Formally Correct Monitors for Hybrid Automata

Goran Frehse[*], *Nikolaos Kekatos*[*], *Dejan Nickovic*[†]

Verimag Research Report n^o TR-2017-5

September 20, 2017

[*] Verimag, University of Grenoble Alpes, Grenoble, France.
[†] Austrian Institute of Technology, Vienna, Austria.

Reports are downloadable at the following address
http://www-verimag.imag.fr

Unité Mixte de Recherche 5104 CNRS - Grenoble INP - UGA

Bâtiment IMAG
Université Grenoble Alpes
700, avenue centrale
38401 Saint Martin d'Hères
France
tel : +33 4 57 42 22 42
fax : +33 4 57 42 22 22
http://www-verimag.imag.fr/

Formally Correct Monitors for Hybrid Automata

Goran Frehse[*], *Nikolaos Kekatos*[*], *Dejan Nickovic*[†]

September 20, 2017

Abstract

The paper *Pattern Templates and Monitors for Verifying Safety Properties of Hybrid Automata* aims to facilitate the integration of formal verification techniques into model-based design. It considers specifications expressed in pattern templates, which are predefined properties with placeholders for state predicates. Pattern templates are close to the natural language and can be easily understood by both expert and non-expert users. In this report, we give formal definitions for selected patterns in the formalism of hybrid automata and provide monitors which encode the properties as the reachability of an error state. By composing these monitors with the formal model under study, the property can be checked by off-the-shelf fully automated verification tools.

Keywords: Hybrid automata, first-order temporal logic, parallel composition, pattern templates

How to cite this report:

```
@techreport {TR-2017-5,
  title = {Formally Correct Monitors for Hybrid Automata},
  author = {Goran Frehse, Nikolaos Kekatos, Dejan Nickovic},
  institution = {{Verimag} Research Report},
  number = {TR-2017-5},
  year = {2017}
}
```

[*] Verimag, University of Grenoble Alpes, Grenoble, France.
[†] Austrian Institute of Technology, Vienna, Austria.

1 Hybrid Automata

In this part, we present the preliminaries and give a formal definition of a hybrid automaton and its run semantics.

Given a set $X = \{x_1, \ldots, x_n\}$ of variables, a *valuation* is a function $v : X \to \mathbb{R}$. Let $V(X)$ denote the set of valuations over X. Let $\dot{X} = \{\dot{x}_1, \ldots, \dot{x}_n\}$ and $X' = \{x'_1, \ldots, x'_n\}$. The *projection* of v to variables $Y \subseteq X$ is $v{\downarrow_Y} = \{x \to v(x) | x \in Y\}$. The *embedding* of a set $U \subseteq V(X)$ into variables $\bar{X} \supseteq X$ is the largest subset of $V(Y)$ whose projection is in U, written as $U|^{\bar{X}}$. Given that a valuation u over X and a valuation v over Y *agree*, i.e., $u{\downarrow_{X\cap\bar{X}}} = v{\downarrow_{X\cap\bar{X}}}$, we use $u \sqcup v$ to denote the valuation w defined by $w{\downarrow_X} = u$ and $w{\downarrow_{\bar{X}}} = v$. Let $\mathit{const}_X(Y) = \{(v, v') | v, v' \in V(X), v{\downarrow_Y} = v'{\downarrow_Y}\}$.

Definition 1 (Hybrid automaton) *[1, 3] A* hybrid automaton

$$H = (\mathsf{Loc}, \mathsf{Lab}, \mathsf{Edg}, X, \mathsf{Init}, \mathsf{Inv}, \mathsf{Flow}, \mathsf{Jump})$$

consists of

- *a finite set of* locations $\mathsf{Loc} = \{\ell_1, \ldots, \ell_m\}$ *which represents the discrete states,*
- *a finite set of* synchronization labels Lab*, also called its* alphabet*, which can be used to coordinate state changes between several automata,*
- *a finite set of edges* $\mathsf{Edg} \subseteq \mathsf{Loc} \times \mathsf{Lab} \times \mathsf{Loc}$*, also called* transitions*, which determines which discrete state changes are possible using which label,*
- *a finite set of* variables $X = \{x_1, \ldots, x_n\}$*, partitioned into uncontrolled variables U and controlled variables Y; a* state *of H consists of a location ℓ and a value for each of the variables, and is denoted by $s = (\ell, \mathbf{x})$;*
- *a set of states* Inv *called* invariant *or* staying condition*; it restricts for each location the values that x can possibly take and so determines how long the system can remain in the location;*
- *a set of* initial *states* $\mathsf{Init} \subseteq \mathsf{Inv}$*; every behaviour of H must start in one of the initial states;*
- *a* flow relation Flow*, where* $\mathsf{Flow}(\ell) \subseteq \mathbb{R}^{\dot{X}} \times \mathbb{R}^X$ *determines for each state $(\ell, \mathbf{x})$ the set of possible derivatives $\dot{\mathbf{x}}$, e.g., using a differential equation such as*

  $$\dot{\mathbf{x}} = f(\mathbf{x});$$

  Given a location ℓ, a trajectory *of duration $\delta \geq 0$ is a continuously differentiable function $\xi : [0, \delta] \to \mathbb{R}^X$ such that for all $t \in [0, \delta]$, $(\dot{\xi}(t), \xi(t)) \in \mathsf{Flow}(\ell)$. The trajectory* satisfies the invariant *if for all $t \in [0, \delta]$, $\xi(t) \in \mathsf{Inv}(\ell)$.*
- *a* jump relation Jump*, where* $\mathsf{Jump}(e) \subseteq \mathbb{R}^X \times \mathbb{R}^{X'}$ *defines for each transition $e \in \mathsf{Edg}$ the set of possible successors $\mathbf{x}'$ of $\mathbf{x}$; jump relations are typically described by a* guard *set $\mathcal{G} \subseteq \mathbb{R}^X$ and an assignment (or reset) $\mathbf{x}' = r(\mathbf{x})$ as*

  $$\mathsf{Jump}(e) = \{(\mathbf{x}, \mathbf{x}') \mid \mathbf{x} \in \mathcal{G} \wedge \mathbf{x}' = r(\mathbf{x})\}.$$

  A jump can be cast as urgent, which means that time cannot elapse when the state is in the guard set.

We define the behavior of a hybrid automaton with a *run*: starting from one of the initial states, the state evolves according to the differential equations whilst time passes, and according to the jump relations when taking an (instantaneous) transition. Special events, which we call *uncontrolled assignments*, model an environment that can make arbitrary changes to the uncontrolled variables.

Definition 2 (Run semantics) *An* execution *of a hybrid automaton H is a sequence*

$$(\ell_0, \mathbf{x}_0) \xrightarrow{\delta_0, \xi_0} (\ell_0, \xi_0(\delta_0)) \xrightarrow{\alpha_0} (\ell_1, \mathbf{x}_1) \xrightarrow{\delta_1, \xi_1} (\ell_1, \xi_1(\delta_1)) \xrightarrow{\alpha_{N-1}} (\ell_N, \mathbf{x}_N),$$

with $\alpha_i \in \mathsf{Lab} \cup \{\tau\}$, satisfying for $i = 0, \ldots, N-1$:

1. Trajectories: *In location ℓ_i, ξ_i is a trajectory of duration δ_i with $\xi_i(0) = \mathbf{x}_i$ and it satisfies the invariant. It does not go through urgent guard sets unless duration δ_i is 0.*

2. Jumps: *If $\alpha_i \in \mathsf{Lab}$, there is a transition $(\ell_i, \alpha_i, \ell_{i+1}) \in \mathsf{Edg}$ with jump relation $\mathsf{Jump}(e)$ such that $(\xi_i(\delta_i), \mathbf{x}_{i+1}) \in \mathsf{Jump}(e)$ and $\mathbf{x}_{i+1} \in \mathsf{Inv}(\ell_{i+1})$.*

3. Uncontrolled assignments: *If $\alpha_i = \tau$, then $\ell_i = \ell_{i+1}$ and $\xi_i(\delta_i) \downarrow_Y = \mathbf{x}_{i+1} \downarrow_Y$. This represents arbitrary assignments that the environment might perform on the uncontrolled variables $U = X \setminus Y$.*

A run *of H is an execution that starts in one of the initial states, i.e., $(\ell_0, \mathbf{x}_0) \in \mathsf{Init}$. A state $(\ell, \mathbf{x})$ is* reachable *if there exists a run with $(\ell_i, \mathbf{x}_i) = (\ell, \mathbf{x})$ for some i.*

Note that the strict alternation of trajectories and jumps in Def. 2 is of no particular importance. Two consecutive jumps can be represented by inserting a trajectory with duration zero (which always exists), and two consecutive trajectories can be represented by inserting an uncontrolled assignment jump that does not modify the variables. Having an event at the end of the run will simplify the notation in the remainder of the paper.

2 Formalizing Pattern Templates for Hybrid Automata

In this section, we introduce the pattern templates, list the requirements considered in this paper, give a compact (intuitive) definition in structured English, and a formal definition based on the runs of the hybrid automaton.

Our work builds upon the pattern templates introduced by Konrad and Cheng in [4]. While in [4], the patterns were formally defined using temporal logics (MTL), these definitions do not immediately carry over to monitoring with hybrid automata. In this respect, we select some common pattern templates and define them in a formalism that is suitable for hybrid automata.

2.1 Preliminaries

We now introduce some notation to denote in a compact manner the states on runs and the times at which these states are taken by the run. This will allow us to express properties of runs in a clear and concise manner.

Let p be a predicate over the state variables, i.e., a function $\mathbb{R}^X \to \mathbb{B}$. We write the shorthand $p(\mathbf{x})$ to denote that p is true for $\mathbf{x}$. Let the set of runs of a hybrid automaton H be $\mathrm{Runs}(H)$. In the following we consider a run $r \in R$ given by locations ℓ_i, continuous states $\mathbf{x}_i$, trajectories ξ_i, and durations δ_i. To simplify the formalization of the properties, we introduce some further notation for the timing of states on runs. For a run r the *event-times* are $t_i = \sum_{j=0}^{i} \delta_i$, so the jump number i takes place at time t_i for $i = 0, \ldots, N-1$. For notational convenience, let $t_{-1} = 0$. We introduce a total order on the time points of the run by looking at pairs (i, t), where i is an index and t is the global time. Formally, let the *event-time* be $\mathbb{T} = \mathbb{N}^0 \times \mathbb{R}^{\geq 0}$. To clarify the difference, we denote real time with t and event-time with $\tau \in \mathbb{T}$. We use the lexicographical order on event-times, formally

$$(i, t) < (i', t') \Leftrightarrow (i < i') \vee (i = i' \wedge t < t').$$

The event-time allows us to uniquely identify discrete and continuous states on the run. The *event-time domain* of a run r is the set of pairs

$$\mathrm{dom}(r) = \big\{(i, t) \bigm| 0 \leq i \leq N-1, t_{i-1} \leq t \leq t_i\big\} \cup \big\{(N, t_{N-1})\big\},$$

where the latter term captures that the last state in the run, $(\ell_N, \mathbf{x}_N)$ is taken at time t_{N-1} (total duration of the run). The *open truncated* event-time domain of a run r excluding the last T time units is the set of pairs

$$\mathrm{dom}_{-T}(r) = \big\{(i, t) \in \mathrm{dom}(r) \bigm| t < t_{N-1} - T\big\}.$$

The truncated domain will be used for properties that refer to future events that are not covered by the domain of the run. We take an optimistic view of such cases: if the property holds on the truncated domain, then it is considered to hold on the run.

For a given $\tau = (i,t) \in \mathrm{dom}(r)$, let $r(\tau) \in \mathbb{R}^X$ be the continuous state $\xi_i(t-t_i)$, and let $r_{\mathsf{Loc}}(\tau) \in \mathsf{Loc}$ be the discrete state (location) ℓ_i. This denotes the time elapsed between two event-times $\tau = (i,t), \tau' = (i',t')$ as

$$d(\tau,\tau') = t' - t.$$

Sometimes, we are interested in the first time that a predicate holds. If the predicate, say q, is true over a left-open interval, the infimum shall be used. Let

$$\mathrm{Infi}\,(r,q) = \inf_{\tau \in \mathrm{dom}(r)} q(r(\tau)).$$

If r is clear from the context, we use the shorthand

$$\tau_{q.1} = \mathrm{Infi}(r,q).$$

Similarly, we look for the first time that a predicate p holds up to and before an event-time τ',

$$\mathrm{first}\,(r,\tau',q) = \inf_{\tau \in \mathrm{dom}(r), \tau \leq \tau'} q(r(\tau)).$$

To formally denote that a predicate holds at time τ for some nonzero amount of time, we define for a run r, a predicate p, and event-time τ,

$$\mathrm{persists}\,(r,p,\tau) = \ \exists \delta > 0 : \forall \tau', \tau \leq \tau', d(\tau,\tau') \leq \delta : r(\tau').$$

2.2 Formal Definitions

We define the properties of a hybrid automaton via its runs. A hybrid automaton H satisfies a property ϕ if and only if all runs $r \in \mathrm{Runs}(H)$ satisfy ϕ. In the following, we can therefore simply define what it means for a run r to satisfy the property ϕ, which we write as $r \models \phi$. Table 1 presents a list of clarifying remarks regarding the pattern templates.

absence. *After q, it is never the case that p holds.*

$r \models \phi$ iff for all $\tau_q, \tau \in \mathrm{dom}(r)$ with $q(r(\tau_q))$ and $\tau \geq \tau_q$, holds $\neg p(r(\tau))$.

absence (timed). *When T time units are measured, after q was first satisfied, it is never the case that p holds.*

$r \models \phi$ iff for all $\tau_q, \tau \in \mathrm{dom}(r)$ with $q(r(\tau_q))$ and $d(\tau_q,\tau) \geq T$ holds $\neg p(r(\tau))$.

minimum duration. *After q, it is always the case that once p becomes satisfied, it holds for at least T time units.*

$r \models \phi$ iff either:

(i) for all $\tau_p^*, \tau_q \in \mathrm{dom}(r)$ with $q(r(\tau_q))$, $\tau_p^* \geq \tau_q$, holds $\neg p(r(\tau_p^*))$ (never q, or never p after q), or

(ii) if $\tau_q \in \mathrm{dom}(r)$ with $q(r(\tau_q))$, then for $\tau_{q.1} = \mathrm{Infi}\,(r,q)$ holds:

  (a) for all $\tau_p^*, \tau_{\bar{p}}^* \in \mathrm{dom}(r)$ with $p(r(\tau_p^*))$, $\neg p(r(\tau_{\bar{p}}^*))$, $\tau_{q.1} \leq \tau_p^* < \tau_{\bar{p}}^*$, $d(\tau_{q.1}, \tau_{\bar{p}}^*) > T$ (p not becoming false within T after $\tau_{q.1}$), and

  (b) for all $\tau_p, \tau_{\bar{p}}, \tau'_{\bar{p}} \in \mathrm{dom}(r)$ with $\tau_{q.1} \leq \tau_{\bar{p}} < \tau_p < \tau'_{\bar{p}}$, $p(r(\tau_p))$, $\neg p(r(\tau_{\bar{p}}))$ and $\neg p(r(\tau'_{\bar{p}}))$, it holds that $d(\tau_{\bar{p}}, \tau'_{\bar{p}}) > T$ (violations of p are more than T apart).

Table 1: Remarks on pattern templates.

- The properties in this paper refer to state predicates $q, p, s : \mathbb{R}^X \to \{\text{true}, \text{false}\}$. These predicates describe states, not events. When p, q, s are always true or false, the monitor automata can be simplified.
- State predicates can express timing properties by adding an extra clock to the monitor, so that the time is now a state variable that can be used in q, p and s.
- We show so-called *triggered* versions of the properties, which only take effect after a predicate q holds. A run, for which $!q$ always holds, satisfies the property.
- There are more than one equivalent definitions for the properties (e.g. switch between universal and existential quantifiers). The selection of the most suitable one has been made to reflect the natural language of the pattern templates in Table 2.
- The universal quantifier of an empty set is always true.
- It is possible to check properties both for the bounded and unbounded time horizon. For some patterns, these two cases are distinguished explicitly.
- There is both a linguistic and practical difference between *becomes true* and *holds*. The former could be seen as an edge, i.e. the signal was false earlier and then became true. The latter could describe a property that was always true.
- The monitor automata are nondeterministic because this can lead to more compact automata.

maximum duration. *After q, it is always the case that once p becomes satisfied, it holds for less than T time units.*

$r \models \phi$ iff for all $\tau_q \in \text{dom}(r)$ with $q(r(\tau_q))$ either

(i) for all $\tau \in \text{dom}(r)$ with $\tau \geq \tau_q$, $\neg p(r(\tau))$ (never q, or p never holds after q), or

(ii) for all $\tau_p, \tau_p' \in \text{dom}(r)$ with $\tau_p \geq \tau_q$, $p(r(\tau_p))$, $p(r(\tau_p'))$ one of the following holds:

  (a) $d(\tau_p, \tau_p') < T$ (τ_p' is early enough, including the $\tau_p = \tau_p'$ case), or

  (b) there is a $\tau_{\bar{p}}$ such that $\neg p(r(\tau_{\bar{p}}))$ and $\tau_p < \tau_{\bar{p}} < \tau_p'$ (p is false in between).

bounded recurrence. *After q, it is always the case that p holds at least every T time units.*

For the unbounded case, $r \models \phi$ iff for all $\tau_q \in \text{dom}(r)$ with $q(r(\tau_q))$ both following criteria hold:

(i) for all $\tau_p \in \text{dom}(r)$ with $p(r(\tau_p))$ and $\tau_p \geq \tau_q$ there is a $\tau_p' \in \text{dom}(r)$ such that $\tau_p < \tau_p'$, $d(\tau_p', \tau_p) \leq T$ and $p(r(\tau_p'))$ (τ_p's with distance less than T).

(ii) there is a $\tau_p \in \text{dom}(r)$ with $\tau_p \geq \tau_q$, $p(r(\tau_p))$ such that $d(\tau_q, \tau_p) \leq T$. (distance between τ_q and first τ_p is less than T).

For a bounded time horizon, $r \models \phi$ iff for all $\tau_q \in \text{dom}_{-T}(r)$ with $q(r(\tau_q))$ both following criteria hold:

(i) for all $\tau_p \in \text{dom}_{-T}(r)$ with $p(r(\tau_p))$ and $\tau_p \geq \tau_q$ there is a $\tau_p' \in \text{dom}(r)$ such that $\tau_p < \tau_p'$, $d(\tau_p', \tau_p) < T$ and $p(r(\tau_p'))$.

(ii) there is a $\tau_p \in \text{dom}(r)$ with $\tau_p \geq \tau_q$, $p(r(\tau_p))$ such that $d(\tau_q, \tau_p) \leq T$.

bounded response (persisting). *After q, it is always the case that if p holds, then s persists (holds for nonzero time) after at most T time units.*

For an unbounded time horizon, $r \models \phi$ iff for all $\tau_q \in \mathrm{dom}(r)$ with $q(r(\tau_q))$ one of the following holds:

(i) for all $\tau \in \mathrm{dom}(r)$ with $\tau \geq \tau_q$, $\neg p(r(\tau))$ (never q, or p never holds after q), or

(ii) for all $\tau_p \in \mathrm{dom}(r)$ with $\tau_p \geq \tau_q$ and $p(r(\tau_p))$, there is a $\tau_s \in \mathrm{dom}(r)$ such that $\tau_p \leq \tau_s$, $d(\tau_s, \tau_p) \leq T$ and $\mathrm{persists}\,(r, \tau_s, s)$.

For a bounded time horizon, $r \models \phi$ iff one of the following holds:

(i) for all $\tau_q, \tau \in \mathrm{dom}(r)$ with $q(r(\tau_q))$, $\tau \geq \tau_q$, holds $\neg p(r(\tau))$ (never q, or p never holds after q), or

(ii) for all $\tau_q, \tau_p \in \mathrm{dom}_{-T}(r)$ with $\tau_p \geq \tau_q$, $q(r(\tau_q))$ and $p(r(\tau_p))$, there is a $\tau_s \in \mathrm{dom}(r)$ such that $\tau_p \leq \tau_s$, $d(\tau_p, \tau_s) \leq T$ and $\mathrm{persists}\,(r, \tau_s, s)$.

Remark 1 *The reason why we require $\tau \in \mathrm{dom}_{-T}(r)$ in the bounded time horizon (with the restricted domain being right-open) is the following: We assume an optimistic interpretation of bounded runs, in the sense that if there is a continuation of the run for which the system satisfies the property, then the bounded run satisfies the property. If the restricted domain was right-closed, then a run ending with $\neg s$ could violate the property, but have a continuation that (in zero time) sets s to true, which then should satisfy the property.*

Remark 2 *We require s to hold for nonzero time, formally with the use of* $\mathrm{persists}\,(\cdot)$*, because the monitor automaton may give a false alarm otherwise.*

bounded invariance. *After q, it is always the case that if p holds, then s holds for at least T time units.*
$r \models \phi$ iff one of the following holds:

(i) for all $\tau_q \in \mathrm{dom}(r)$ with $q(r(\tau_q))$, there is no τ_p with $\tau_p \geq \tau_q$ such that $p(r(\tau_p))$ (never q, or p never holds after q), or

(ii) for all $\tau_p \in \mathrm{dom}(r)$ with $\tau_p \geq \tau_q.1$, $p(r(\tau_p))$, and for all $\tau \in \mathrm{dom}(r)$ such that $\tau_p \leq \tau$, $d(\tau_p, \tau) < T$, the predicate $s(r(\tau))$ is true.

Remark 3 *Note that in the case that predicates $s = p$, then p has to hold forever (by recursion).*

3 Monitor Automata for Reachability

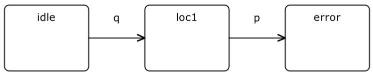
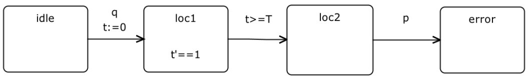
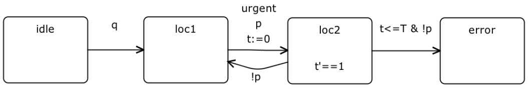
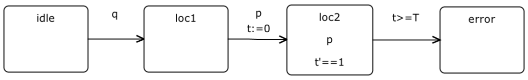
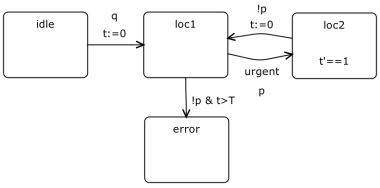
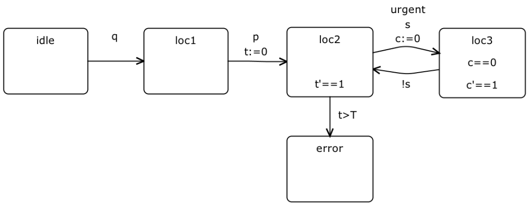
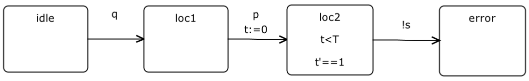
In this section, we define monitor automata that, composed with the system under test, encode the requirements as reachability properties as follows. Consider a system under test H and a monitor automaton M. The goal is that H satisfies a property ϕ if and only if the location *error* is unreachable in the parallel composition $H||M$. We prove correctness of M by showing that every violating run of H has a corresponding run in $H||M$ that reaches the error location, and vice versa. The monitor automata are shown in Table 2.

3.1 Parallel Composition

We now give a formal definition of the standard way to couple two hybrid automata. We will use this operation to connect the system under test with its monitor. Intuitively, both automata must agree on every change of a variable. The operator is similar to the composition operator in [2].

The jump relations of synchronized transitions result from the conjunction of the participating transitions. Independent transitions, i.e., those that do not synchronize, are allowed to change variables arbitrarily and the variables over which their jump relation is not defined are set to remain constant.

Table 2: Pattern templates and translation to monitor automata.

Pattern name	**Language Template**	**Monitor Automaton**
absence	After q, it is never the case that p holds.	
absence (timed)	When T time units are measured, after q was first satisfied, it is never the case that p holds.	 idle q t:=0 loc1 t'==1 t>=T loc2 p error
minimum duration	After q, it is always the case that once p becomes satisfied, it holds for at least T time units.	 idle q loc1 urgent p t:=0 !p loc2 t'==1 t<=T & !p error
maximum duration	After q, it is always the case that once p becomes satisfied, it holds for less than T time units.	 idle q loc1 p t:=0 loc2 p t'==1 t>=T error
bounded recurrence	After q, it is always the case that p holds (for nonzero time) at least every T time units.	 idle q t:=0 loc1 !p t:=0 loc2 t'==1 urgent p !p & t>T error
bounded response (persisting)	After q, it is always the case that if p holds, then s persists (holds for nonzero time) after at most T time units.	 idle q loc1 p t:=0 loc2 t'==1 urgent s c:=0 !s loc3 c==0 c'==1 t>T error
bounded invariance	After q, it is always the case that if p holds, then s holds for at least T time units.	 idle q loc1 p t:=0 loc2 t<T t'==1 !s error

Definition 3 (Composition of HA) *The* parallel composition *of hybrid automata H_1 and H_2 is the hybrid automaton $H = H_1||H_2$*

- $Loc = Loc_1 \times Loc_2$,
- $Lab = Lab_1 \cup Lab_2$,
- $Edg = \{((\ell_1, \ell_2), \alpha, (\ell'_1, \ell'_2)) \mid (\alpha \in Lab_1 \Rightarrow (\ell_1, \alpha, \ell'_1)) \wedge (\alpha \in Lab_2 \Rightarrow (\ell_2, \alpha, \ell'_2))\}$,
- $X = X_1 = X_2$ *(by assumption)*, $Y = Y_1 \cup Y_2$, $U = (U_1 \cup U_2) \setminus Y$,
- $Jump((\ell_1, \ell_2), a, (\ell'_1, \ell'_2))$ *with* $\mu = \{(v, v') \in \mu_i\}$ *iff for* $i = 1, 2$,
  - $a \in Lab_i$ *and* $(\ell_i, a_i, \mu_i, \ell'_i) \in Edg_i$, *or*
  - $a \notin Lab_i$, $\ell'_i = \ell_i$, *and* $\mu_i = const_{X_i}(Z_i)$, *where* $Z_1 = Y_1 \backslash Y_2$ *and* $Z_2 = Y_2 \backslash Y_1$;
- $Flow(\ell_1, \ell_2) = Flow_1(\ell_1) \cap Flow_2(\ell_2)$;
- $Inv(\ell_1, \ell_2) = Inv_1(\ell_1) \cap Inv_2(\ell_2)$;
- $Init(\ell_1, \ell_2) = Init_1(\ell_1) \cap Init_2(\ell_2)$.

Without loss of generality we can assume that H and M have the same variables. If M has a variable not in H, e.g., a clock variable for measuring the time between events, we can add it to H without restricting it in the invariants, guards, or flows. Note that all transitions in M have the label τ, so they do not synchronize with any transitions in H.

A run $r_{H||M}$ in $H||M$ is given by locations $\ell_i = (\ell_i^H, \ell_i^M)$, continuous states $\mathbf{x}_i$, trajectories ξ_i, durations δ_i, and labels α_i. Let r_H be the projection of the run onto H, obtained by replacing ℓ_i with ℓ_i^H, and let r_M be the projection of the run onto M, obtained by replacing ℓ_i with ℓ_i^M and α_i with τ. Then by definition, we have that for any run $r_{H||M}$ in $\text{Runs}(H||M)$, $r_H \in \text{Runs}(H)$ and $r_M \in \text{Runs}(M)$.

3.2 Operations on Runs

We use the following shorthand notation to improve the readability of the proofs. As shorthand, we will define a run by the sequence $(\ell_i, \mathbf{x}_i, \delta_i, \xi_i, \alpha_i)_{i=0,\ldots,N-1}$. Given a run r and an event-time $\tau^* = (k^*, t^*) \in \text{dom}(r)$, the run can be split into the *prefix* up to τ^*, and the *postfix* after τ^*. The prefix is extended with a silent transition, which by definition can be injected anywhere:

$$\text{prefix}\,(r, (k^*, t^*)) = (\ell_i, \mathbf{x}_i, \delta_i, \xi_i, \alpha_i)_{i=0,\ldots,k^*-1}; (\ell_{k^*}, \mathbf{x}_{k^*}, t^* - t_{k^*-1}, \xi_{k^*}, \tau). \tag{1}$$

$$\text{postfix}\,(r, (k^*, t^*)) = \left(\ell_{k^*}, r(k^*, t^*), \delta_{k^*} - t_{k^*-1}, \xi^*, \alpha_{k^*}\right); (\ell_i, \mathbf{x}_i, \delta_i, \xi_i, \alpha_i)_{i=k^*+1,\ldots,N-1}, \tag{2}$$

where $r(k^*, t^*) = \xi_{k^*}(t^* - t_{k^*-1})$, and $\xi^*(t) = \xi_{k^*}(t - t_{k^*-1})$ is the trajectory $\xi_{k^*}(t)$ shifted backwards in time by t_{k^*-1}. Similarly, the *infix* between event-times $\tau_a = (k_a, t_a) \in \text{dom}(r)$, $\tau_b = (k_b, t_b) \in \text{dom}(r)$, with $\tau_a \leq \tau_b$, is

$$\text{infix}\,(r, (k_a, t_a), (k_b, t_b)) = \text{prefix}\,(\text{postfix}\,(r, (k_a, t_a)), (k_b - k_a, t_b - t_a))\,. \tag{3}$$

It is straightforward that the concatenation

$$\text{prefix}\,(r, \tau)\ ;\ \text{postfix}\,(r, \tau)$$

is a run of H. Similarly, the concatenation

$$\text{prefix}\,(r, \tau_a)\ ;\ \text{infix}\,(r, \tau_a, \tau_b)\ ;\ \text{postfix}\,(r, \tau_b)$$

is a run of H. With a slight abuse of notation, we write $r \times \ell^*$ to denote the run $(\ell_i \times \ell^*, \mathbf{x}_i, \delta_i, \xi_i, \alpha_i)_{i=0,\ldots,N-1}$. This is not necessarily a run of $H||M$, but it can be one, such as under the following condition.

Lemma 1 *Let $r = (\ell_i, \mathbf{x}_i, \delta_i, \xi_i, \alpha_i)$ be a run of H. If a location ℓ_M in M has (i) no invariant constraints and (ii) no urgent outgoing transitions, then $r \times \ell_M$ is a run of $H||M$.*

Lemma 2 *Let $r = (\ell_i, \mathbf{x}_i, \delta_i, \xi_i, \alpha_i)$ be a run of H. If a location ℓ_M in M has (i) no invariant constraints and (ii) one urgent outgoing transition with guard condition p, leading to location ℓ'_M that has (iii) no invariant constraints and (iv) no urgent outgoing transitions then*

$$\text{prefix}\,(r, \tau_{p.1}) \times \ell_M \ ; \ \text{postfix}\,(r, \tau_{p.1}) \times \ell'_M$$

is a run of $H||M$, where $\tau_{p.1} = \text{Infi}\,(r, p)$ is the smallest event time where p holds.

We call a monitor M non-blocking if for any run r_H of H, there is a corresponding run $r_{H||M}$ of $H||M$ such that r_H is the projection of $r_{H||M}$ onto H. Simply put, there is no deadlock that caused a run to be terminated.

3.3 Correctness Proofs

A monitor automaton is correct if its error location is reachable exactly when the system H violates the property. Formally, let h be a run of H that violates a given property ϕ. Then we first show (a) that there exists a run r of $H||M$ that reaches the error location. Second, we show (b) that for any run r of $H||M$ that reaches the error location, the run projected onto H violates the property.

3.3.1 Sufficient Conditions.

absence. Since $r \not\models \phi$, there exist $\tau_q, \tau_p \in \text{dom}(r)$ with $q(r(\tau_q))$, $\tau_p \geq \tau_q$, and $p(r(\tau_p))$.

With Lemma 1 and the definition of a jump,

$$\text{prefix}\,(h, \tau_q) \times \text{idle} \ ; \ \text{infix}\,(h, \tau_q, \tau_p) \times \text{loc1} \ ; \text{postfix}\,(h, \tau_p) \times \text{error}$$

is a run of $H||M$.

absence (timed). Since $r \not\models \phi$, there exist $\tau_q, \tau_p \in \text{dom}(r)$ with $q(r(\tau_q))$, $d(\tau_q, \tau_p) \geq T$, and $p(r(\tau_p))$.

M can remain in idle location during $\text{prefix}\,(h, \tau_q)$, then transition to loc1 and remain there during $\text{infix}\,(h, \tau_q, \tau_p)$. M can then transition to loc2 with Lemma 1. In loc2, M can take the transition to error, as p holds.

minimum duration. $r \not\models \phi$, so there is $\tau_p \geq \tau_q$ with $p(r(\tau_p^*))$ and $q(r(\tau_q))$, and one of the following is true:

(a) there are $\tau_p, \tau'_{\bar{p}}$ with $\tau_{q.1} \leq \tau_p < \tau'_{\bar{p}}$, $p(r(\tau_p))$, $\neg p(r(\tau'_{\bar{p}}))$, and $d(\tau_{q.1}, \tau_{\bar{p}}) \leq T$, or

(b) there are $\tau_p, \tau_{\bar{p}}, \tau'_{\bar{p}} \in \text{dom}(r)$ with $\tau_{q,1} \leq \tau_{\bar{p}} < \tau_p < \tau'_{\bar{p}}$, $p(r(\tau_p))$, $\neg p(r(\tau_{\bar{p}}))$, $\neg p(r(\tau'_{\bar{p}}))$, and $d(\tau_{\bar{p}}, \tau'_{\bar{p}}) \leq T$.

In case (a), let $\tau_{p.1} = \text{first}\,(r, \tau_{q.1}, p)$, so $\tau_{p.1} \leq \tau_p$. M can remain in idle location during $\text{prefix}\,(h, \tau_{q.1})$, then transition to loc1 and with Lemma 2 remain there during $\text{infix}\,(h, \tau_q, \tau_{p.1})$. M can then transition to loc2, setting t to zero with Lemma 1. M can remain in loc2 during $\text{infix}\,\left(h, \tau_p, \tau'_{\bar{p}}\right)$. Since $d(\tau_{q,1}, \tau'_{\bar{p}}) \leq T$, we have $t \leq T$. M can then transition to error.

In case (b), we first show that M can be at loc1 at $\tau_{\bar{p}}$. After τ_q, M can go to loc2 as soon as p is satisfied, and move back to loc1 as soon as p is violated. We can therefore assume that M can be loc1 at $\tau_{\bar{p}}$. We match the remainder of the run in analogy to case (a), replacing $\tau_{q.1}$ by $\tau_{\bar{p}}$.

In the following, we only highlight the differences with the aforementioned proofs (ignoring what happens before τ_q and τ_p).

maximum duration. $r \not\models \phi$ implies that

(i) there is $\tau_p \geq \tau_q$ with $r(\tau_p)$ satisfying p and $r(\tau_q)$ satisfying q, and

(ii)(a) there is τ'_p with $p(r(\tau'_p))$ and $d(\tau_p, \tau'_p) \geq T$, and

(ii)(b) there is no $\tau_{\bar{p}}$ such that $\neg(p(r(\tau_{\bar{p}}))$ and $\tau_p < \tau_{\bar{p}} < \tau'_p$.

At τ_p, M can be either in $\mathrm{loc1}$ or $\mathrm{loc2}$. In $\mathrm{loc1}$, M can take the transition to $\mathrm{loc2}$, as p holds. Once in $\mathrm{loc2}$, M can wait there for T time units, since with (ii)(a) and (ii)(b), p still holds. M can then transition to error.

bounded recurrence. For the unbounded case (i), there is $\tau_p \in \mathrm{dom}(r)$ with $p(r(\tau_p))$ and $\tau_p \geq \tau_q$, such that there is no $\tau'_p > \tau_p$, with $d(\tau'_p, \tau_p) \leq T$ and $p(r(\tau'_p))$ (τ_p's with distance less than T).

If M is in $\mathrm{loc1}$ at τ_p, it takes the transition from $\mathrm{loc1}$ to $\mathrm{loc2}$. If M is in $\mathrm{loc2}$, there are two subcases:

(a) p does not hold within T time units after τ_p, in which case M can go to $\mathrm{loc1}$, wait for more than T time and then go to error.

(b) p holds after τ_p, which means that it holds at a time τ'_p with $d(\tau_p, \tau'_p) > T$. Let $\delta = d(\tau_p, \tau'_p) - T$. Then M can wait for $\delta/3$ time in loc2, after which p is false. Then M can go to loc1, wait for $T + \delta/3$ time, and since only $T + 2\delta/3$ time has passed since τ_p, p is still false. Since $t > T$, M can go to error.

bounded response (persisting). For the unbounded time horizon, $r \not\models \phi$ implies that

(i) there is $\tau_p \geq \tau_q$ with $r(\tau_p)$ satisfying p and $r(\tau_q)$ satisfying q, and

(ii) there is no $\tau_s \geq \tau_p$ such that $r(\tau_s)$ satisfies s, $d(\tau_p, \tau_s) \leq T$ and $\mathrm{persists}\,(r, \tau_s, s)$.

At τ_p, M can be either in $\mathrm{loc1}$, $\mathrm{loc2}$, or $\mathrm{loc3}$. In $\mathrm{loc1}$, M can transition immediately to $\mathrm{loc2}$ (because p is true). In $\mathrm{loc2}$, there are two options. If s is false ($\nexists \tau_s : d(\tau_p, \tau_s) \leq T$), M can stay there for more than T time units. M can then transition to error. If s is not always false, there is a τ_s such that $\neg\mathrm{persists}\,(r, \tau_s, s)$. At τ_s, M instantaneously moves to loc3 and then back to loc2 when s does not hold. From $\mathrm{loc3}$, if $\neg s$, M can transition to loc2. If s and $\neg\mathrm{persists}\,(r, \tau_s, s)$, M can transition to loc2, since $\neg\mathrm{persists}\,(r, \tau_s, s) : \exists \tau'_s > \tau_s$ with $d(\tau_s, \tau'_s) = 0$ and $\neg s(\tau'_s)$.

For the bounded time horizon, $r \not\models \phi$ implies that

(i) there is $\tau_p \geq \tau_q$ with $r(\tau_p)$ satisfying p and $r(\tau_q)$ satisfying q, and

(ii) $\tau_q, \tau_p \in \mathrm{dom}_{-T}(r)$ and there is no $\tau_s \in \mathrm{dom}(r)$ such that $\tau_p \leq \tau_s$, $d(\tau_p, \tau_s) \leq T$ and $\mathrm{persists}\,(r, \tau_s, s)$.

The proof is analogous to the unbounded case.

bounded invariance. $r \not\models \phi$ implies that

(i) there is $\tau_p \geq \tau_q$ with $r(\tau_p)$ satisfying p and $r(\tau_q)$ satisfying q, and

(ii) there is a τ with $\tau \geq \tau_p$ such that $d(\tau_p, \tau) < T$ and $s(r(\tau))$ is false.

At τ_p, M can be either in $\mathrm{loc1}$ or $\mathrm{loc2}$. In $\mathrm{loc1}$, M can transition immediately to $\mathrm{loc2}$ (because p is true). Once in $\mathrm{loc2}$, M can wait there for $t = d(\tau_p, \tau)$. Since $\mathrm{d}(\tau_p, \tau) < T$, M can remain in $\mathrm{loc2}$ until τ. Since $\neg s$ at τ ($\neg s(r(\tau)$ holds), M can then transition to error.

3.3.2 Necessary Conditions.

For the necessary condition, we need to show that a run r in $H||M$ that ends in location error implies a run in H that violates the property. Let r_H be the projection of the run onto H (removing the locations and clocks of M). It is straightforward that r_H is a run of H. In the following, we show that $r_H \not\models \phi$. Note that r starts in location idle. Note also that any event-times of r are also event-times of r_H.

absence. To get from idle to error, M had to take first a transition with guard q and then a transition with guard p. Consequently, there exist τ_q and τ_p with $\tau_q \leq \tau_p$, $q(r_H(\tau_q))$ and $p(r_H(\tau_p))$. τ_q and τ_p are witnesses that violate ϕ.

absence(timed). To get from idle to error, M had to take first a transition with guard q, wait for T time units, and then take a transition with guard p. Consequently, there exist τ_q and τ_p with $d(\tau_q, \tau_p) \geq T$, $q(r_H(\tau_q))$ and $p(r_H(\tau_p))$. τ_q and τ_p are witnesses that violate ϕ.

minimum duration. Similarly to the above proof of the absence pattern, we can stipulate the existence of τ_q, τ_p and $\tau'_{\bar{p}}$ with $\tau_q \leq \tau_p \leq \tau'_{\bar{p}}$, $q(r_H(\tau_q))$, $p(r_H(\tau_p))$ and $\neg p(r_H(\tau'_{\bar{p}}))$. τ_q and τ_p are witnesses that violate case (i).

For case (ii), let $\tau_{q.1} = \text{first}\,(r, 0, q)$, so that $\tau_{q.1} \leq \tau_q$. Without loss of generality, we can assume that τ_p is the last event-time on r where M entered loc2, so $t = d(\tau_p, \tau'_{\bar{p}})$. Because of the transition guard from $\mathrm{loc2}$ to error, $d(\tau_p, \tau'_{\bar{p}}) \leq t \leq T$. There are two subcases:

(a) If there is no $\tau_{\bar{p}}$ with $\tau_{q.1} \leq \tau_{\bar{p}} \leq \tau_p$ and $\neg p(r_H(\tau_{\bar{p}}))$, we can conclude that $\tau_{q.1} = \tau_{p.1}$, where $\tau_{p.1} = \text{first}\,(r, \tau_{q.1}, p)$. In this case, the run in M goes from idle to loc1 to loc2, so $\tau_{q.1} = \tau_{p.1} = \tau_p$. Consequently, $d(\tau_{q.1}, \tau'_{\bar{p}}) = d(\tau_p, \tau'_{\bar{p}}) \leq T$, which violates case (a).

(b) Otherwise, we have $\tau_{q.1} \leq \tau_{\bar{p}} \leq \tau_{p.1} \leq \tau'_{\bar{p}}$. We will show that there is a $\tau^* \leq \tau_p$, with $d(\tau^*, \tau_p) = 0$ and where $r(\tau^*)$ violates p. Then $d(\tau^*, \tau'_{\bar{p}}) \leq T$, which violates case (b). We now show the existence of τ^*, by first identifying some $\tau' \leq \tau_p$ such that M is in loc1 for all $\tau' \leq \tau \leq \tau_p$, and for which $r(\tau')$ violates p. Consider that we can assume that $\mathrm{loc1}$ was entered either from idle with p being violated (otherwise case (a) applies), or from $\mathrm{loc2}$, which also means p is violated. Since the transition from $\mathrm{loc1}$ to $\mathrm{loc2}$ is urgent, p can not hold for any τ with $\tau' \leq \tau < \tau_p$ where $d(\tau, \tau_p) > 0$ (no time can elapse while p is true). So there exists a τ^* with $\tau' \leq \tau^* \leq \tau_p$ with $d(\tau^*, \tau_p) = 0$.

maximum duration. Let τ_p be the last event-time on r where M entered loc2. As the $\mathrm{loc2}$ has invariant p and the transition guard from loc2 to error has the constraint $t \geq T$, we know that at least T time units have elapsed in $\mathrm{loc2}$. That means that there exist τ_p and τ'_p so that $d(\tau_p, \tau'_p) \geq T$ without any $\tau_{\bar{p}}$ in between them. Therefore, τ_q, τ_p and the absence of $\neg p$ witnesses the violation of ϕ.

bounded recurrence. Since time can only elapse in $\mathrm{loc2}$ while $\neg p$ and t is reset on all incoming transitions, we know that $\neg p$ holds for more than T time units, which violates the property.

bounded response (persisting). Similarly to the above proof of the absence pattern, we can stipulate the existence of τ_q and τ_p. Let τ_p be the last event-time on r where M entered $\mathrm{loc2}$. Cycles between $\mathrm{loc2}$ and $\mathrm{loc3}$ take zero time: because of the urgent transition from $\mathrm{loc2}$ to $\mathrm{loc3}$, s was false during this time, with the possible exception of switching to true and back to false in zero time (which doesn't satisfy the definition of "persists"). Because the transition guard from $\mathrm{loc2}$ to error has the constraint $t > T$, we know that more than T time units have elapsed in $\mathrm{loc2}$. Therefore, τ_q, τ_p and the absence of s witness the violation of ϕ.

bounded invariance. Assuming that M is in the error location, due to the guard conditions of the incoming transitions, we know that at some point τ on the run, s did not hold. In $\mathrm{loc2}$, we know from the incoming guard conditions and its invariance, that p held at some point τ_p with $\tau_p \leq \tau$ and $d(\tau_p, \tau) < T$. Therefore, τ_q, τ_p, and τ witness the violation of ϕ.

Acknowledgments

The authors gratefully acknowledge financial support by the European Commission project UnCoVerCPS under grant number 643921.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 1995. 1

[2] R. Alur, T. A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *ITSE*, 1996. 3.1

[3] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *JCSS*, 1998. 1

[4] S. Konrad and B. Cheng. Real-time specification patterns. In *ICSE Conference*, 2005. 2