathrel{|\!\!\sim} \neg\mathcal{G}$, otherwise it is called *consistent*. Note that if $\mathcal{G}$ is consistent and A is any formula or pre-formula, one of the sets $\mathcal{G} \cup \{A\}$ or $\mathcal{G} \cup \{\neg A\}$ is again consistent.

We inductively define a set $\tau(A)$ for any formula or pre-formula A, as follows:

$$\begin{array}{rlrl} \tau(v) &= \{v\} & \tau(\neg A) &= \{\neg A\} \cup \tau(A) \\ \tau(A \Rightarrow B) &= \{A \Rightarrow B\} \cup \tau(A) \cup \tau(B) & \tau(\Box F) &= \{\Box F\} \cup \tau(F) \\ \tau(\Box[P]_v) &= \{\Box[P]_v, v, \circ v\} \cup \tau(P) & \tau(\circ F) &= \{\circ F\} \end{array}$$

For a set $\mathcal{G}$, we define $\tau(\mathcal{G})$ as the union of all $\tau(A)$, for all (pre-)formulas A contained in $\mathcal{G}$. Note that our definitions ensure that $\tau(\mathcal{G})$ is finite whenever $\mathcal{G}$ is finite.

We say that $\mathcal{G}$ is *complete* if it contains either A or $\neg A$, for every (pre-)formula A from $\tau(\mathcal{G})$. Observe that for every finite and consistent $\mathcal{G}$ there exist only finitely many finite, consistent, and complete $\mathcal{G}^* \supseteq \mathcal{G}$, since $\tau(\mathcal{G})$ is itself finite; we call any such $\mathcal{G}^*$ a *completion* of $\mathcal{G}$. We note the following elementary facts about complete sets. The proofs of assertions 1 and 3 are standard, whereas the second assertion follows from the first and theorem 8 by propositional reasoning, since $\mathcal{G} \Rightarrow \mathcal{G}^F$ holds for any set $\mathcal{G}$ by (ax0).

Proposition 10.

1. *Assume that $\mathcal{G}$ is finite and consistent, and that $\mathcal{G}_1^*, \ldots, \mathcal{G}_n^*$ are all the different completions of $\mathcal{G}$. Then $\mathrel{|\!\sim} \mathcal{G} \Rightarrow \mathcal{G}_1^* \vee \ldots \vee \mathcal{G}_n^*$.*
2. *Assume that $\mathcal{F}$ is a finite and consistent set of formulas, and that $\mathcal{G}_1, \ldots, \mathcal{G}_n$ are all the different completions of $\mathcal{F}$. Then $\vdash \mathcal{F} \Rightarrow \mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F$.*
3. *Assume that $\mathcal{G}$ is consistent and complete and that A, B are (pre-)formulas.*
   1. *(a) If $A \in \mathcal{G}$, $B \in \tau(\mathcal{G})$ and $\vdash A \Rightarrow B$ or $\mathrel{|\!\sim} A \Rightarrow B$ then $B \in \mathcal{G}$.*
   2. *(b) If $A \Rightarrow B \in \tau(\mathcal{G})$ then $A \Rightarrow B \in \mathcal{G}$ iff $A \notin \mathcal{G}$ or $B \in \mathcal{G}$.*

We now define a set $\sigma(\mathcal{G})$ of formulas that, intuitively, transfer information from one state of the model under construction to the next one.

$$\begin{array}{ll} \sigma_1(\mathcal{G}) = \{F : \circ F \in \mathcal{G}\} & \sigma_2(\mathcal{G}) = \{\neg F : \neg \circ F \in \mathcal{G}\} \\ \sigma_3(\mathcal{G}) = \{\Box F : \Box F \in \mathcal{G}\} & \sigma_4(\mathcal{G}) = \{\neg \Box F : \neg \Box F \in \mathcal{G}, F \in \mathcal{G}\} \\ \sigma_5(\mathcal{G}) = \{\Box[P]_v : \Box[P]_v \in \mathcal{G}\} & \\ \sigma_6(\mathcal{G}) = \{\neg\Box[P]_v : \neg\Box[P]_v \in \mathcal{G} \text{ and} & \\ \qquad\qquad P \in \mathcal{G} \text{ or } \{v, \circ v\} \subseteq \mathcal{G} \text{ or } \{\neg v, \neg \circ v\} \subseteq \mathcal{G}\} & \\ \sigma(\mathcal{G}) \;= \sigma_1(\mathcal{G}) \cup \sigma_2(\mathcal{G}) \cup \sigma_3(\mathcal{G}) \cup \sigma_4(\mathcal{G}) \cup \sigma_5(\mathcal{G}) \cup \sigma_6(\mathcal{G}) & \end{array}$$

Lemma 11. *Assume that $\mathcal{G}$ is finite.*

1. *$\mathrel{|\!\sim} \mathcal{G} \Rightarrow \circ\sigma(\mathcal{G})$.*
2. *If $\mathcal{G}$ is consistent, then so is $\sigma(\mathcal{G})$.*

Proof. 1. By (T8), it is enough to show $\mathrel{|\!\sim} \mathcal{G} \Rightarrow \circ F$, for every formula $F \in \sigma(G)$. We distinguish the different cases in the definition of $\sigma(\mathcal{G})$.

- For $F \in \sigma_1(\mathcal{G})$, we have $\circ F \in \mathcal{G}$, so the assertion follows by (pax0).
- If $F \equiv \neg G \in \sigma_2(\mathcal{G})$, then $\neg \circ F \in \mathcal{G}$, and the assertion follows using (pax1).
- If $F \equiv \Box G \in \sigma_3(\mathcal{G})$, we have $\Box G \in \mathcal{G}$; use (pax3) to prove the assertion.
- If $F \equiv \neg \Box G \in \sigma_4(\mathcal{G})$, the definition ensures $\mathrel{|\!\sim} \mathcal{G} \Rightarrow G \wedge \neg \Box G$, and the assertion follows by (T7), (pax1), and propositional logic.

  - For $F \equiv \Box[P]_v \in \sigma_5(\mathcal{G})$, use (pax4) to prove the assertion.
  - If $F \equiv \neg\Box[P]_v \in \sigma_6(\mathcal{G})$, the definition and (pax0) yield $\mathcal{G} \Rightarrow [P]_v \wedge \neg\Box[P]_v$, and the assertion follows by (pax4) and (pax1).

2. If $\sigma(\mathcal{G})$ is inconsistent, we have $\mid\!\sim \neg\sigma(\mathcal{G})$. By rule (nex), we obtain $\mid\!\sim \circ\neg\sigma(\mathcal{G})$. Using axiom (pax1) and propositional logic, assertion (1) implies $\mid\!\sim \neg\mathcal{G}$, that is, $\mathcal{G}$ is inconsistent. ∵

Given a finite and consistent set $\mathcal{F}$ of formulas, we inductively define a graph $\mathcal{T}(\mathcal{F})$ of sets of pre-formulas as follows:

- All different completions of $\mathcal{F}$ are nodes of $\mathcal{T}(\mathcal{F})$, called the *roots* of $\mathcal{T}(\mathcal{F})$.
- If $\mathcal{G}$ is a node in $\mathcal{T}(\mathcal{F})$ then its successors are all different completions of $\sigma(\mathcal{G})$.

It follows that every node $\mathcal{G}$ is finite, consistent, and complete. Also, the sub-graph of $\mathcal{T}(\mathcal{F})$ that consists of all nodes reachable from the successors of $\mathcal{G}$ is just $\mathcal{T}(\sigma(\mathcal{G}))$.

Lemma 12. *Assume that $\mathcal{F}$ is a finite and consistent set of formulas.*

1. *$\mathcal{T}(\mathcal{F})$ contains only finitely many different nodes $\mathcal{G}_1, \ldots, \mathcal{G}_n$.*
2. *Assume that $\mathcal{G}_1, \ldots, \mathcal{G}_n$ are all the different nodes in $\mathcal{T}(\mathcal{F})$.*
   (i) $\mid\!\sim \mathcal{G}_i^F \Rightarrow \mathcal{G}_1 \vee \ldots \vee \mathcal{G}_n$ *(for $i = 1, \ldots, n$).*
   (ii) $\mid\!\sim \mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F \Rightarrow \circ(\mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F)$.
   (iii) $\vdash \mathcal{F} \Rightarrow \Box(\mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F)$.

Proof. 1. The completions of a finite set $\mathcal{G}$ only contain – possibly negated – pre-formulas from the set $\tau(\mathcal{G})$, which is also finite. On the other hand, the only pre-formulas in $\sigma(\mathcal{G})$ that are possibly not in $\tau(\mathcal{G})$ are of the form F or $\neg F$ such that $\mathcal{G}$ contains $\circ F$ or $\neg\circ F$, hence the number of $\circ$ operators decreases, which is possible only finitely often. Therefore, only finitely many different (pre-)formulas occur in $\mathcal{T}(\mathcal{F})$, hence $\mathcal{T}(\mathcal{F})$ can contain only finitely many different nodes.

2. (i) Let $i \in \{1, \ldots, n\}$ be arbitrary, and consider the set $\mathcal{F}'$ of formulas from which the node $\mathcal{G}_i$ was constructed—either the initial set $\mathcal{F}$ or the set $\sigma(\mathcal{G}')$ where $\mathcal{G}'$ is a predecessor of $\mathcal{G}$ in $\mathcal{T}(\mathcal{F})$. Proposition 10.1 implies $\mid\!\sim \mathcal{F}' \Rightarrow \mathcal{G}_1 \vee \ldots \vee \mathcal{G}_n$ because all consistent completions of $\mathcal{F}'$ are contained in $\mathcal{T}(\mathcal{F})$. Since $\mathcal{G}_i$ is a completion of $\mathcal{F}'$, it follows that $\mathcal{F}' \subseteq \mathcal{G}_i^F$, hence we have $\vdash \mathcal{G}_i^F \Rightarrow \mathcal{F}'$ by (ax0), and therefore the assertion.
   (ii) We first note $\mid\!\sim \mathcal{G}_j \Rightarrow \circ\sigma(\mathcal{G}_j)$, for every node $\mathcal{G}_j$ of $\mathcal{T}(\mathcal{F})$, by lemma 11. Proposition 10.2 ensures $\vdash \sigma(\mathcal{G}_j) \Rightarrow \mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F$. Applying rule (nex) and (pax2), we obtain $\mid\!\sim \mathcal{G}_j \Rightarrow \circ(\mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F)$, for every j, hence also $\mid\!\sim \mathcal{G}_1 \vee \ldots \vee \mathcal{G}_n \Rightarrow \circ(\mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F)$. The assertion follows with the help of (i) and propositional logic.
   (iii) Let $\mathcal{I}$ denote the formula $\mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F$. Assertion (ii) and rule (sq) imply $\vdash \Box[\mathcal{I} \Rightarrow \circ\mathcal{I}]_{\mathcal{I}}$, hence $\vdash \mathcal{I} \Rightarrow \Box\mathcal{I}$ by axiom (ax3). On the other hand, proposition 10.2 implies $\vdash \mathcal{F} \Rightarrow \mathcal{I}$, and the assertion follows. ∵

We will construct a model for $\mathcal{F}$ from the paths in $\mathcal{T}(\mathcal{F})$. Let us call a path $\mathcal{G}_0, \mathcal{G}_1, \ldots$ *complete* iff it satisfies the two following conditions, for every $i \geq 0$:

- If $\neg\Box F \in \mathcal{G}_i$ then $\neg F \in \mathcal{G}_j$ for some $j \geq i$.
- If $\neg\Box[P]_v \in \mathcal{G}_i$ then for some $j \geq i$, $\neg P \in \mathcal{G}_j$ and either $\{v, \neg\circ v\} \subseteq \mathcal{G}_j$ or $\{\neg v, \circ v\} \subseteq \mathcal{G}_j$.

Lemma 13. *Assume that $\mathcal{F}$ is a finite and consistent set of formulas. Then $\mathcal{T}(\mathcal{F})$ contains a complete path starting at some root.*

Proof. We first prove that for every node $\mathcal{G}$ of $\mathcal{T}(\mathcal{F})$ and any formula F such that $\{\neg\Box F, F\} \subseteq \mathcal{G}$ there is some node $\mathcal{H}$ in $\mathcal{T}(\sigma(\mathcal{G}))$ that contains $\neg F$. Suppose not. Then, in particular, every root $\mathcal{W}$ of $\mathcal{T}(\sigma(\mathcal{G}))$ contains $\neg\Box F$ and F (because $F \in \tau(\sigma(\mathcal{G}))$ and $\mathcal{W}$ is a completion of $\sigma(\mathcal{G})$), hence $\neg\Box F \in \sigma(\mathcal{W})$. Inductively, it follows that $\{\neg\Box F, F\} \subseteq \mathcal{H}$ holds for every node $\mathcal{H}$ of $\mathcal{T}(\sigma(\mathcal{G}))$. Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be all nodes of $\mathcal{T}(\sigma(\mathcal{G}))$, and let $\mathcal{I}$ denote the formula $\mathcal{G}_1^F \vee \ldots \vee \mathcal{G}_n^F$. Then (ax0) gives $\mathcal{I} \Rightarrow F$, which proves $\mathcal{I} \vdash \Box F$, using rule (alw). By theorem 7, we conclude $\vdash \Box\mathcal{I} \Rightarrow \Box F$. Lemma 12.2(iii) yields $\vdash \sigma(\mathcal{G}) \Rightarrow \Box F$, but on the other hand we have $\vdash \sigma(\mathcal{G}) \Rightarrow \neg\Box F$ because $\neg\Box F \in \sigma(\mathcal{G})$. Therefore, $\sigma(\mathcal{G})$ and (by lemma 11.2) also $\mathcal{G}$ is inconsistent, and a contradiction is reached.

Similarly, we show that there is some node $\mathcal{H}$ in $\mathcal{T}(\sigma(\mathcal{G}))$ that contains $\neg P$ and either $\{v, \neg\circ v\}$ or $\{\neg v, \circ v\}$ whenever $\neg\Box[P]_v \in \mathcal{G}$ and either $P \in \mathcal{G}$ or $\{v, \circ v\} \subseteq \mathcal{G}$ or $\{\neg v, \neg\circ v\} \subseteq \mathcal{G}$. Suppose not. Then an argument analogous to the one above establishes that every node $\mathcal{H}$ contains P or $\{v, \circ v\}$ or $\{\neg v, \neg\circ v\}$. By axiom (pax0), this shows $\mid\!\sim \mathcal{H} \Rightarrow [P]_v$. Lemma 12.2(i) implies $\mid\!\sim \mathcal{I} \Rightarrow [P]_v$, and by (ax1) and (pre), a fortiori $\mid\!\sim \Box\mathcal{I} \Rightarrow [P]_v$. Using rule (sq) and (ax4), this shows $\vdash \Box[\Box\mathcal{I}]_v \Rightarrow \Box[[P]_v]_v$, and (T3) implies that $\vdash \Box[\Box\mathcal{I}]_v \Rightarrow \Box[P]_v$. But as above we have $\vdash \sigma(\mathcal{G}) \Rightarrow \Box\mathcal{I}$, and thus also $\vdash \sigma(\mathcal{G}) \Rightarrow \Box[\Box\mathcal{I}]_v$ by (ax2), which proves $\vdash \sigma(\mathcal{G}) \Rightarrow \Box[P]_v$. On the other hand, we know $\vdash \sigma(\mathcal{G}) \Rightarrow \neg\Box[P]_v$ by assumption and reach a contradiction.

These two claims ensure that for every node $\mathcal{G}$ in $\mathcal{T}(\mathcal{F})$ that contains either $\neg\Box F$ or $\neg\Box[P]_v$ there exists some node $\mathcal{G}'$ reachable from $\mathcal{G}$ that satisfies the condition from the definition of a complete path. For if $\mathcal{G}$ itself does not satisfy the condition, the formula is contained in $\sigma(\mathcal{G})$, hence $\mathcal{T}(\sigma(\mathcal{G}))$, which is just the subgraph of $\mathcal{T}(\mathcal{F})$ whose roots are the sons of $\mathcal{G}$, contains a node as required.

The assertion is now proved by fixing some order on the finite set of formulas $\neg\Box F$ and $\neg\Box[P]_v$ that occur in $\mathcal{T}(\mathcal{F})$ and an iterative construction that constructs a complete path piecewise by repeatedly considering the eventuality formulas in the chosen order. The details of this construction are standard [8, 9]. ∵

Lemma 14. *Assume that $\mathcal{F}$ is a finite and consistent set of formulas and that $\mathcal{G}_0, \mathcal{G}_1, \ldots$ is a complete path in $\mathcal{T}(\mathcal{F})$. For every $i \geq 0$, the following assertions hold:*

1. *If $\circ F \in \tau(\mathcal{G}_i)$ then $\circ F \in \mathcal{G}_i$ iff $F \in \mathcal{G}_{i+1}$.*
2. *If $\Box F \in \tau(\mathcal{G}_i)$ then $\Box F \in \mathcal{G}_i$ iff $F \in \mathcal{G}_j$ for all $j \geq i$.*
3. *If $\Box[P]_v \in \tau(\mathcal{G}_i)$ then $\Box[P]_v \in \mathcal{G}_i$ iff for all $j \geq i$, $P \in \mathcal{G}_j$ or $\{v, \circ v\} \subseteq \mathcal{G}_j$ or $\{\neg v, \neg\circ v\} \subseteq \mathcal{G}_j$.*

Proof. 1. If $\circ F \in \mathcal{G}_i$ then $F \in \sigma(\mathcal{G}_i)$ and therefore $F \in \mathcal{G}_{i+1}$, which is a completion of $\sigma(\mathcal{G}_i)$.

If $\circ F \notin \mathcal{G}_i$ then $\neg\circ F \in \mathcal{G}_i$ (because $\mathcal{G}_i$ is complete), so $\neg F \in \sigma(\mathcal{G}_i)$, and again $\neg F \in \mathcal{G}_{i+1}$. The consistency of $\mathcal{G}_{i+1}$ implies $F \notin \mathcal{G}_{i+1}$.

2. Assume $\Box F \in \mathcal{G}_i$. Then we have $F \in \tau(\mathcal{G}_i)$, and because of $\vdash \Box F \Rightarrow F$ (ax1) and proposition 10.3, it follows that $F \in \mathcal{G}_i$. Moreover, $\Box F \in \sigma(\mathcal{G}_i)$ and therefore $\Box F \in \mathcal{G}_{i+1}$. Inductively, we conclude that $F \in \mathcal{G}_j$ holds for all $j \geq i$.
   Conversely, if $F \in \mathcal{G}_j$ for all $j \geq i$ then the definition of a complete path and the consistency of the $\mathcal{G}_j$ ensure that $\neg\Box F \in \mathcal{G}_i$ cannot hold. The assumption $\Box F \in \tau(\mathcal{G}_i)$ and the fact that $\mathcal{G}_i$ is complete imply $\Box F \in \mathcal{G}_i$.
3. Assume $\Box[P]_v \in \mathcal{G}_i$. Then $\{P, v, \circ v\} \subseteq \tau(\mathcal{G}_i)$, and by $\mathrel{|\!\sim} \Box[P]_v \Rightarrow [P]_v$ (pax4) and proposition 10.3, the assertion follows for $j = i$ using the completeness and consistency of $\mathcal{G}_i$ and propositional logic. Moreover, $\Box[P]_v \in \sigma(\mathcal{G}_i)$ and therefore $\Box[P]_v \in \mathcal{G}_{i+1}$. Inductively, the assertion follows for all $j \geq i$.
   Conversely, if $P \in \mathcal{G}_j$ or $\{v, \circ v\} \subseteq \mathcal{G}_j$ or $\{\neg v, \neg\circ v\} \subseteq \mathcal{G}_j$ holds for all $j \geq i$, the consistency of the $\mathcal{G}_j$ implies that there can be no $j \geq i$ such that $\neg P \in \mathcal{G}_j$ and either $\{v, \neg\circ v\} \subseteq \mathcal{G}_j$ or $\{\neg v, \circ v\} \subseteq \mathcal{G}_j$. Therefore, using the definition of a complete path, it follows that $\neg\Box[P]_v \in \mathcal{G}_i$ cannot hold, hence $\Box[P]_v \in \mathcal{G}_i$. $\because$

We now have all the bits and pieces to construct a model for a finite and consistent set $\mathcal{F}$ from $\mathcal{T}(\mathcal{F})$.

Lemma 15. *For every finite and consistent set $\mathcal{F}$ of formulas there is a behavior σ such that $\sigma \models F$ holds for all $F \in \mathcal{F}$.*

Proof. Assume that $\mathcal{F}$ is a finite and consistent set of formulas. Construct $\mathcal{T}(\mathcal{F})$ and choose some complete path $\mathcal{G}_0, \mathcal{G}_1, \ldots$ that starts at some root of $\mathcal{T}(\mathcal{F})$; such a path exists by lemma 13. Now define the behavior $\sigma = s_0 s_1 \ldots$ by $s_i(v) = \text{tt}$ iff $v \in \mathcal{G}_i$, for every $v \in \mathcal{V}$.

By induction on the structure of (pre-)formulas, we prove that for all (pre-)formulas A and all $i \geq 0$, if $A \in \tau(\mathcal{G}_i)$ then $\sigma|_i \mathrel{|\!\approx} A$ iff $A \in \mathcal{G}_i$.

Because of $\mathcal{F} \subseteq \mathcal{G}_0$ and $F \in \tau(\mathcal{F}) = \tau(\mathcal{G}_0)$ for every $F \in \mathcal{F}$, this in particular implies $\sigma \models F$ for all formulas $F \in \mathcal{F}$.

The inductive proof of the assertion is again standard; we only give a few cases:

$\Box[P]_v$: Assume $\Box[P]_v \in \tau(\mathcal{G}_i)$. Therefore, either $\Box[P]_v \in \mathcal{G}_i$ or $\neg\Box[P]_v \in \mathcal{G}_i$. In the former case, lemma 14.3 implies that, for all $j \geq i$, $P \in \mathcal{G}_j$ or $\{v, \circ v\} \subseteq \mathcal{G}_j$ or $\{\neg v, \neg\circ v\} \subseteq \mathcal{G}_j$. By induction hypothesis and lemma 14.1, this implies that, for all $j \geq i$, $\sigma|_j \mathrel{|\!\approx} P$ or $s_j(v) = s_{j+1}(v)$, and therefore $\sigma|_i \mathrel{|\!\approx} \Box[P]_v$.
If $\neg\Box[P]_v \in \mathcal{G}_i$, then the definition of a complete path ensures that for some $j \geq i$, we have $\neg P \in \mathcal{G}_j$ and either $\{v, \neg\circ v\} \subseteq \mathcal{G}_j$ or $\{\neg v, \circ v\} \subseteq \mathcal{G}_j$, and the induction hypothesis and lemma 14.1 ensure $\sigma|_i \mathrel{|\!\approx} \neg\Box[P]_v$.

$\circ F$: Assume $\circ F \in \tau(\mathcal{G}_i)$. By lemma 14.1, $\circ F \in \mathcal{G}_i$ iff $F \in \mathcal{G}_{i+1}$ iff (by induction hypothesis) $\sigma|_{i+1} \mathrel{|\!\approx} F$ iff $\sigma|_i \mathrel{|\!\approx} \circ F$. $\because$

Theorem 16 (Completeness). *For every formula F, if $\models F$ then $\vdash F$.*

Proof. Assume $\models F$. Then $\sigma \models \neg F$ holds for no behavior σ, and lemma 15 implies that $\{\neg F\}$ is inconsistent, that is $\mathrel{|\!\sim} \neg\neg F$, from which $\vdash F$ follows by theorem 8.1 and propositional logic. $\because$

(hx0)	F whenever F is tautological	(hx7)	$\Box[P]_v$ whenever P is tautological
(hx1)	$\Box F \Rightarrow F$	(hx8)	$\Box[\bigcirc\neg F \equiv \neg\bigcirc F]_v$
(hx2)	$\Box F \Rightarrow \Box[F]_v$	(hx9)	$\Box[\bigcirc(F \Rightarrow G) \Rightarrow (\bigcirc F \Rightarrow \bigcirc G)]_v$
(hx3)	$\Box F \Rightarrow \Box[\bigcirc\Box F]_v$	(hx10)	$\Box[\Box[P]_v \Rightarrow [P]_v]_w$
(hx4)	$\Box[F \Rightarrow \bigcirc F]_F \Rightarrow (F \Rightarrow \Box F)$	(hx11)	$\Box[P]_v \Rightarrow \Box[\bigcirc\Box[P]_v]_w$
(hx5)	$\Box[P \Rightarrow Q]_v \Rightarrow (\Box[P]_v \Rightarrow \Box[Q]_v)$	(hx12)	$\Box[[P]_v \wedge \bigcirc\Box[P]_v \Rightarrow \Box[P]_v]_w$
(hx6)	$\Box[v' \not\equiv v]_v$	(hx13)	$\Box[\bigcirc\Box F \Rightarrow \Box[\bigcirc F]_v]_w$
(hmp)	$F, F \Rightarrow G \vdash^{h} G$	(alw)	$F \vdash^{h} \Box F$

Fig. 2. The proof system Σ^h_{GTLA}.

5 A Homogeneous Axiomatization

The system Σ_{GTLA} is based on the auxiliary relation $\mid\!\sim$ besides the relation $\vdash$ that we are really interested in. One may argue that one could instead simply translate propositional (G)TLA to PTL and use any standard PTL proof system. Still, proofs may then contain PTL formulas such as $\Box\bigcirc F$ that are not even pre-formulas of GTLA. We now show that it is possible to eliminate the auxiliary relation $\mid\!\sim$ and define a "homogeneous" axiomatization of GTLA based on a single provability relation $\vdash^{h}$. The key observation is that in Σ_{GTLA}, a derived pre-formula can only be used via rule (sq) in the derivation of a formula. It therefore suffices to "box" the axioms (pax0)–(pax5) and rephrase (pre), (nex), and (pmp) accordingly. The proof system Σ^h_{GTLA} shown in figure 2 is based on this idea and some further simplifications. The following theorems and rules can be derived in Σ^h_{GTLA}; again, we refer to the full version [14] of this paper.

(H1)	$\Box[P]_v, \Box[P \Rightarrow Q]_v \vdash^{h} \Box[Q]_v$	(H2)	$F \vdash^{h} \Box[F]_v$
(H3)	$\Box[P \Rightarrow Q]_v, \Box[Q \Rightarrow R]_v \vdash^{h} \Box[P \Rightarrow R]_v$		
(H4)	$\Box[[P]_v \Rightarrow P]_v$	(H5)	$\Box[\Box F \Rightarrow \bigcirc\Box F]_v$

Again, it is easy to derive analogues of these rules where the "index" v is replaced by a finite set of atomic propositions, or by a GTLA formula.

We now prove that the two provability relations agree (where $\mathcal{F} \vdash^{h} F$ is defined in the obvious way). In particular, Σ^h_{GTLA} is also sound and complete. It is therefore a matter of taste and convenience which axiomatization to use. The homogeneous proof system is aesthetically more satisfactory, but the heterogeneous system may be easier to use. (This is why the completeness proof was given for Σ_{GTLA}.)

Theorem 17. *For any set $\mathcal{F}$ of formulas and any formula F, $\mathcal{F} \vdash F$ iff $\mathcal{F} \vdash^{h} F$.*

Proof. "only if": By induction on the length of the assumed derivation in Σ_{GTLA}, we prove that $\mathcal{F} \vdash^{h} F$ whenever $\mathcal{F} \vdash F$ and that $\mathcal{F} \vdash^{h} \Box[P]_v$, for all atomic propositions v, whenever $\mathcal{F} \mid\!\sim P$, for any pre-formula P.

If F is from $\mathcal{F}$ or if it is an instance of (ax0), (ax1), (ax3), (ax4) or (ax5) then the assertion holds trivially because these axioms are also contained in $\varSigma^h_{\mathrm{GTLA}}$. Axiom (ax2) is derived in $\varSigma^h_{\mathrm{GTLA}}$ as follows:

(1)	$\Box[\Box F \Rightarrow \bigcirc\Box F]_F$	(H5)
(2)	$\Box F \Rightarrow \Box\Box F$	(1)(hx4)(mp)
(3)	$\Box\Box F \Rightarrow \Box[\Box F]_v$	(hx2)
(4)	$\Box F \Rightarrow \Box[\Box F]_v$	(prop)(2)(3)

If the last step in the derivation of $\mathcal{F} \vdash F$ is an application of (mp) to previously derived formulas G and $G \Rightarrow F$ then by induction hypothesis we have $\mathcal{F} \overset{h}{\vdash} G$ and $\mathcal{F} \overset{h}{\vdash} G \Rightarrow F$, so $\mathcal{F} \overset{h}{\vdash} F$ follows by rule (hmp).

If the last step in the derivation of $\mathcal{F} \vdash F$ is an application of (sq) to some previously derived pre-formula P (so F is $\Box[P]_v$) then by the induction hypothesis for the second assertion we already have $\mathcal{F} \overset{h}{\vdash} \Box[P]_v$.

The second assertion is trivial if the last step in the derivation of $\mathcal{F} \mathrel{|\!\!\sim} P$ is an instance of (pax0), (pax1), (pax2) or (pax5) because $\varSigma_{\mathrm{GTLA}}$ contains corresponding axioms. The case of (pax3) is taken care of by (H5). As for (pax4), it could obviously be replaced by

(pax4a)	$\mathrel{|\!\!\sim} \Box[P]_v \Rightarrow [P]_v$
(pax4b)	$\mathrel{|\!\!\sim} \Box[P]_v \Rightarrow \bigcirc\Box[P]_v$
(pax4c)	$\mathrel{|\!\!\sim} [P]_v \wedge \bigcirc\Box[P]_v \Rightarrow \Box[P]_v$

without changing the set of pre-formulas derivable in $\varSigma_{\mathrm{GTLA}}$. The axioms (hx10) and (hx12) directly correspond to (pax4a) and (pax4c), so it remains to consider the case of (pax4b):

(1)	$\Box[P]_v \Rightarrow \Box[\bigcirc\Box[P]_v]_w$	(hx11)
(2)	$\Box[\Box[P]_v \Rightarrow \Box[\bigcirc\Box[P]_v]_w]_w$	(H2)(1)
(3)	$\Box[\Box[\bigcirc\Box[P]_v]_w \Rightarrow [\bigcirc\Box[P]_v]_w]_w$	(hx10)
(4)	$\Box[\Box[P]_v \Rightarrow [\bigcirc\Box[P]_v]_w]_w$	(H3)(2)(3)
(5)	$\Box[[\bigcirc\Box[P]_v]_w \Rightarrow \bigcirc\Box[P]_v]_w$	(H4)
(6)	$\Box[\Box[P]_v \Rightarrow \bigcirc\Box[P]_v]_w$	(H3)(4)(5)

Considering the rules, the case of (pmp) is handled by the induction hypothesis and (H1). If the last step in the derivation of $\mathcal{F} \mathrel{|\!\!\sim} P$ is an application of (pre), then P is actually a formula and has already been derived, so we may assume $\mathcal{F} \overset{h}{\vdash} P$ by induction hypothesis. We obtain $\mathcal{F} \overset{h}{\vdash} \Box[P]_v$ by (H2).

If the last step is an application of (nex), then P is $\bigcirc F$, for some previously derived formula F, and by induction hypothesis we may assume $\mathcal{F} \overset{h}{\vdash} F$. We continue as follows:

(1)	F	(ind.hyp.)
(2)	$\Box F$	(alw)(1)
(3)	$\Box[\circ\Box F]_v$	(2)(hx3)(hmp)
(4)	$\Box[\circ\Box F \Rightarrow \Box[\circ F]_v]_v$	(hx13)
(5)	$\Box[\Box[\circ F]_v]_v$	(H1)(3)(4)
(6)	$\Box[\Box[\circ F]_v \Rightarrow [\circ F]_v]_v$	(hx10)
(7)	$\Box[[\circ F]_v]_v$	(H1)(5)(6)
(8)	$\Box[\circ F]_v$	(7)(H4)(H1)

"if": The proof is again by induction on the assumed derivation of $\mathcal{F} \overset{h}{\vdash} F$. The cases of (hx0), (hx1), (hx4), (hx5), and (hx6) are trivial because Σ_{GTLA} contains the same axioms. For (hx7), (hx8), (hx9), (hx10), (hx12), and (hx13), the proof uses the corresponding axioms of Σ_{GTLA} and rule (sq). For (hmp) and (alw), the assertion follows from the induction hypothesis and rules (mp) and (alw), which is a derived rule in Σ_{GTLA}.

The axiom (hx2) is derived in Σ_{GTLA} as follows:

(1)	$\mid\!\sim \Box F \Rightarrow F$	(ax1)(pre)
(2)	$\vdash \Box[\Box F \Rightarrow F]_v$	(sq)(1)
(3)	$\vdash \Box[\Box F]_v \Rightarrow \Box[F]_v$	(2)(ax4)(mp)
(4)	$\vdash \Box F \Rightarrow \Box[\Box F]_v$	(ax2)
(5)	$\vdash \Box F \Rightarrow \Box[F]_v$	(prop)(3)(4)

The derivation of (hx3) is similar, using (pax3) instead of (ax1). The derivation of (hx11) is very similar to that of (T4) and is omitted. ∵

6 Quantification and Expressiveness

We have remarked in section 2 that propositional TLA is a sublanguage of GTLA whose pre-formulas are restricted to boolean combinations of primed and unprimed proposition symbols. On the other hand, GTLA can be considered as a sublanguage of PTL by removing the distinction between formulas and pre-formulas and considering $\Box[P]_v$ as a short-hand notation for the PTL formula $\Box(P \vee (\circ v \equiv v))$. Lamport's intention in introducing TLA was to allow the implementation relation between two descriptions of systems, even at different levels of abstraction, to be represented by model inclusion on the semantic side, and by validity of implication inside the logic [13]. Theorem 5 gives a formal expression to this intention, so GTLA satisfies Lamport's requirement.

Does GTLA add any undesired expressiveness to TLA? We will now show that this is not the case by proving that TLA and GTLA become equi-expressive once we add quantification over atomic propositions.

We introduce two auxiliary relations on behaviors that are used in a stuttering-invariant semantics of quantification over atomic propositions.

Definition 18. *For $v \in \mathcal{V}$ we define the relations $=_v$ and $\approx_v$ on behaviors as follows:*

1. *Two behaviors $\sigma = s_0 s_1 \dots$ and $\tau = t_0 t_1 \dots$ are* equal up to v, *written $\sigma =_v \tau$ if $s_i(w) = t_i(w)$ for all $i \geq 0$ and $w \in \mathcal{V}$, except possibly v.*
2. *The relation $\approx_v$, called* similarity up to v, *is defined as $\approx_v = (\simeq \circ =_v \circ \simeq)$, where $\simeq$ is stuttering equivalence and $\circ$ denotes relational composition.*

Proposition 19.

1. *For any $v \in \mathcal{V}$, the relations $=_v$ and $\approx_v$ are equivalence relations.*
2. *$(\simeq_V \circ \approx_v) = (\approx_v \circ \simeq_{V \cup \{v\}})$, for any $v \in \mathcal{V}$ and $V \subseteq \mathcal{V}$.*

We now extend GTLA by quantification over atomic propositions. Conceptually, existential quantification corresponds to the hiding of state components in specifications. Following Lamport, we use a bold quantifier symbol $\boldsymbol{\exists}$ to emphasize that its semantics is non-standard, which helps to preserve stuttering invariance.

Definition 20 ($\boldsymbol{\exists}$-GTLA).

1. *Formulas and pre-formulas of* $\boldsymbol{\exists}$-GTLA *are given inductively as in definition 1, except by adding the following clause:*
   6. *If F is a formula and $v \in \mathcal{V}$ then $\boldsymbol{\exists}\, v : F$ is a formula.*
2. *The semantics of* $\boldsymbol{\exists}$-GTLA *is obtained by adding the following clause to definition 2.*
   $\sigma \models \boldsymbol{\exists}\, v : F$ *iff* $\tau \models F$ *holds for some $\tau \approx_v \sigma$.*

For a formula $F \equiv \boldsymbol{\exists}\, v : G$, we define the set $At(F)$ as $At(G) \setminus \{v\}$, since v becomes bound by the quantifier. Our definition of the semantics of quantification agrees with that of Lamport [12] who motivates it by showing that a naive definition would not preserve stuttering invariance. In fact, $\boldsymbol{\exists}$-GTLA is again insensitive to stuttering:

Theorem 21. *For any* $\boldsymbol{\exists}$-GTLA *formula F and behaviors σ, τ such that $\sigma \simeq_{At(F)} \tau$, we have $\sigma \models F$ iff $\tau \models F$.*

Proof. Extending the proof of theorem 5, we need only consider the case of a quantified formula $F \equiv \boldsymbol{\exists}\, v : G$. So assume that $\sigma \models F$ and that $\tau \simeq_{At(F)} \sigma$. Choose some behavior $\rho \approx_v \sigma$ such that $\rho \models G$, by the definition of $\sigma \models \boldsymbol{\exists}\, v : G$. Then $\tau \; (\simeq_{At(F)} \circ \approx_v) \; \rho$, and by proposition 19 it follows that $\tau \; (\approx_v \circ \simeq_{At(F) \cup \{v\}}) \; \rho$, which in turn implies $\tau \; (\approx_v \circ \simeq_{At(G)}) \; \rho$, because $\simeq_{At(F) \cup \{v\}} \subseteq \simeq_{At(G)}$. Hence, there exists some behavior π such that $\tau \approx_v \pi$ and $\pi \simeq_{At(G)} \rho$. By induction hypothesis it follows that $\pi \models G$, and thus $\tau \models F$ as required. ∵

The semantics of quantified formulas is defined for $\boldsymbol{\exists}$-GTLA in the same way as for TLA. It is therefore immediate that quantified propositional TLA is again a sublogic of $\boldsymbol{\exists}$-GTLA. We now show that the two logics are equally expressive by effectively constructing an equivalent (quantified) TLA formula for every $\boldsymbol{\exists}$-GTLA formula.

Theorem 22. *For every* $\boldsymbol{\exists}$-GTLA *formula F there is a TLA formula F^{TLA} such that for every behavior σ, $\sigma \models F$ iff $\sigma \models F^{\mathrm{TLA}}$.*

Proof. In a first step, eliminate all quantified subformulas of F by successively choosing a fresh atomic proposition u for every (innermost) subformula $\exists v : G$ of F, and replacing F by $\exists u : \Box(u \equiv \exists v : G) \wedge F^*$, where F^* is obtained from F by replacing the subformula $\exists v : G$ by u. It is easy to see that the resulting formula is equivalent to the original formula F.

If F does not contain any quantified subformulas except those introduced above, the final formula F^* and every formula G in $\exists v : G$ is translated as follows: choose a new atomic proposition v_H for every (topmost) non-atomic formula H such that H or $\circ H$ occurs inside a subformula $\Box[P]_v$. If $v_{H_1}, \ldots v_{H_n}$ are all the atomic propositions added in this way, replace the formula G under consideration by the TLA formula

$$\exists v_{H_1}, \ldots v_{H_n} : \Box(v_{H_1} \equiv H_1) \wedge \ldots \wedge \Box(v_{H_n} \equiv H_n) \wedge G^\dagger$$

where $G^\dagger$ results from G by replacing H_i by v_{H_i}, $\circ H_i$ by v'_{H_i}, and all remaining preformulas $\circ u$ by u'.

For example, if F is the formula

$$\Box[\Box v \Rightarrow \circ\exists w : \Box[u \Rightarrow \circ\Box w]_u]_v$$

the first step produces

$$\exists x : \Box(x \equiv \exists w : \Box[u \Rightarrow \circ\Box w]_u) \wedge \Box[\Box v \Rightarrow \circ x]_v$$

and F^{TLA} is the TLA formula

$$\exists x : \Box(x \equiv \exists w, y : \Box(y \equiv \Box w) \wedge \Box[u \Rightarrow y']_u) \wedge \exists z : \Box(z \equiv \Box v) \wedge \Box[z \Rightarrow x']_v$$

Given a behavior $\sigma = s_0 s_1 \ldots$, define the behavior $\tau = t_0 t_1 \ldots$ such that, for all $i \geq 0$, s_i and t_i agree on all propositions, except possibly on $v_{H_1}, \ldots v_{H_n}$, and where $t_i(v_{H_j}) = \mathrm{tt}$ iff $\sigma|_i \models H_j$. The assertion now follows from the following fact, which is proved by structural induction: For any subformula H of G, $\sigma|_i \models H$ iff $\tau|_i \models H^\dagger$ where $H^\dagger$ is obtained from H in the same way as $G^\dagger$ is obtained from G. ∵

For the GTLA formula $\Diamond\big\langle A \wedge \langle B\rangle_v\big\rangle_v$ considered in section 1, the procedure outlined in the proof of theorem 22 produces the TLA formula

$$\exists x : \Box(x \equiv \langle B\rangle_v) \wedge \Diamond\langle A \wedge x\rangle_v$$

7 Conclusion

The logic GTLA defined in this paper is a variant of Lamport's Temporal Logic of Actions. Like TLA, its formulas do not distinguish between behaviors that are stuttering equivalent. However, GTLA removes some apparently unnecessary restrictions on the syntax of formulas. We have also shown that the propositional fragment of GTLA admits a complete and reasonably simple axiomatization. In fact, our proof systems Σ_{GTLA} and Σ^h_{GTLA} are much simpler than Abadi's axiomatization [1] of a previous version of TLA. We have been careful to adhere to TLA as closely as possible. In particular, every TLA formula is a GTLA formula, and the two logics are equally expressive once we add (stuttering-invariant) quantification over flexible proposition symbols, as proposed by Lamport. By Rabinovich's result of expressive completeness for TLA [18], it follows that $\exists$-GTLA is expressively complete for all stuttering-invariant ω-languages definable in the monadic second-order theory of linear orders. We believe that GTLA is a more natural explanation of TLA's concepts. The difference between TLA and GTLA lies in the fact that in GTLA, formulas and pre-formulas are defined by mutual induction, whereas the syntax of TLA is defined in succeeding layers. In particular, GTLA allows temporal formulas to occur inside the $\Box[\_]_v$ operator. The fact that such formulas can already expressed in TLA via quantification over flexible variables (cf. the proof of theorem 22) is easily overlooked in the original definition of TLA. It will remain to be seen whether the added flexibility of GTLA is useful for writing system specifications.

There are alternative definitions of stuttering-invariant temporal logics. The easiest way to obtain invariance under stuttering is to interpret the $\circ$ operator of PTL not as referring to the immediate successor state, but to the first state in the future that differs in the valuation of some proposition (and to let $\circ F$ be true if no such state exists). The resulting logic is axiomatized by a minor variant of the standard PTL proof system, and it is "globally" stuttering-invariant with respect to $\simeq$, but not "locally" with respect to $\simeq_{At(F)}$, as determined by the formula under consideration. Unfortunately, "global" stuttering invariance is not enough to represent implementation by model inclusion. Another example for a globally stuttering-invariant logic is Pnueli's TLR [17]. The logic MTL defined by Mokkedem and Méry [16] is "locally" stuttering-invariant, but the authors did not prove a completeness result. On the other hand, one could obtain an axiomatization of TLA or GTLA by interpreting their formulas in PTL. However, this approach breaks when it comes to quantified formulas, due the stuttering-invariant definition of the semantics for $\exists$ (see also [18]).

GTLA is easily extended to a first-order logic where atomic propositions are replaced by atomic predicate-logic formulas, except for the "subscripts" v in formulas $\Box[P]_v$, which should then be state variables. (The generalization to arbitrary terms can be introduced as a short-hand notation as we have done in this paper.) Of course, one cannot hope for full completeness of first-order GTLA. Nevertheless, the ability to reason about the propositional fragment, together with some simple rules about (rigid) quantification has turned out to be extremely useful in the application of standard linear-time temporal logic, and we believe the same to be true for TLA.

References

[1] Martín Abadi. An axiomatization of Lamport's Temporal Logic of Actions. In Jos C. M. Baeten and Jan W. Klop, editors, *CONCUR '90, Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 57–69, Berlin, 1990. Springer-Verlag. A revised version is available on the Web at http://www.research.digital.com/SRC/personal/Martin_Abadi/allpapers.html.

[2] Martín Abadi and Leslie Lamport. The existence of refinement mappings. *Theoretical Computer Science*, 81(2):253–284, May 1991.

[3] Martín Abadi and Leslie Lamport. An old-fashioned recipe for real time. Research Report 91, Digital Equipment Corporation, Systems Research Center, 1992. An earlier version, without proofs, appeared in [7, pages 1–27].

[4] Martín Abadi and Leslie Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, May 1995.

[5] Martín Abadi and Stephan Merz. An abstract account of composition. In Jiří Wiedermann and Petr Hajek, editors, *Mathematical Foundations of Computer Science*, volume 969 of *Lecture Notes in Computer Science*, pages 499–508, Berlin, 1995. Springer-Verlag.

[6] Martín Abadi and Stephan Merz. On TLA as a logic. In Manfred Broy, editor, *Deductive Program Design*, NATO ASI series F, pages 235–272. Springer-Verlag, Berlin, 1996.

[7] J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg, editors. *Real-Time: Theory in Practice*, volume 600 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1992. Proceedings of a REX Real-Time Workshop, held in The Netherlands in June, 1991.

[8] Dov Gabbay, Amir Pnueli, S. Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In *Proceedings of the 7th Annual ACM Symposium on Principles of Programming Languages*, pages 163–173. ACM, 1980.

[9] Fred Kröger. *Temporal Logic of Programs*, volume 8 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1987.

[10] Leslie Lamport. What good is temporal logic? In R. E. A. Mason, editor, *Information Processing 83: Proceedings of the IFIP 9th World Congress*, pages 657–668, Paris, September 1983. IFIP, North-Holland.

[11] Leslie Lamport. Hybrid systems in TLA$^+$. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 77–102. Springer-Verlag, 1993.

[12] Leslie Lamport. The Temporal Logic of Actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, May 1994.

[13] Leslie Lamport. Refinement in state-based formalisms. Technical Note 1996–001, Digital Equipment Corporation, Systems Research Center, Palo Alto, California, December 1996.

[14] Stephan Merz. A more complete TLA. Technical Report, Institut für Informatik, Universität München. Available on the WWW at URL http://www.pst.informatik.uni-muenchen.de/~merz/papers/gtla.html, 1999.

[15] Stephan Merz. Isabelle/TLA. Available on the WWW at URL http://www.pst.informatik.uni-muenchen.de/~merz/isabelle/, 1997. Revised 1999.

[16] Abdelillah Mokkedem and Dominique Méry. A stuttering closed temporal logic for modular reasoning about concurrent programs. In *Temporal Logic (ICTL '94)*, volume 827 of *Lecture Notes in Computer Science*, pages 382–397, Bonn, 1994. Springer-Verlag.

[17] Amir Pnueli. System specification and refinement in temporal logic. In R.K. Shyamasundar, editor, *Foundations of Software Technology and Theoretical Computer Science*, volume 652 of *Lecture Notes in Computer Science*, pages 1–38. Springer-Verlag, 1992.

[18] Alexander Rabinovich. Expressive completeness of temporal logic of action. In L. Brim, J. Gruska, and J. Zlatuska, editors, *Mathematical Foundations of Computer Science*, volume 1450 of *Lecture Notes in Computer Science*, Brno, Czech Republic, August 1998. Springer-Verlag.