Verification in Continuous Time by Discrete Reasoning*

Luca de Alfaro and Zohar Manna

Computer Science Department
Stanford University
Stanford, CA 94305, USA
{luca,zm}@cs.stanford.edu

1 Introduction

There are two common choices for the semantics of real-time and hybrid systems. The first is a *discrete semantics*, in which the temporal evolution of the system is represented as an enumerable sequence of snapshots, each describing the state of the system at a certain time. The second is a *continuous semantics*, in which the system evolution is represented by a sequence of intervals of time, together with a description of the system state during each interval.

The weakness of the discrete semantics is that the sequence of snapshots does not correspond directly to the physical behavior of the system [6]. Moreover, the truth of temporal logic formulas depends not only on the behavior of the system, but also on the way in which the behavior has been sampled to produce the sequence of snapshots. In other words, the discrete semantics is not *sample invariant* [12]. What has made the discrete semantics popular, however, is the simplicity and power of the verification rules that can be formulated for temporal logic [7, 17]. The continuous semantics overcomes the weaknesses of the discrete one, but no comparable verification rules have been formulated for it.

This paper shows how the advantages of both semantics can be combined by adapting the simple verification rules of the discrete semantics to the continuous one. Specifically, we show that if a temporal logic formula has the property of *finite variability* (FV), its validity in the discrete semantics implies its validity in the continuous one. Thus, verification rules that have as conclusion a FV formula are sound also in the continuous semantics. Most formulas that arise in practice are FV, and we give criteria that help to characterize them. The ability to transfer results between the semantics, together with a deductive system for the continuous semantics, provides a powerful methodology for verifying temporal logic properties of real-time and hybrid systems in the continuous semantics.

We first present the case for real-time systems, and then we extend the results to hybrid systems.

* This research was supported in part by the National Science Foundation under grant CCR-92-23226, by the Defense Advanced Research Projects Agency under contract NAG2-892, and by the United States Air Force Office of Scientific Research under contract F49620-93-1-0139.

2 Real-Time Systems

Real-time systems will be modeled by *timed transition systems* [7, 15]. A timed transition system $S = \langle \mathcal{V}, \Sigma, \Theta, \mathcal{T}, L, U \rangle$ consists of the following components.

1. A set $\mathcal{V}$ of variables called *state variables*, where each variable has its own type.
2. A set Σ of states: each state $s \in \Sigma$ is a type-consistent interpretation of all the variables in $\mathcal{V}$. We denote by $s(x)$ the value at state s of variable x, for every $x \in \mathcal{V}$.
3. A set $\Theta \subseteq \Sigma$ of initial states. Θ has an associated assertion $\Theta_f(\mathcal{V})$, such that $\Theta = \{s \in \Sigma \mid s \models \Theta_f\}$.
4. A set $\mathcal{T}$ of transitions, where $\tau \subseteq \Sigma \times \Sigma$ for all $\tau \in \mathcal{T}$. Each $\tau \in \mathcal{T}$ has an associated assertion $\rho_\tau(\mathcal{V}, \mathcal{V}')$ such that $\tau = \{(s, s') \mid (s, s') \models \rho_\tau\}$, where (s, s') interprets $x \in \mathcal{V}$ as $s(x)$ and x' as $s'(x)$.
5. Two sets L, U of minimum and maximum delays of transitions, s.t. $0 \leq l_\tau \leq u_\tau \leq \infty$ for all $\tau \in \mathcal{T}$.

We denote by c_τ the enabling condition of transition τ, defined by $c_\tau = \{s \mid \exists s'.\ (s, s') \in \tau\}$. If $s \in c_\tau$, we say that transition τ is enabled in state s. We will assume that transitions are self-disabling: $(s, s') \in \tau \rightarrow s' \notin c_\tau$.

The temporal behavior of a real-time system will be represented by *discrete traces* or by *continuous traces*, thus giving rise to the discrete and continuous semantics [6, 15, 17].

Definition 1 (discrete trace). A *discrete trace* σ_{d} is an enumerable sequence of observations $\langle s_0, t_0 \rangle, \langle s_1, t_1 \rangle, \langle s_2, t_2 \rangle, \ldots$, with $s_n \in \Sigma$, $t_n \in \mathbb{R}^+$ for $n \in \mathbb{N}$, such that $t_0 = 0$, $\lim_{n \to \infty} t_n = \infty$, and $t_n \leq t_{n+1}$, $(t_n = t_{n+1}) \vee (s_n = s_{n+1})$ for all $n \in \mathbb{N}$. A *moment* of σ_{d} is simply an integer $n \in \mathbb{N}$. □

In the continuous semantics, the behavior of the system is represented by a mapping from intervals of non-negative real numbers (representing intervals of time) to states of the system. The intervals can overlap at most at the endpoints [6, 8, 2].

Definition 2 (continuous trace). A *continuous trace* σ_{c} is a sequence of pairs σ_{c}: $\langle s_0, I_0 \rangle$, $\langle s_1, I_1 \rangle$, $\langle s_2, I_2 \rangle$, $\ldots$, where I_n is an interval of $\mathbb{R}$ (open or closed at the endpoints) and $s_n \in \Sigma$ for all $n \in \mathbb{N}$, such that $\bigcup_{n \in \mathbb{N}} I_n = \mathbb{R}^+$ and $\sup I_n = \inf I_{n+1}$ for all $n \in \mathbb{N}$. For every $n \in \mathbb{N}$, either $s_n = s_{n+1}$, or I_n must be right-closed and I_{n+1} left-closed. A *moment* of a continuous trace σ_{c} is a pair (n, t) such that $t \in I_n$. The moments are ordered in lexicographic order: $(n, t) \leq (n', t')$ iff $n < n' \vee (n = n' \wedge t \leq t')$. □

Definition 3 (admission). A timed transition system S *admits* a trace σ, written $S \triangleright \sigma$, if the following conditions are satisfied.

1. For all $n \in \mathbb{N}$, either $s_n = s_{n+1}$ or $(s_n, s_{n+1}) \in \tau$ for a transition $\tau \in \mathcal{T}$ that has been continuously enabled at least for its minimum delay l_τ.
2. Transitions are never continuously enabled for a time longer than their maximum delay. □

3 Logic and Verification

Temporal logic. It is possible to define linear-time temporal logics for both the discrete and the continuous semantics. In both cases, we will use the same syntax, but the logics will have two different sets of models. We denote by $\mathrm{TL_D}$ the logic defined for the discrete semantics, and by $\mathrm{TL_C}$ the logic defined for the continuous semantics.

Syntax. The logic we consider includes the future temporal operators $\Box$, $\Diamond$, $\mathcal{U}$, and the corresponding past operators $\boxminus$, $\diamondminus$, $\mathcal{S}$ [7, 16]. However, the logic does not include the next and previous time operators $\bigcirc$, $\ominus$. The logic includes the *age function* Γ: for a formula ϕ, at any point in time, the term $\Gamma(\phi)$ denotes for how long in the past ϕ has been continuously true [17]. The language also includes the special symbol T, whose value represents the global time. Thus, for any point in time, the equality $T = \Gamma(true)$ holds.

The logic allows quantification over *rigid* variables, whose value is constant in time [5]. We use Greek letters, like ξ, ζ, for the rigid variables and Latin letters, like x, y, for the state variables of the system.

Semantics. The truth of temporal logic formulas is evaluated with respect to a model and an interpretation. The models of $\mathrm{TL_D}$ are the discrete traces, the models of $\mathrm{TL_C}$ are the continuous traces. The interpretation $\mathcal{I}$ assigns the value $\mathcal{I}(\xi)$ to each rigid variable ξ. In both cases, the truth of a formula is evaluated at each moment of a trace. We write $\mathcal{I}, \sigma_\mathrm{d} \models^\mathrm{D}_n \phi$ (resp. $\mathcal{I}, \sigma_\mathrm{c} \models^\mathrm{C}_{(n,t)} \phi$) to denote that, under the interpretation $\mathcal{I}$, ϕ is true at moment n of σ_d (resp. (n,t) of σ_c). If the interpretation $\mathcal{I}$ is omitted, the above statement holds under all possible interpretations.

In our notation, $\sigma_\mathrm{d} \models^\mathrm{D} \phi$ (resp. $\sigma_\mathrm{c} \models^\mathrm{C} \phi$) means that ϕ is true in every moment of σ_d (resp. σ_c): this is called *floating* semantics, and differs from the *anchored* semantics of [16], where $\sigma \models \phi$ means that ϕ is true in the *first* moment of σ. This leads to a simpler axiomatization of $\mathrm{TL_C}$.

Validity of the formula ϕ in $\mathrm{TL_D}$ (resp. $\mathrm{TL_C}$) is denoted by $\models^\mathrm{D} \phi$ (resp. $\models^\mathrm{C} \phi$), and the provability relation is denoted by $\vdash^\mathrm{D}$ (resp. $\vdash^\mathrm{C}$). We say that a system S satisfies a formula ϕ of $\mathrm{TL_D}$ (resp. $\mathrm{TL_C}$) if all the traces of S satisfy ϕ, and we denote this by $S \models^\mathrm{D} \phi$ (resp. $S \models^\mathrm{C} \phi$).

An Overview of Verification

The logics $\mathrm{TL_D}$ and $\mathrm{TL_C}$ have different sets of valid formulas [18]. From the point of view of system specification, $\mathrm{TL_C}$ is preferable, as the continuous semantics corresponds more closely to the physical behavior of the system. The following is an example of a formula expressing a $\mathrm{TL_C}$-valid property of time that is intuitively obvious, yet is not valid in $\mathrm{TL_D}$ — in fact, its negation is valid in $\mathrm{TL_D}$!

Example 1. The formula

$$\phi_1 \ : \ \forall\xi\,\forall\zeta\,\left[\Diamond(T=\xi)\wedge\Diamond(T=\zeta)\rightarrow\Diamond\left(T=\frac{\xi+\zeta}{2}\right)\right]$$

expressing the density of time is such that $\models^{\mathrm{C}} \phi_1$, $\models^{\mathrm{D}} \neg\phi_1$. □

The methods proposed in [7, 15, 17] to verify properties written in TL_{D} rely on two concepts: *verification conditions* and *verification rules*. The proof of the soundness of the verification conditions and of the verification rules makes an essential use of the discreteness of the semantics. Therefore, this approach cannot be immediately transferred to TL_{C}.

In this paper, we propose a methodology for verifying system properties expressed in the continuous semantics based on two components.

The first component consists of the SFV-theorem, relating the discrete and continuous semantics. The SFV-theorem states that if a temporal formula ϕ has the property of *syntactic finite variability*, or SFV, then $S \models^{\mathrm{D}} \phi$ implies $S \models^{\mathrm{C}} \phi$. This will allow us to use existing verification rules of TL_{D}, and then transfer their conclusions to TL_{C}.

The second component is a deductive system for TL_{C} that includes axioms about time. This deductive system can be used to construct a proof of the system specification from the time axioms and from the conclusions of the verification rules. The reasoning in TL_{C} can often be kept to a minimum, if desired.

Related work. A related approach to proving $S \models^{\mathrm{C}} \phi$ has been proposed in [6] for similar semantics and logics. It suggests *rephrasing* the property ϕ into a form ϕ' which is suited for the discrete semantics. If the rephrasing is perfect, then $S \models^{\mathrm{D}} \phi' \leftrightarrow S \models^{\mathrm{C}} \phi$; otherwise, it is sometimes possible to find a stronger property ϕ' such that $S \models^{\mathrm{D}} \phi' \rightarrow S \models^{\mathrm{C}} \phi$. In [6] it is explained how to rephrase some formulas, and how to approximate others with stronger conditions. However, no methodology is given for verifying properties that have no useful rephrasing or strengthening, and the case of hybrid systems is not considered.

4 Relationship Between the Semantics and SFV-Theorem

We say that a trace is a *refinement* of another if it has been obtained by sampling the state of the system more frequently in time [12, 13, 2].

Definition 4 (partitioning function). A *partitioning function* μ is a sequence μ_0, μ_1, μ_2, ... of adjacent and disjoint intervals of $\mathbb{N}$ covering $\mathbb{N}$. Formally, $\bigcup_{n\in\mathbb{N}} \mu_n = \mathbb{N}$, and $\max \mu_n = \min \mu_{n+1} - 1$ for all $n \in \mathbb{N}$. □

Definition 5 (refinement). A discrete trace σ'_{d}: $\langle s'_0, t'_0\rangle$, $\langle s'_1, t'_1\rangle$, $\langle s'_2, t'_2\rangle$, ... is a *refinement* of σ_{d}: $\langle s_0, t_0\rangle$, $\langle s_1, t_1\rangle$, $\langle s_2, t_2\rangle$, ... by the partitioning function μ, written $\sigma'_{\mathrm{d}} \succeq^{\mu} \sigma_{\mathrm{d}}$, if $t'_{\min \mu_n} = t_n$ and $s'_j = s_n$ for all $n \in \mathbb{N}$ and $j \in \mu_n$.

A continuous trace $\sigma'_{\rm c}$: $\langle r'_0, I'_0\rangle$, $\langle r'_1, I'_1\rangle$, $\langle r'_2, I'_2\rangle$, ... is a *refinement* of $\sigma_{\rm c}$: $\langle r_0, I_0\rangle$, $\langle r_1, I_1\rangle$, $\langle r_2, I_2\rangle$, ... by the partitioning function μ, written $\sigma'_{\rm c} \succeq^{\mu} \sigma_{\rm c}$, if $I_n = \bigcup_{j\in\mu_n} I'_j$ and $r'_j = r_n$ for all $n \in \mathbb{N}$ and $j \in \mu_n$.

We write $\sigma \succeq \sigma'$ if there is some μ such that $\sigma \succeq^{\mu} \sigma'$. □

If two traces have a common refinement, they are said to be *sample equivalent* [12, 10]. Sample equivalent traces correspond to different samplings of the same physical behavior of the system.

Definition 6 (sample equivalence). Two discrete (resp. continuous) traces σ, σ' are *sample equivalent*, written $\sigma \approx \sigma'$, if there is a discrete (resp. continuous) trace σ'' such that $\sigma'' \succeq \sigma$, $\sigma'' \succeq \sigma'$. □

Theorem 7. *If* $\sigma_{\rm d} \approx \sigma'_{\rm d}$, *then* $S \triangleright \sigma_{\rm d}$ *iff* $S \triangleright \sigma'_{\rm d}$. *If* $\sigma_{\rm c} \approx \sigma'_{\rm c}$, *then* $S \triangleright \sigma_{\rm c}$ *iff* $S \triangleright \sigma'_{\rm c}$.

We say that a temporal logic is *sample invariant* if $\sigma \approx \sigma'$ implies $\mathcal{I}, \sigma \models \phi \leftrightarrow \mathcal{I}, \sigma' \models \phi$. This means that the logic does not distinguish between sample equivalent traces, which is a desirable property. While it is known that $\mathrm{TL_D}$ is not sample invariant, it can be proved that $\mathrm{TL_C}$ is sample invariant.

Theorem 8 (sample invariance of $\mathrm{TL_C}$). *If* $\sigma'_{\rm c} \succeq^{\mu} \sigma_{\rm c}$ *and* $j \in \mu_i$, *then*

$$\mathcal{I}, \sigma'_{\rm c} \models^{\rm C}_{(j,t)} \phi \leftrightarrow \mathcal{I}, \sigma_{\rm c} \models^{\rm C}_{(i,t)} \phi \ .$$

for all $t \in I_i \cap I'_j$. *If* $\sigma'_{\rm c} \approx \sigma_{\rm c}$, *then* $\sigma'_{\rm c} \models \phi \leftrightarrow \sigma_{\rm c} \models \phi$.

Next, we define a translation from continuous to discrete traces that preserves admission and refinement.

Definition 9 ($\Omega : \sigma_{\rm c} \mapsto \sigma_{\rm d}$). The translation function Ω associates with $\sigma_{\rm c}$: $\langle r_0, I_0\rangle$, $\langle r_1, I_1\rangle$, $\langle r_2, I_2\rangle$, ... the discrete trace $\sigma_{\rm d}$: $\langle s_0, t_0\rangle$, $\langle s_1, t_1\rangle$, $\langle s_2, t_2\rangle$, ... such that, for all $n \in \mathbb{N}$, $s_{2n} = s_{2n+1} = r_n$, and:

1. if I_n is closed, then $t_{2n} = \inf I_n$, $t_{2n+1} = \sup I_n$;
2. if I_n is left open, then $t_{2n} = t_{2n+1} = \sup I_n$;
3. if I_n is right open, then $t_{2n} = t_{2n+1} = \inf I_n$;
4. if I_n is open, then $t_{2n} = t_{2n+1} = (\inf I_n + \sup I_n)/2$. □

Lemma 10. *If* $S \triangleright \sigma_{\rm c}$, *then* $S \triangleright \Omega(\sigma_{\rm c})$. *If* $S \triangleright \sigma_{\rm c}$ *and* $\sigma'_{\rm c} \succeq \sigma_{\rm c}$, *then* $S \triangleright \Omega(\sigma'_{\rm c})$ *and* $\Omega(\sigma'_{\rm c}) \succeq \Omega(\sigma_{\rm c})$.

We now define finite variability, the main property that will be used in establishing the connection between the discrete and the continuous semantics.

Definition 11 (finite variability). A formula ϕ has the property of *finite variability* (FV) if for every $\sigma_{\rm c}$ and every $\mathcal{I}$ there exists a $\sigma'_{\rm c} \succeq \sigma_{\rm c}$ such that

$$\mathcal{I}, \sigma'_{\rm c} \models^{\rm C}_{(i,t)} \psi \leftrightarrow \mathcal{I}, \sigma'_{\rm c} \models^{\rm C}_{(i,t')} \psi$$

for all subformulas ψ of ϕ, and for all pairs (i,t), (i,t') of moments belonging to the same interval I'_i of $\sigma'_{\rm c}$, for all $i \geq 0$. The trace $\sigma'_{\rm c}$ is called a ground trace for ϕ, $\sigma_{\rm c}$, $\mathcal{I}$. □

Example 2. The formula $\phi_2 : T < 4 \rightarrow \Diamond(\cos 1/(T-4) > 0)$ is not FV, as it is not possible to subdivide $\mathbb{R}^+$ into a finite number of intervals in which the subformula $\cos(1/(T-4)) > 0$ has constant value.

Also the formula ϕ_1 of Example 1 is not FV. In fact, for each value of ξ and ζ it is possible to find a σ'_c such that the subformulas $\xi = T$, $\zeta = T$ and $T = (\xi + \zeta)/2$ have constant value in the intervals. However, it is not possible to find a σ'_c that has this property for all possible values of ξ and ζ. □

The interest of finite variability lies in the following theorems, that relate the satisfaction relation of the two semantics. Note that the converse of Theorem 13 is not valid.

Theorem 12. *If σ'_c is a ground trace for ϕ, σ_c, $\mathcal{I}$, then for all $n \in \mathbb{N}$ and for all moments (n,t) of σ'_c,*

$$\mathcal{I}, \Omega(\sigma'_c) \models^{D}_{2n} \phi \;\leftrightarrow\; \mathcal{I}, \sigma'_c \models^{C}_{(n,t)} \phi \ .$$

Theorem 13. *If $S \models^{D} \phi$ and ϕ is FV, then $S \models^{C} \phi$. If $\models^{D} \phi$ and ϕ is FV, then $\models^{C} \phi$.*

Proof. We prove only the first statement, as the proof of the second is similar. We prove the counterpositive: assume $S \not\models^{C} \phi$. Then there are $\mathcal{I}$, σ_c and a moment (n,t) of σ_c such that $S \triangleright \sigma_c$, and $\mathcal{I}, \sigma_c \not\models_{(n,t)} \phi$. As ϕ is FV, there is a trace $\sigma'_c \succeq^{\mu} \sigma_c$ that is ground for ϕ, σ_c, $\mathcal{I}$. There is a $k \in \mu_n$ such that (k,t) is a moment of σ'_c, and from Theorem 8 we have that $\mathcal{I}, \sigma'_c \not\models_{(k,t)} \phi$. As σ'_c is ground for $\mathcal{I}$, ϕ, by Theorem 12 we have $\mathcal{I}, \Omega(\sigma'_c) \not\models_{2k} \phi$. Lemma 10 ensures that $S \triangleright \Omega(\sigma'_c)$, and we finally get $S \not\models^{D} \phi$, which concludes the proof. □

Syntactic finite variability. According to Definition 11, to show that a formula is FV it is necessary to give an argument for the existence of a ground trace for each trace and variable interpretation. As this might be complex, we propose instead a syntactic criterion, called *syntactic finite variability* (SFV), sufficient to establish that a formula is FV.

To be able to define SFV, we first define *well-behaved* functions as functions that are analytical (in the calculus sense) along the real axis in some of their variables. If a function $f(z)$ is analytical at the point b, there exists a region of positive radius centered on b in which $f(z)$ can be expanded as a series $f(z) = \sum_{n=0}^{\infty} c_n z^n$, where the coefficients c_n do not depend on z. This implies that around each zero of $f(z)$ there is a region of positive radius in which $f(z)$ has no other zero. This, in turn, leads to the well-known theorem of calculus stating that a function that is analytical along the real axis has at most a finite number of zeros in any finite interval of the real axis.

Definition 14 (well-behaved function). We say that a function

$$f(z_0, \ldots, z_n, v_1, \ldots, v_k) : \mathbb{R}^{n+k+1} \mapsto \mathbb{R}$$

is *well-behaved* if, for all $0 \leq i \leq n$, when considered as a function of z_i only, f is analytical along the real axis. □

Definition 15 (syntactic finite variability). The set of SFV formulas is defined inductively as follows.

1. If P is a predicate symbol and $u_1, \ldots, u_n$ are terms not containing T or Γ, then $P(u_1 \ldots u_n)$ is SFV.
2. If $f(z_0, \ldots, z_n, v_1, \ldots, v_k)$ is a well-behaved function, then

$$f(T, \Gamma(\phi_1), \ldots, \Gamma(\phi_n), c_1, \ldots, c_k) = 0 \ ,$$
$$f(T, \Gamma(\phi_1), \ldots, \Gamma(\phi_n), c_1, \ldots, c_k) > 0$$

   are SFV formulas, where $c_1, \ldots, c_k$ are constants or rigid variables of the logic (thus different from T), and $\phi_1, \ldots \phi_n$ are formulas not containing T or Γ. Such SFV formulas are called *T-atoms.*
3. A formula constructed from SFV formulas using propositional connectives or temporal operators is a SFV formula.
4. If ϕ is a SFV formula, and ξ does not occur in any T-atom of ϕ, then $\forall\xi.\phi$ is a SFV formula. □

If ϕ is SFV, the requirement that $f(z_0, \ldots, z_n, v_1, \ldots, v_k)$ is well-behaved insures that all the inequalities in ϕ, and thus all subformulas of ϕ, change truth value at most finitely often within each interval of a continuous trace. We will say that a formula is SFV even if it is not in a form described by the above definition, but can be easily transformed and put in such a form. SFV is a sufficient, but not necessary criterion for FV, but it is general enough to encompass many formulas used in the specification and verification of systems.

Example 3. The formula ϕ_2 of Example 2 is not SFV, as the function $\cos(1/(x-4))$ is not analytical at $x = 4$, a point of the real axis. The formula $T < 4 \rightarrow \Diamond[T = 4]$ is SFV. The formula ϕ_1 of Example 1 is not SFV, as it quantifies over ξ and ζ that appear in the T-atoms $T = \xi$, $T = \zeta$ and $T = (\xi + \zeta)/2$. □

Theorem 16 (SFV-theorem). *If ϕ is SFV, it is also FV. Hence, the following inference rules are sound, with the proviso that ϕ is SFV:*

$$\frac{S \vdash^{\mathrm{D}} \phi}{S \vdash^{\mathrm{C}} \phi} \ , \qquad \frac{\vdash^{\mathrm{D}} \phi}{\vdash^{\mathrm{C}} \phi} \ , \qquad \frac{S \vdash^{\mathrm{D}}_0 \phi}{S \vdash^{\mathrm{C}}_{(0,0)} \phi} \ , \qquad \frac{\vdash^{\mathrm{D}}_0 \phi}{\vdash^{\mathrm{C}}_{(0,0)} \phi} \ .$$

Using syntactic finite variability, we can also establish a connection with propositional temporal logic. Let PTL be the propositional temporal logic of discrete linear time, with temporal operators $\mathcal{U}$, $\mathcal{S}$, $\Box$, $\boxminus$, $\Diamond$, $\diamondminus$, and based on the floating semantics. This logic is the same as the one presented in [14], except for the absence of the next and previous time operators $\bigcirc$, $\ominus$.

Theorem 17 (from PTL to $\mathrm{TL_C}$). *Let α be a formula of PTL, containing occurrences of the propositional letters $p_1, \ldots, p_n$. If $\models^{\mathrm{PTL}} \alpha[p_1, \ldots p_n]$, then*

$\models^{\mathrm{C}} \alpha[\phi_1, \ldots, \phi_n]$ *provided* ϕ_1, ..., ϕ_n *are FV. Similarly for initial validity. Therefore, the following inference rules*

$$\frac{\vdash^{\mathrm{PTL}} \alpha[p_1, \ldots p_n]}{\vdash^{\mathrm{C}} \alpha[\phi_1, \ldots, \phi_n]} \quad , \qquad \frac{\vdash_0^{\mathrm{PTL}} \alpha[p_1, \ldots p_n]}{\vdash^{\mathrm{C}}_{(0,0)} \alpha[\phi_1, \ldots, \phi_n]} \quad ,$$

with the proviso that ϕ_1, ..., ϕ_n *are SFV, are sound.*

This result is of interest, as deductive systems for PTL are well-studied [14], and efficient decision algorithms for the problem of initial validity exist [9].

5 Verification in the Continuous Semantic

We propose the following methodology for verifying properties expressed in $\mathrm{TL_C}$. Suppose that the property to be verified is $S \models^{\mathrm{C}} \phi$, or $S \models^{\mathrm{C}}_{(0,0)} \phi$.

First, we use the verification rules for the discrete semantics, like the ones proposed in [15, 7, 17], to establish temporal properties of the system that hold in $\mathrm{TL_D}$. Let $S \models_0^{\mathrm{D}} \psi_1, \ldots, S \models_0^{\mathrm{D}} \psi_k$ be the conclusions reached. Then, we can use a deductive system for $\mathrm{TL_D}$ to reason about $S \models_0^{\mathrm{D}} \psi_1, \ldots, S \models_0^{\mathrm{D}} \psi_k$. Although $\mathrm{TL_D}$ does not have a complete axiomatization [5, 1, 2], a deductive system for initial validity in $\mathrm{TL_D}$ is given in [16]. Results can be transferred from initial validity to global validity using the rules

$$\frac{\vdash_0^{\mathrm{D}} \Box\phi}{\vdash^{\mathrm{D}} \phi} \quad , \qquad \frac{S \vdash_0^{\mathrm{D}} \Box\phi}{S \vdash^{\mathrm{D}} \phi} \quad .$$

Let $S \models_0^{\mathrm{D}} \gamma_1, \ldots, S \models_0^{\mathrm{D}} \gamma_m$ be the conclusions reached by reasoning in $\mathrm{TL_D}$.

If the formulas $\gamma_1, \ldots, \gamma_m$ are SFV, the SFV-theorem enables us to conclude $S \models^{\mathrm{C}}_{(0,0)} \gamma_1, \ldots, S \models^{\mathrm{C}}_{(0,0)} \gamma_m$. We can then use temporal reasoning in $\mathrm{TL_C}$, if necessary, to reach the desired conclusion $S \models^{\mathrm{C}} \phi$ or $S \models^{\mathrm{C}}_{(0,0)} \phi$. $\mathrm{TL_C}$, like $\mathrm{TL_D}$, also lacks a complete axiomatization. An (incomplete) deductive system for $\mathrm{TL_C}$ is presented below.

Temporal reasoning in $\mathrm{TL_C}$ is usually limited to the use of axioms about the completeness and divergence of time. To prove $S \models^{\mathrm{C}} \phi$, it is often possible to find a formula $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$, where $\alpha_1, \ldots, \alpha_n$ are $\mathrm{TL_C}$-valid formulas stating the progress of time. Then, if it is possible to prove $S \models^{\mathrm{D}} \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$, the only reasoning necessary in $\mathrm{TL_C}$ is to use axioms about time to obtain $S \models^{\mathrm{C}} \phi$.

5.1 Deductive System for $\mathrm{TL_C}$

Propositional and first-order axioms. A list of axioms for propositional linear-time temporal logic with $\mathcal{U}$, $\mathcal{S}$ has been presented in [3]. The axiomatization given there refers to an irreflexive temporal logic, i.e. to a logic in which the future and the past do not include the present; but it can be adapted to a reflexive logic, such as $\mathrm{TL_C}$. A list of the adapted propositional axioms is given in [4], along with an entirely classical list of first-order axioms.

Table 1. Time axiom schemas for $\mathrm{TL_C}$.

$T \geq 0$	$\vdash^{\mathrm{C}}_{(0,0)} T = 0$	$T = \xi \ \rightarrow \ \Box(T \geq \xi)$
$\neg\phi \ \rightarrow \ \Gamma(\phi) = 0$	$0 \leq \Gamma(\phi) \leq T$	$\xi \geq T \ \rightarrow \ \Diamond(T = \xi)$
$T = \xi \ \wedge \ \boxminus\neg[\phi \,\mathcal{U}\, (T = \xi)] \ \rightarrow \ \Gamma(\phi) = 0$		
$(T = \xi + \zeta) \ \wedge \ \phi \,\mathcal{S}\, (T = \xi \wedge \Gamma(\phi) = \upsilon) \ \rightarrow \ \Gamma(\phi) = \upsilon + \zeta$		

Table 2. Inference rules. The rules denoted by (†) have the proviso that ξ must not occur free in ϕ. The rules denoted by (§) have the proviso that ϕ, ϕ_1, ..., ϕ_n are SFV. In all of them, if the premise(s) is (are) S-valid, the conclusion is S-valid.

$\dfrac{\vdash^{\mathrm{C}} \phi \rightarrow \psi, \quad \vdash^{\mathrm{C}} \phi}{\vdash^{\mathrm{C}} \psi}$	$\dfrac{\vdash^{\mathrm{C}}_{(0,0)} \phi \rightarrow \psi, \quad \vdash^{\mathrm{C}}_{(0,0)} \phi}{\vdash^{\mathrm{C}}_{(0,0)} \psi}$	$\dfrac{\vdash^{\mathrm{C}} \phi}{\vdash^{\mathrm{C}} \Box\phi}$	$\dfrac{\vdash^{\mathrm{C}} \phi}{\vdash^{\mathrm{C}} \boxminus\phi}$
$\dfrac{\vdash^{\mathrm{C}} \phi \rightarrow \psi}{\vdash^{\mathrm{C}} \phi \rightarrow \forall\xi\, \psi}(\dagger)$	$\dfrac{\vdash^{\mathrm{C}}_{(0,0)} \phi \rightarrow \psi}{\vdash^{\mathrm{C}}_{(0,0)} \phi \rightarrow \forall\xi\, \psi}(\dagger)$	$\dfrac{\vdash^{\mathrm{C}}_{(0,0)} \Box\phi}{\vdash^{\mathrm{C}} \Box\phi}$	$\dfrac{\vdash^{\mathrm{C}} \phi}{\vdash^{\mathrm{C}}_{(0,0)} \phi}$
$\dfrac{\vdash^{\mathrm{PTL}} \alpha[p_1, \ldots p_n]}{\vdash^{\mathrm{C}} \alpha[\phi_1, \ldots, \phi_n]}(\S)$	$\dfrac{\vdash^{\mathrm{PTL}}_0 \alpha[p_1, \ldots p_n]}{\vdash^{\mathrm{C}}_{(0,0)} \alpha[\phi_1, \ldots, \phi_n]}(\S)$	$\dfrac{\vdash^{\mathrm{D}} \phi}{\vdash^{\mathrm{C}} \phi}(\S)$	$\dfrac{\vdash^{\mathrm{D}}_0 \phi}{\vdash^{\mathrm{C}}_{(0,0)} \phi}(\S)$

Time axioms. The set of axioms listed in Table 1 is used to reason about time. As usual, we list an axiom ϕ to mean $\vdash^{\mathrm{C}} \phi$. In the case where we claim only the initial validity of the axiom, as in the case of the second one, we write it explicitly.

Inference rules. The inference rules we propose are listed in Table 2.

5.2 An Example of Verification

Imprecise oscillator. Consider the system OSC, consisting of an oscillator whose state is represented by the variable x. The oscillator can be in any of two states, $x = 0$ and $x = 1$, and it can stay in each of them for 3 to 5 seconds before switching to the other one. The oscillator starts in the state $x = 0$. The system can be described by:

$$\Theta_f : \ x = 0 \qquad \mathcal{T} : \ \{\tau_0, \tau_1\}$$

$$\begin{array}{lll} \rho_{\tau_0} : \ x = 0 \wedge x' = 1 & l_{\tau_0} : \ 3 & u_{\tau_0} : \ 5 \\ \rho_{\tau_1} : \ x = 1 \wedge x' = 0 & l_{\tau_1} : \ 3 & u_{\tau_1} : \ 5 \ . \end{array}$$

We want to verify that OSC satisfies the following property:

> *The oscillator is in the state $x = 1$ some time between 5 and 6 seconds after it is started.*

This property can be written as

$$\text{OSC} \models^{\text{C}}_{(0,0)} \Diamond(x = 1 \wedge 5 < T < 6) \ . \tag{1}$$

Note that the corresponding property in TL_D, $\text{OSC} \models^{\text{D}}_0 \Diamond(x = 1 \wedge 5 < T < 6)$, does not hold. To prove (1), define the abbreviations

$$\begin{aligned} \psi &: \ x = 0 \wedge [\Gamma(x = 0) = T] \wedge T \leq 3 \ , \\ \phi &: \ T \leq 5.5 \rightarrow [x = 1 \wedge \Gamma(x = 1) \leq T - 3] \ . \end{aligned}$$

The following implications hold: $\phi \wedge T = 5.5 \ \rightarrow \ x = 1$, $\psi \wedge T = 5.5 \ \rightarrow \ x = 1$. The proof of (1) proceeds as follows:

$\text{OSC} \vdash^{\text{D}}_0 \psi \, \mathcal{W} \, \phi$	from wait-for rule for TL_D	(2)
$\text{OSC} \vdash^{\text{D}} \phi \rightarrow \Box\phi$	from invariance rule for TL_D	(3)
$\text{OSC} \vdash^{\text{D}}_0 \psi \, \mathcal{W} \, \Box\phi$	from (3) by temporal reasoning in TL_D	(4)
$\text{OSC} \vdash^{\text{D}}_0 \Box(T = 5.5 \rightarrow x = 1)$	from (4) by temporal reasoning in TL_D	(5)
$\text{OSC} \vdash^{\text{C}}_{(0,0)} \Box(T = 5.5 \rightarrow x = 1)$	from (5), as it is SFV	(6)
$\text{OSC} \vdash^{\text{C}}_{(0,0)} \Diamond(T = 5.5)$	from the time axioms of TL_C	(7)
$\text{OSC} \vdash^{\text{C}}_{(0,0)} \Diamond(T = 5.5 \wedge x = 1)$	from (6), (7) by temp. reas. in TL_C	(8)
$\text{OSC} \vdash^{\text{C}}_{(0,0)} \Diamond(x = 1 \wedge 5 < T < 6)$	from (8).	(9)

6 Hybrid Systems

The results obtained for real-time systems can be transferred to hybrid systems, by giving a new definition of SFV to account for the fact that the state of the system can change continuously in time.

We will model hybrid systems by *phase transition systems* derived from those of [15, 17]. A phase transition system (PTS) $S = \langle \mathcal{V}, \Sigma, \mathcal{P}, \mathcal{T}, L, U, \Theta \rangle$ consists of the following components.

1. A set $\mathcal{V}$ of state variables. $\mathcal{V}$ is partitioned into two disjoint subsets: $\mathcal{V}_\text{d}$ and $\mathcal{V}_\text{c}$. The variables in $\mathcal{V}_\text{d}$ are called *discrete* variables, they can be of any type and they can change only in an instantaneous way. The variables in $\mathcal{V}_\text{c}$ are called *continuous* variables, have the real numbers as domain, and can change both in an instantaneous and in a continuous way.

2. A set Σ of states: each state is a type consistent interpretation of the variables. Again, we write $s(x)$ to denote the interpretation of $x \in \mathcal{V}$ at state s. We write $s|_{\mathcal{V}_\mathrm{d}}$, $s|_{\mathcal{V}_\mathrm{c}}$ to denote the restrictions of the interpretation s to only discrete and only continuous variables, respectively.
3. A set $\Theta \subseteq \Sigma$ of initial states.
4. A set $\mathcal{P}$ of *phases*. $\mathcal{P}$ is partitioned into disjoint subsets, one for each variable in $\mathcal{V}_\mathrm{c}$. The subset corresponding to $x \in \mathcal{V}_\mathrm{c}$ will be denoted by $\mathcal{P}_x$.
5. A set $\mathcal{T}$ of transitions, where $\tau \subseteq \Sigma \times \Sigma$ for each $\tau \in \Sigma$. $\mathcal{T}$ is partitioned into two disjoint subsets $\mathcal{T}_\mathrm{i}$ and $\mathcal{T}_\mathrm{d}$. The set $\mathcal{T}_\mathrm{i}$ is the set of *immediate* transitions, that must be executed no later than the time at which they become enabled. The set $\mathcal{T}_\mathrm{d}$ is the set of *delayed* transitions, whose enabling does not depend on the continuous variables.
6. Two sets L, U of minimum and maximum delays for the transitions in $\mathcal{T}_\mathrm{d}$.

Phases. For each $x \in \mathcal{V}_\mathrm{c}$, every phase $\wp \in \mathcal{P}_x$ consists of an *enabling condition* $c_\wp \subseteq \Sigma$ and a *phase function* $f_\wp : \Sigma \mapsto \mathbb{R}$. The phase $\wp$ is used to represent a differential equation governing x: the intended meaning is that if $c_\wp$ holds, then $\dot{x} = f_\wp(s)$ whenever no transition is taken. The enabling condition $c_\wp$ can depend on the discrete variables only: formally, for all $s, s' \in \Sigma$, $s|_{\mathcal{V}_\mathrm{d}} = s'|_{\mathcal{V}_\mathrm{d}} \rightarrow (s \in c_\wp \leftrightarrow s' \in c_\wp)$.

We say that a phase $\wp$ is *linear* if the function $f_\wp$ is a linear function of the continuous variables. It is not required that $f_\wp$ is linear in the discrete variables.

Transitions. We define the *enabling condition* c_τ of a transition $\tau \in \mathcal{T}$ as the set of states that have a successor according to the transition, or $c_\tau = \{s \mid \exists s'[(s, s') \in \tau]\}$. Transitions must be self-disabling, that is, $(s, s') \in \tau \rightarrow s' \notin c_\tau$.

If an immediate transition becomes enabled at time t, it has to be taken or disabled by some other transition before time advances past t. There are no restrictions on the enabling conditions of immediate transitions: they can depend on both the continuous and the discrete part of the state.

Each delayed transition $\tau \in \mathcal{T}_\mathrm{d}$ has an associated minimum delay $l_\tau \in L$ and maximum delay $u_\tau \in U$, with $0 \leq l_\tau \leq u_\tau \leq \infty$. After τ is enabled, it can wait for a time t_d: $l_\tau \leq t_d \leq u_\tau$ before being taken. The enabling condition of delayed transitions can depend only on the discrete component of the state: for all $s, s' \in \Sigma$, $\quad s|_{\mathcal{V}_\mathrm{d}} = s'|_{\mathcal{V}_\mathrm{d}} \rightarrow (s \in c_\tau \leftrightarrow s' \in c_\tau)$.

Continuous semantics. The continuous semantics of hybrid systems is defined in terms of *hybrid traces*. They differ from the continuous traces used for real-time systems, as the value of the continuous variables can change in the intervals composing the trace.

Definition 18 (hybrid trace). A *hybrid trace* σ_h is a sequence of pairs σ_h: $\langle g_0, I_0\rangle$, $\langle g_1, I_1\rangle$, $\langle g_2, I_2\rangle$, $\ldots$, where I_n is an interval of $\mathbb{R}$ and $g_n : I_n \mapsto \Sigma$, for all $n \in \mathbb{N}$. The intervals can overlap at most at the endpoints, and they cover all $\mathbb{R}^+$: $\bigcup_{n\in\mathbb{N}} I_n = \mathbb{R}^+$, and $\sup I_n = \inf I_{n+1}$ for all $n \in \mathbb{N}$.

Each function g_n gives the state of the system $g_n(t) \in \Sigma$ corresponding to time $t \in I_n$. The value of variable x at time t of interval I_n is thus $g_n(t)(x)$. The discrete variables must have constant value in an interval: for all $n \in \mathbb{R}$, all $t_1, t_2 \in I_n$, and all $x \in \mathcal{V}_\mathrm{d}$, $g_n(t_1)(x) = g_n(t_2)(x)$. Moreover, for all $x \in \mathcal{V}_\mathrm{d}$, $n \in \mathbb{N}$ and $t \in I_n$, $t' \in I_{n+1}$, we require that either $g_n(t)(x) = g_{n+1}(t')(x)$, or that I_n is right-closed and I_{n+1} is left-closed. □

Definition 19 (admission, hybrid traces). A PTS S *admits* a trace σ_h: $\langle g_0, I_0\rangle$, $\langle g_1, I_1\rangle$, $\langle g_2, I_2\rangle$, ..., written $S \triangleright \sigma_\mathrm{h}$, if the following conditions are satisfied:

1. All phases are respected: for each $x \in \mathcal{V}_\mathrm{c}$ and $n \in \mathbb{N}$, if $\inf I_n < \sup I_n$, there is a $\wp \in \mathcal{P}_x$ such that, for all $t \in I_n$:

$$g_n(t) \in c_\wp \ , \qquad f_\wp(g_n(t)) = \left.\frac{d\, g_n(u)(x)}{du}\right|_{u=t} \ , \tag{10}$$

   where it is assumed that for $\inf I_n < t < \sup I_n$ the derivative $dg_n(u)(x)/du$ is defined in $u = t$, and for $u = \inf I_n$, $u = \sup I_n$, the left-hand and right-hand derivatives, respectively, exist.
2. No immediate transition is skipped: for all $n \in \mathbb{N}$, $\tau \in \mathcal{T}_\mathrm{i}$, and $\inf I_n \le t < \sup I_n$, it is $g_n(t) \notin c_\tau$.
3. All discrete state changes are due to a transition: for all $n \in \mathbb{N}$, either $g_n(\sup I_n) = g_{n+1}(\inf I_{n+1})$ or there is $\tau \in \mathcal{T}$ such that

$$\big(g_n(\sup I_n), g_{n+1}(\inf I_{n+1})\big) \in \tau \ .$$

   If such a τ is a delayed transition, we also require that it has been continuously enabled for at least l_τ.
4. Delayed transitions are never continuously enabled for a time longer than their maximum delay. □

Discrete semantics. The discrete semantics of hybrid systems is defined in terms of discrete traces, which are defined as in Definition 1 except that the last clause is replaced by $(t_n = t_{n+1}) \vee (s_n|_{\mathcal{V}_\mathrm{d}} = s_{n+1}|_{\mathcal{V}_\mathrm{d}})$ for all $n \in \mathbb{N}$. The definition of admission of discrete traces is given in the next section.

Temporal logic. The temporal logic corresponding to discrete semantics is denoted by $\mathrm{TL_D}$, as before. The logic corresponding to the hybrid semantics will be denoted by $\mathrm{TL_H}$, its satisfaction relation will be denoted with $\models^\mathrm{H}$, and its provability relation with $\vdash^\mathrm{H}$.

7 From Discrete to Continuous Validity

Refinement of discrete traces was defined in Definition 5. Refinement of hybrid traces is defined as follows.

Definition 20 (refinement, hybrid traces). A hybrid trace σ_{h}: $\langle g_0, I_0\rangle$, $\langle g_1, I_1\rangle$, $\langle g_2, I_2\rangle$, ... is a *refinement* of σ'_{h}: $\langle g'_0, I'_0\rangle$, $\langle g'_1, I'_1\rangle$, $\langle g'_2, I'_2\rangle$, ... by the partitioning function μ, denoted $\sigma'_{\mathrm{h}} \succeq^{\mu} \sigma_{\mathrm{h}}$, if $I_i = \bigcup_{j\in\mu_i} I'_j$, and for every $i, j \in \mathbb{N}$ such that $j \in \mu_i$, it is $\forall t \in I'_j\big[g_j(t) = g_i(t)\big]$. □

Sampling equivalence is then defined as before. We then define a translation function Υ from discrete traces to hybrid traces. A discrete trace does not encode all the information required to reconstruct a hybrid trace: it contains the information about the state at the beginning and at the end of each closed interval, but it does not represent the evolution of the state in the interior of the interval. Therefore, to a single discrete trace will correspond all the hybrid traces that agree with the discrete one at the endpoints of the intervals.

Definition 21 ($\Upsilon : \sigma_{\mathrm{d}} \mapsto \sigma_{\mathrm{h}}$). The translation function Υ associates to σ_{d}: $\langle s_0, t_0\rangle$, $\langle s_1, t_1\rangle$, $\langle s_2, t_2\rangle$, ... a set of hybrid traces $\Upsilon(\sigma_{\mathrm{d}})$, such that, for every σ_{h}: $\langle g_0, I_0\rangle$, $\langle g_1, I_1\rangle$, $\langle g_2, I_2\rangle$, ... $\in \Upsilon(\sigma_{\mathrm{d}})$, and for every $n \in \mathbb{N}$, it is $\min I_n = t_n$, $\max I_n = t_{n+1}$, $g_n(\min I_n) = s_n$, $g_n(\max I_n) = s_{n+1}$. □

It is not possible to check directly whether a discrete trace respects the phases of a system, as in (10) for hybrid traces. Therefore, the definition of admission is indirect: a PTS S admits a discrete trace σ_{d} if there is a hybrid trace corresponding to σ_{d} admitted by S. This is the implicit meaning of the definition given in [17].

Definition 22 (admission, discrete traces). A PTS S *admits* a discrete trace σ_{d}, written $S \triangleright \sigma_{\mathrm{d}}$, if there is a $\sigma_{\mathrm{h}} \in \Upsilon(\sigma_{\mathrm{d}})$ such that $S \triangleright \sigma_{\mathrm{h}}$. □

In defining finite variability for hybrid systems, it is essential to define it with respect to a given PTS, so that the behavior of the continuous variables is constrained by the phases of the PTS. It is then possible to prove the corresponding of Theorem 16.

Definition 23 (hybrid finite variability). A formula ϕ is *hybrid finite variability* (HFV) with respect to a PTS S if for every σ_{h} admitted by S and every $\mathcal{I}$, there exists a $\sigma'_{\mathrm{h}} \succeq \sigma_{\mathrm{h}}$ such that: $\mathcal{I}, \sigma'_{\mathrm{c}} \models^{\mathrm{C}}_{(i,t)} \psi \leftrightarrow \mathcal{I}, \sigma'_{\mathrm{c}} \models^{\mathrm{C}}_{(i,t')} \psi$ for all subformulas ψ of ϕ, and for all pairs (i, t), (i, t') of moments belonging to the same interval I'_i of σ'_{h}, for all $i \geq 0$. □

Theorem 24. *If $S \models^{\mathrm{D}} \phi$ and ϕ is HFV with respect to S, then $S \models^{\mathrm{H}} \phi$. If $\models^{\mathrm{D}} \phi$ and ϕ is HFV with respect to S, then $S \models^{\mathrm{H}} \phi$. Similarly for initial validity.*

The property of *hybrid syntactic finite variability* (HSFV) provides a sufficient criterion for HFV, and is also defined with respect to a PTS S.

Definition 25 (simple age function). We say that an age function $\Gamma(\phi)$ is *simple* with respect to a system S if its argument ϕ contains neither T, nor Γ, nor occurrences of continuous state variables of S. □

Definition 26 (hybrid syntactic finite variability). A formula is *hybrid syntactic finite variability* (HSFV) with respect to a PTS S if the phases of S are linear, and if the formula is constructed in the following inductive way.

1. If P is a predicate symbol and $u_1, \ldots, u_n$ are terms not containing T, Γ, nor continuous state variables, then $P(u_1, \ldots, u_n)$ is HSFV.
2. If $f(z_0, \ldots, z_n, v_1, \ldots, v_k)$ is a well-behaved function, then
$$f(b_0, \ldots, b_n, c_1, \ldots, c_k) = 0 \ , \qquad f(b_0, \ldots, b_n, c_1, \ldots, c_k) > 0$$
are HSFV formulas, provided $b_0, \ldots, b_k$ are either constants, rigid variables, state variables (continuous or discrete), T or simple age functions, and $c_1, \ldots, c_k$ are rigid variables or constants. We call these types of HSFV formulas *HT-atoms.*
3. A formula constructed from HSFV formulas using propositional connectives or temporal operators is a HSFV formula.
4. If ϕ is a HSFV formula, and ξ does not occur in any HT-atom of ϕ, then $\forall \xi.\phi$ is a HSFV formula. □

The restriction requiring the linearity of the phases is important, and cannot be lifted without being substituted by some other kind of condition insuring that the solutions of the differential equations are well-behaved in the sense of Definition 14. Again, HSFV is a sufficient condition for finite variability of a formula with respect to a PTS, providing thus the connection between the discrete and continuous semantics.

Theorem 27 (from $\mathrm{TL_D}$ to $\mathrm{TL_H}$). *If ϕ is HSFV with respect to a PTS S, then it is SFV with respect to S. Therefore, the inference rules*
$$\frac{S \vdash^{\mathrm{D}} \phi}{S \vdash^{\mathrm{H}} \phi} \ , \qquad \frac{S \vdash_0^{\mathrm{D}} \phi}{S \vdash_{(0,0)}^{\mathrm{H}} \phi} \ ,$$
with the proviso that ϕ is HSFV with respect to S, are sound.

A deductive system for $\mathrm{TL_H}$. Since the definition of syntactic finite variability is now relative to a PTS, we need to modify slightly the deductive system proposed for $\mathrm{TL_C}$. We take the same set of axioms, and all the inference rules listed in Table 2 apart from the last four, denoted by (§). Those four are replaced by the following rules:
$$\frac{\vdash^{\mathrm{PTL}} \alpha[p_1, \ldots p_n]}{S \vdash^{\mathrm{H}} \alpha[\phi_1, \ldots, \phi_n]} \ , \qquad \frac{\vdash_0^{\mathrm{PTL}} \alpha[p_1, \ldots p_n]}{S \vdash_{(0,0)}^{\mathrm{H}} \alpha[\phi_1, \ldots, \phi_n]} \ , \qquad \frac{S \vdash^{\mathrm{D}} \phi}{S \vdash^{\mathrm{H}} \phi} \ , \qquad \frac{S \vdash_0^{\mathrm{D}} \phi}{S \vdash_{(0,0)}^{\mathrm{H}} \phi} \ ,$$
with the proviso that ϕ, ϕ_1, …, ϕ_n are HSFV with respect to S.

Acknowledgments. We wish to thank Anuchit Anuchitanukul, Nikølaj Bjorner, Arjun Kapur, Amir Pnueli and Henny Sipma for their valuable comments and suggestions.

References

1. R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. In *Proc. 10th ACM Symp. Princ. of Dist. Comp.*, pages 139–152, 1991.
2. R. Alur and T.A. Henzinger. Logics and models of real time: a survey. In J.W. de Bakker, K. Huizing, W.-P. de Roever, and G. Rozenberg, editors, *Real Time: Theory in Practice*, Lecture Notes in Computer Science 600, pages 74–106. Springer-Verlag, 1992.
3. J.P. Burgess. Axioms for tense logic I. "since" and "until". *Notre Dame journal of Formal Logic*, 23(4):367–374, October 1982.
4. L. de Alfaro and Z. Manna. Continuous verification by discrete reasoning. Technical Report STAN-CS-TR-94-1524, Stanford University, September 1994.
5. J.W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, chapter 5. D. Reidel Publishing Company, 1984.
6. T. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In W. Kuich, editor, *Proc. 19th Int. Colloq. Aut. Lang. Prog.*, volume 623 of *Lect. Notes in Comp. Sci.*, pages 545–558. Springer-Verlag, 1992.
7. T. Henzinger, Z. Manna, and A. Pnueli. Temporal proof methodologies for timed transition systems. *Inf. and Comp.*, 112(2):273–337, August 1994.
8. T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, (111):193–244, 1994.
9. Y. Kesten, Z. Manna, H. McGuire, and A. Pnueli. A decision algorithm for full propositional temporal logic. In *Computer Aided Verification, 5th International Workshop*, Lect. Notes in Comp. Sci. Springer-Verlag, 1993.
10. Y. Kesten, Z. Manna, and A. Pnueli. Temporal verification of simulation and refinement. In *Proc. of the REX Workshop "A Decade of Concurrency"*, volume 803 of *Lect. Notes in Comp. Sci.*, pages 273–346. Springer-Verlag, 1994.
11. L. Lamport. What good is temporal logic? In R.E.A. Mason, editor, *Proc. IFIP 9th World Congress*, pages 657–668. Elsevier Science Publishers (North-Holland), 1983.
12. L. Lamport. The temporal logic of actions. Technical Report 79, DEC SRC, Palo Alto, CA, December 1991.
13. L. Lamport. Verification and specification of concurrent programs. Technical report, DEC SRC, Palo Alto, CA, 1993.
14. O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Proc. Conf. Logics of Programs*, volume 193 of *Lect. Notes in Comp. Sci.*, pages 196–218. Springer-Verlag, 1985.
15. O. Maler, Z. Manna, and A. Pnueli. From timed to hybrid systems. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, *Proc. of the REX Workshop "Real-Time: Theory in Practice"*, volume 600 of *Lect. Notes in Comp. Sci.*, pages 447–484. Springer-Verlag, 1992.
16. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, New York, 1991.
17. Z. Manna and A. Pnueli. Models for reactivity. *Acta Informatica*, 30:609–678, 1993.
18. A. Pnueli. Development of hybrid systems. In *Proceedings of the Third International Symposium on Formal Techniques in Real-time and Fault-tolerant Systems*, volume 863 of *Lect. Notes in Comp. Sci.*, 1994. Extended abstract.