Hybrid Systems Verification by Location Elimination

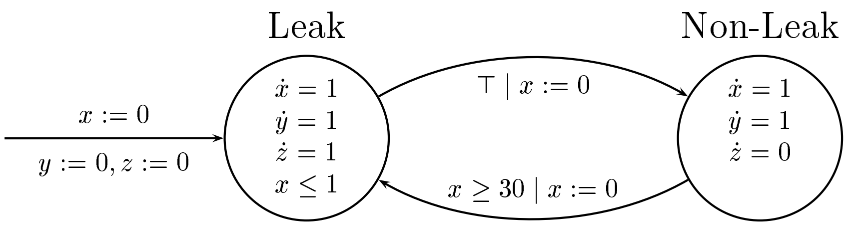
Andreas Nonnengart

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI GmbH)
Im Stadtwald, Geb. 36, 66123 Saarbrücken, Germany
http://www.dfki.de/~nonnenga/
Andreas.Nonnengart@dfki.de

Abstract. In this paper we propose a verification method for hybrid systems that is based on a successive elimination of the various system locations involved. Briefly, with each such elimination we compute a weakest precondition (strongest postcondition) on the predecessor (successor) locations such that the property to be proved cannot be violated. Experiments show that this approach is particularly interesting in cases where a standard reachability analysis would require to travel often through some of the given system locations.

1 Introduction

Hybrid Systems are real-time systems that are embedded in analog environments. They contain discrete and continuous components and interact with the physical world through sensors and actuators. A common model for hybrid systems can be found in hybrid automata. These are finite graphs whose nodes correspond to global states as illustrated in the famous "Leaking Gas Burner" example [ACH+95]:



Nodes "Leak" and "Non-Leak" represent discrete locations, whereas x, y, and z are data variables. Each location may contain a location invariant ($x \leq 1$ in the example) and the continuous activity which describes how the values of the data variables change in time. In the above example the value of x and y increase by 1 per time unit (say, second), i.e., the first derivative of the function describing the behavior of x and y over time is the constant 1. z also increases by one per second in location "Leak", however, it remains unchanged ($\dot{z} = 0$) in location "Non-Leak". Edges are annotated with guards and discrete actions. Guards form a constraint on the data variables to hold if a transition via the corresponding edge is to be performed. The discrete action specifies how the data variables are to be changed after taking the transition. In the above example the guard of the edge from "Leak" to "Non-Leak" is logical truth ($\top$), i.e., no special condition has to be fulfilled, whereas the guard of the edge from "Non-Leak" to "Leak" is $x \geq 30$, i.e., this edge may only be taken in case the value of the data variable x is at least 30. The discrete action for both edges is to reset x to 0.

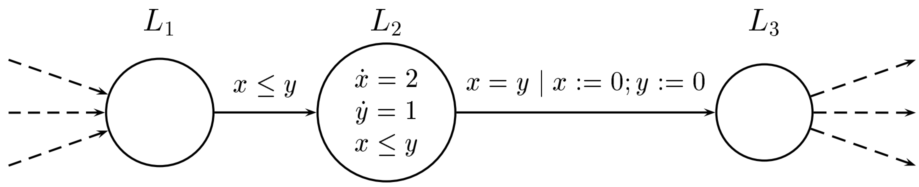
A *computation* of such an automaton is a sequence of state changes (steps). Within each step the system state evolves continuously according to a dynamical law until a transition from one node to another one occurs.

Since hybrid systems typically operate in safety-critical situations, the development of rigorous analysis techniques is of high importance. In the last ten years several proposals for a verification methodology for hybrid systems arose [ACD90, ACH$^+$95, ACHH93, AD94, AH92, AHH96, AHS96, ANKS95, GNRR93, Hen96, HNSY92]. Most of them are based on a so-called (forward or backward) reachability analysis. Intuitively, a forward reachability analysis (for safety properties) performs the following operation: starting from the initial situation (state), all possible (immediate) time and edge successor states are computed. Then the resulting set is reentered as an input to compute further time and edge successors, and so on. This will go on until no further new states can be derived (reached). Provided this procedure at all terminates, it ultimately comes up with the set of all reachable states (reachable from the initial state) which may be used to check the property to be proved.

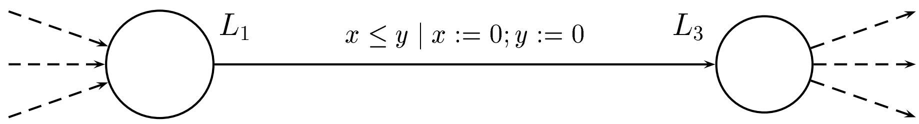
Backward reachability, on the other hand, starts from a description of the states that do *not* fulfill the property to be proved and tries to compute all possible predecessor states, i.e., all the states from which one of the unsafe states could possibly be reached. Again, upon termination, it ends up with a set of states all of whose elements may lead to an unsafe situation, and what remains to be done is to check whether the initial state is contained in this set or not.

At the first glance, at least for forward reachability, and upon termination, it can hardly be seen that anything else could behave better. After all, forward reachability computes exactly the reachable states (and nothing else) and we need to know about all the reachable states in order to perform our verification task. Indeed, forward reachability does not compute any redundant information. However, it may perform redundant computations. For instance, if the property to be proven requires several passes through certain locations, the actual effect is usually very similar to earlier (or later) passes. Only the values of the variables involved might vary a bit, although in some more or less regular way.

The purpose of the approach proposed in this paper is to show how to gain such a knowledge and how to take advantage of it, i.e., to compute the behavior of locations once and for all and to forget about these locations later on. This can result in certain extra properties to be proved for the other locations (weakest preconditions or strongest postconditions) that take over the responsibility of the location just eliminated. The following example illustrates this:



Let us assume that we want to prove that $x+y \leq 10$ is an invariant of the system (from which we can only see a small portion in this illustration). It might very well be that any reachability analysis will have to go through location L_2 several times and therefore we want to eliminate this very location by computing a weakest precondition on location L_1 that guarantees that L_2 could impossibly violate the desired property. The approach to be presented in this paper will come up with the following result: the invariant indeed holds for the whole system if and only if the invariant holds for the "simplified" system



provided we can guarantee that $x \leq y \rightarrow 2y \leq x + 5$ whenever we are within location L_1. In what follows we describe formally how such a condition can be computed and how this method can be used as a verification tool for hybrid systems.

2 Preliminaries

Given a fixed variable set X we define the set CT of *Constraint Terms* (over the variable set X) as the smallest set containing the reals and the set X and that is closed under addition, subtraction and multiplication with reals. The set CF of *Constraint Formulas* (over the variable set X) is defined as the smallest set containing $\top$, $\bot$ (logical truth and falsity respectively), $t_1 > t_2$, $t_1 \geq t_2$, $t_1 < t_2$, $t_1 \leq t_2$, and $t_1 = t_2$ (where t_1 and t_2 are constraint terms), and that is closed under logical conjunction.

A hybrid system is a tuple of the form $\mathcal{H} = (X, \mathcal{L}, \mathcal{E}, \mathit{dif}, \mathit{inv}, \mathit{guard}, \mathit{act})$ where X is a finite set of real-valued *data variables*, $\mathcal{L}$ is a finite set of *locations*, i.e., nodes of a graph, $\mathcal{E} \subseteq \mathcal{L} \times \mathcal{L}$ is a finite (multi)set of *transitions*, i.e., edges of the graph with nodes from $\mathcal{L}$, $\mathit{dif}\colon \mathcal{L} \times X \mapsto \mathrm{CT}$ is a mapping that associates with each location and each data variable a constraint term (over X), $\mathit{inv}\colon \mathcal{L} \mapsto \mathrm{CF}$ is a mapping that associates with each location a constraint formula, representing the location invariant, $\mathit{guard}\colon \mathcal{E} \mapsto \mathrm{CF}$ is a mapping that associates with each edge a constraint formula, representing the enabling condition for this transition, and, $\mathit{act}\colon \mathcal{E} \times X \mapsto \mathrm{CT}$ is a mapping that associates with each edge and each data variable a constraint term, representing the value of the variable after traveling along the edge.

As usual, we define a *state* of a hybrid system as a pair (L, ϕ) where $L \in \mathcal{L}$ is a location and $\phi\colon X \mapsto \mathbb{R}$ is a valuation of the data variables. ϕ naturally extends to (constraint) terms and (constraint) formulas. A state (L, ϕ) is called *admissible* if $\phi(inv(L))$ holds. Given two admissible states $s = (L, \phi)$ and $s' = (L', \phi')$ we say that s' is *transition-reachable* from s – denoted by $s \overset{\text{tr}}{\mapsto} s'$ – if there exists a transition $t = (L, L') \in \mathcal{E}$ with source L and target L', and both $\phi(guard(t))$ and $\phi'(x) = \phi(act(t, x))$ for each $x \in X$. We call s' *timely-reachable* from s with delay δ – denoted by $s \overset{\delta}{\mapsto} s'$ – where δ is a non-negative real number, if $L = L'$ and for each $x \in X$ there exists a differentiable function $f_x\colon [0, \delta] \mapsto \mathbb{R}$, with the first derivative $\dot{f}_x\colon (0, \delta) \mapsto \mathbb{R}$, such that (1) $f_x(0) = \phi(x)$ and $f_x(\delta) = \phi'(x)$ and (2) for all $\epsilon \in \mathbb{R}$ with $0 < \epsilon < \delta$: both $inv(L)[x_1/f_{x_1}(\epsilon), \ldots, x_n/f_{x_n}(\epsilon)]$ and $\dot{f}_x(\epsilon) = dif(L, x)[x_1/f_{x_1}(\epsilon), \ldots, x_n/f_{x_n}(\epsilon)]$ are true. s' is *timely-reachable* from s – denoted by $s \overset{*}{\mapsto} s'$ – if there exists a non-negative $\delta \in \mathbb{R}$ such that $s \overset{\delta}{\mapsto} s'$. s' is said to be *reachable* from s if $(s, s') \in (\overset{*}{\mapsto} \cup \overset{\text{tr}}{\mapsto})^*$. A *run* ρ of $\mathcal{H}$ with initial state $\sigma_0 = (L_0, \phi_0)$ is a maximal sequence of states represented as

$$\rho = \sigma_0 \mapsto_{f_0}^{t_0} \sigma_1 \mapsto_{f_1}^{t_1} \sigma_2 \mapsto_{f_2}^{t_2} \sigma_3 \mapsto_{f_3}^{t_3} \cdots$$

where $t_i \in \mathbb{R}^{\geq 0}$ and $f_i\colon [0, t_i] \mapsto (X \mapsto \mathbb{R})$, such that (i) $f_i(0) = \phi_i$, (ii) $inv(L_i)[X/f_i(t)(X)]$ holds for all $0 \leq t \leq t_i$, (iii) $(L_i, f_i(t_i)) \overset{\text{tr}}{\mapsto} \sigma_{i+1}$ and (iv) for all $0 \leq t' \leq t' + \delta \leq t_i : (L_i, f_i(t')) \overset{\delta}{\mapsto} (L_i, f_i(t' + \delta))$. The set of states contained in such a run ρ is given as $\{(L_i, f_i(t)) \mid t \in \mathbb{R}, 0 \leq t \leq t_i\}$. The set of all runs of a hybrid system $\mathcal{H}$ with initial state σ is denoted by $\text{runs}(\mathcal{H}, \sigma)$. A *position* π of a run $\rho = \sigma_0 \mapsto_{f_0}^{t_0} \sigma_1 \mapsto_{f_1}^{t_1} \sigma_2 \mapsto_{f_2}^{t_2} \sigma_3 \mapsto_{f_3}^{t_3} \cdots$ is a pair $\pi = (i, r) \in \mathbb{N} \times \mathbb{R}$ such that $0 \leq r \leq t_i$. Positions are ordered lexicographically, i.e., $(i, r) < (j, s)$ if and only if $i < j$ or ($i = j$ and $r < s$). Also, $(i, r) \leq (j, s)$ if and only if $(i, r) < (j, s)$ or ($i = j$ and $r = s$). By $\rho(\pi)$ with $\pi = (i, r)$ we denote the state $(L_i, f_i(r))$.
A run is said to be *non-zeno* if $\sum t_i$ diverges. In the sequel we assume that the runs of the hybrid system under consideration are all non-zeno.[1]

In order to formulate properties of hybrid systems we consider (a fragment of) ICTL, the Integrator Computation Tree Logic [AHH96]. For simplicity we omit the Until-operators in this paper. Their introduction does not cause much more effort, though (see [Non99]).

Given some hybrid system with locations $\mathcal{L}$ and data variables X, the set of ICTL formulas is defined as the smallest set containing all constraint formulas from CF over X, all location names from $\mathcal{L}$, and that is closed under the usual boolean connectives together with temporal (ICTL) operators AG , EF , EG , and AF . Moreover, if Φ is an ICTL formula, z is a new data variable, and $\{L_1, \ldots, L_n\} \in \mathcal{L}$ then $z^{\{L_1, \ldots, L_n\}}.\Phi$ is an ICTL formula as well. Intuitively, the

[1] The assumption of non-zenoness implies that hybrid systems are deadlock-free, i.e., there is no reachable state that has no successor. So-called livelocks, however, are not excluded. This means that we absolutely allow states which have only themselves as future states. The latter case just states that the situation does not change in time, whereas the former case (deadlock) would claim that time itself has come to an end.

temporal operators $AG\ \Phi, AF\ \Phi, EG\ \Phi, EF\ \Phi$ mean "always", "inevitably", "possibly always", and "possibly" respectively. Their formal semantics with respect to hybrid systems is defined below.

Definition 1. *Given a hybrid system* $\mathcal{H} = (X, \mathcal{L}, \mathcal{E}, \mathit{dif}, \mathit{inv}, \mathit{guard}, \mathit{act})$ *and a state* $\sigma = (L, \phi)$*, the semantics of ICTL with respect to* $\mathcal{H}$ *and* σ *is defined as:*

$$
\begin{array}{lll}
\mathcal{H}, \sigma \models c & \textit{iff} & \models \phi(c) \textit{ for constraint formula } c \\
\mathcal{H}, \sigma \models N & \textit{iff} & \textit{locations } N \textit{ and } L \textit{ are identical} \\
\mathcal{H}, \sigma \models \neg\Phi & \textit{iff} & \mathcal{H}, \sigma \not\models \Phi \\
\mathcal{H}, \sigma \models \Phi \wedge \Psi & \textit{iff} & \mathcal{H}, \sigma \models \Phi \ \&\ \mathcal{H}, \sigma \models \Psi \\
 & & \textit{and similarly for the other boolean connectives} \\
\mathcal{H}, \sigma \models AG\ \Phi & \textit{iff} & \forall \rho\ (\rho \in \mathit{runs}(\mathcal{H}, \sigma) \Rightarrow \forall \pi\ (\pi \in \mathit{positions}(\rho) \Rightarrow \mathcal{H}, \rho(\pi) \models \Phi)) \\
\mathcal{H}, \sigma \models EG\ \Phi & \textit{iff} & \exists \rho\ (\rho \in \mathit{runs}(\mathcal{H}, \sigma)\ \&\ \forall \pi\ (\pi \in \mathit{positions}(\rho) \Rightarrow \mathcal{H}, \rho(\pi) \models \Phi)) \\
\mathcal{H}, \sigma \models z^{\mathcal{N}}.\Phi & \textit{iff} & \mathcal{H}^{z^{\mathcal{N}}}, (L, \phi[z/0]) \models \Phi, \qquad \textit{where } \mathcal{N} \subseteq \mathcal{L},
\end{array}
$$

where $EF\ \Phi \equiv \neg AG\ \neg\Phi$ *and* $AF\ \Phi \equiv \neg EG\ \neg\Phi$*. By* $\mathcal{H}^{z^{\{L_1,\dots,L_n\}}}$ *we mean the extended system we obtain by adding the new clock* z *(initialized with 0) which is supposed to run with slope* 1 *within locations* $L_1, \dots, L_n$ *and with slope* 0*, i.e., it is stopped, for all other locations.*

Given a variable valuation ϕ *we define the new valuation* $\phi[z/0]$ *as*

$$
\phi[z/0](x) = \begin{cases} \phi(x) & \textit{if } x \neq z \\ 0 & \textit{otherwise.} \end{cases}
$$

3 The Verification Approach

Here we restrict our view to *linear* hybrid systems, where $\mathit{dif}(L, x)$ is a constant, say k_L^x, for each location L and data variable x. This restriction can easily be weakened to *rectangular* hybrid systems (where $\mathit{dif}(L, x)$ is given as an interval of reals) without any real effort. For a better readability we denote sequences of the form $x_1 + k_L^{x_1}\delta, \dots, x_n + k_L^{x_n}\delta$ by $X + k_L^X\delta$ where $X = \{x_1, \dots, x_n\}$, and, similarly we mean $L(\mathit{act}(T, x_1), \dots, \mathit{act}(T, x_n))$ whenever we write $L(\mathit{act}(T, X))$.

3.1 First-order Theories for Reachability and Inevitability

As usual, an *interpretation* $\Im = (\mathcal{D}, \Im_{\mathcal{L}}, \phi)$ for a first-order theory associated with a hybrid system $\mathcal{H}$ with locations $\mathcal{L}$ has a fixed domain $\mathcal{D}$ (the reals or the rationals, say), a valuation ϕ for the data variables in X, and a meaning function $\Im_{\mathcal{L}}$ for the locations in $\mathcal{L}$ such that $\Im_{\mathcal{L}}(L) \in \mathcal{D}^n$, where n is the number of data variables in X. A *model* of a formula Φ is an interpretation satisfying this formula.

We often also speak of a model as a set of ground atoms of the form

$$
\{L(\Im(t_1), \dots, \Im(t_n)) \mid \Im \models L(t_1, \dots, t_n)\}, \text{where } t_i \text{ are constraint terms}\}
$$

where $\Im$ is a model in the above sense. Interpretations (models) are partially ordered by set-inclusion. A *minimal model* of Φ is a model of Φ such that there exists no proper subset of it that also satisfies Φ.

We now define two different kind of first-order theories for a given hybrid system: one that is responsible for the possible states, and one that is responsible for the unavoidable states.

Definition 2 (Reachability Theory). *Let $\mathcal{H} = (X, \mathcal{L}, \mathcal{E}, \mathit{dif}, \mathit{inv}, \mathit{guard}, \mathit{act})$ be a hybrid system. For each $L \in \mathcal{L}$ we define the first-order theory*

$$\forall X\ L(X) \rightarrow \begin{cases} \mathit{inv}(L)\ \wedge \\ \forall \delta\ (\delta \geq 0 \wedge \mathit{inv}(L)[X/X + k_L^X \delta] \rightarrow L(X + k_L^X \delta))\ \wedge \\ \displaystyle\bigwedge_{T=(L,N)\in\mathcal{E}} \mathit{guard}(T) \rightarrow N(\mathit{act}(T, X)) \end{cases}$$

as the local reachability theory *of L in $\mathcal{H}$, $\mathcal{R}_{\mathcal{H}}^L$ for short. By the* reachability theory of $\mathcal{H}$ *– which we call $\mathcal{R}_{\mathcal{H}}$, or simply $\mathcal{R}$ if $\mathcal{H}$ is clear from the context – we understand the conjunction of all local reachability theories, i.e., $\mathcal{R}_{\mathcal{H}} = \bigwedge_{L\in\mathcal{L}} \mathcal{R}_{\mathcal{H}}^L$.*

What the reachability theory expresses is that (for each location L) (i) the location invariant must hold, (ii) that there is a possible time transition, and (iii) for each outgoing edge: if the enabling guard is true then the target location can be reached provided the corresponding discrete action has been peformed.

Definition 3 (Inevitability Theory). *Let $\mathcal{H} = (X, \mathcal{L}, \mathcal{E}, \mathit{dif}, \mathit{inv}, \mathit{guard}, \mathit{act})$ be a hybrid system. For each $L \in \mathcal{L}$ we define the first-order theory*

$$\forall X\ \ L(X)\ \rightarrow\ \mathit{inv}(L)$$

$$\forall X\ \ L(X)\ \rightarrow \begin{cases} \forall \delta\ \delta \geq 0 \rightarrow L(X + k_L^X \delta)\ \ \vee \\ \exists \delta \begin{cases} \delta \geq 0\ \ \wedge \\ \forall \delta'\ 0 \leq \delta' \leq \delta \rightarrow L(X + k_L^X \delta')\ \ \wedge \\ \displaystyle\bigvee_{T=(L,N)\in\mathcal{E}} \begin{array}{l} \mathit{guard}(T)[X/X + k_L^X \delta]\ \wedge \\ \quad N(\mathit{act}(T, X)[X/X + k_L^X \delta]) \end{array} \end{cases} \end{cases}$$

as the local inevitability theory *of L in $\mathcal{H}$, $\mathcal{I}_{\mathcal{H}}^L$ for short. By the* inevitability theory of $\mathcal{H}$ *– which we call $\mathcal{I}_{\mathcal{H}}$, or simply $\mathcal{I}$ if $\mathcal{H}$ is clear from the context – we understand the conjunction of all local inevitability theories, i.e., $\mathcal{I}_{\mathcal{H}} = \bigwedge_{L\in\mathcal{L}} \mathcal{I}_{\mathcal{H}}^L$.*

The inevitability theory might require some more explanation. In a sense it expresses (for each given state) between which possibilities the system can choose. The first part of any local inevitability theory is trivial. It just guarantees the mere fact that for each location predicate the corresponding location invariant is supposed to hold. The second part is more complicated and more interesting.

Note that, given an arbitrary state represented by the location predicate $L(X)$, either the system remains forever in this location, i.e., $\forall\delta\ \delta \geq 0 \rightarrow L(X + k_L^X \delta)$, or it will sooner or later leave this very location. In the latter case we know that there is a time delay δ after which one of the guards of the outgoing edges is true and until then the system remains within location L. This is exactly what is expressed by the complicated second part of the local inevitability theories.

Intuitively, the reachability theory tells us what *can* be done in certain situations (states), whereas the inevitability theory describes what *must* be done, it collects all the immediate future possibilities.

The importance of these two theories will become apparent from the following Lemma.

Lemma 1. *Given a hybrid system $\mathcal{H}$ and an initial state (L, ϕ).*

- *The (unique) minimal model of $L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}}$ corresponds to the set of states that are reachable from (L, ϕ) in the hybrid system $\mathcal{H}$.*
- *Each minimal model of $L(\phi(X)) \wedge \mathcal{I}_{\mathcal{H}}$ corresponds to the set of states of one of the runs of $\mathcal{H}$.*
- *The set of states of each run of $\mathcal{H}$ forms a model of $L(\phi(X)) \wedge \mathcal{I}_{\mathcal{H}}$.*

Proof. *Can be found in [Non99].*

The above lemma provides us with a formal connection between the reachability theory (inevitability theory) and the reachable (inevitable) states. Briefly, Φ holds always ($AG\ \Phi$) iff Φ holds in all reachable states iff Φ holds for every element in the unique minimal model of the reachability theory (together with the initial state) iff Φ holds for every element of some model of the reachability theory (together with the initial state). This observation leads to the following definition and main theorem.

Definition 4 (Characteristic Constraint Formulas). *The (second-order) formula associated with an ICTL formula Φ, the hybrid system $\mathcal{H}$, and location L, $\lceil \Phi \rceil_{\mathcal{H}}^{L(X)}$, representing a characteristic constraint formula for Φ given $\mathcal{H}$ in L, is recursively defined by*

$$\begin{aligned}
\lceil c \rceil_{\mathcal{H}}^{L(X)} &= c \\
\lceil L' \rceil_{\mathcal{H}}^{L(X)} &= \begin{cases} \top & \textit{if } L \textit{ and } L' \textit{ are identical} \\ \bot & \textit{otherwise} \end{cases} \\
\lceil \neg\Phi \rceil_{\mathcal{H}}^{L(X)} &= \neg \lceil \Phi \rceil_{\mathcal{H}}^{L(X)} \\
\lceil \Phi \wedge \Psi \rceil_{\mathcal{H}}^{L(X)} &= \lceil \Phi \rceil_{\mathcal{H}}^{L(X)} \wedge \lceil \Psi \rceil_{\mathcal{H}}^{L(X)} \\
&\quad \textit{and similarly for the other boolean connectives} \\
\lceil z^{\mathcal{N}}.\Phi \rceil_{\mathcal{H}}^{L(X)} &= \lceil \Phi \rceil_{\mathcal{H}^{z^{\mathcal{N}}}}^{L(X,0)} \textit{ where } \mathcal{N} \subseteq \mathcal{L} \\
\lceil AG\ \Phi \rceil_{\mathcal{H}}^{L(X)} &= \exists L_1, \ldots, L_n\ L(X) \wedge \mathcal{R}_{\mathcal{H}} \wedge \textstyle\bigwedge_{N \in \mathcal{L}}\ \forall X\, N(X) \rightarrow \lceil \Phi \rceil_{\mathcal{H}}^{N(X)} \\
\lceil EG\ \Phi \rceil_{\mathcal{H}}^{L(X)} &= \exists L_1, \ldots, L_n\ L(X) \wedge \mathcal{I}_{\mathcal{H}} \wedge \textstyle\bigwedge_{N \in \mathcal{L}}\ \forall X\, N(X) \rightarrow \lceil \Phi \rceil_{\mathcal{H}}^{N(X)}
\end{aligned}$$

Theorem 1. *Given a hybrid system $\mathcal{H}$ with data variables X, an initial state (L,ϕ) and an ICTL formula Φ. Then*

$$\mathcal{H},(L,\phi) \models \Phi \qquad \textit{iff} \qquad \models \phi\left(\lceil \Phi \rceil_{\mathcal{H}}^{L(X)}\right)$$

Proof. *By induction on the structure of Φ.*
The base cases are trivial. Also in case of a boolean connective there are no problems at all. The induction steps are exemplified for the case $\Phi = AG\,\Psi$. For the other cases see [Non99].

$$\begin{aligned}
&\mathcal{H},(L,\phi) \models AG\,\Psi\\
&\quad \textit{iff}\ \ \mathcal{H},\sigma \models \Psi \textit{ for every } \sigma \textit{ reachable from } (L,\phi)\\
&\quad \textit{iff}\ \ \forall\sigma\ ((L,\phi),\sigma) \in (\overset{*}{\mapsto} \cup \overset{tr}{\mapsto})^* \Rightarrow \mathcal{H},\sigma \models \Psi\\
&\quad \textit{iff}\ \ \forall N,\phi'\ N(\phi'(X)) \in \textit{minMod}(L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}}) \Rightarrow \mathcal{H},(N,\phi') \models \Psi\\
&\qquad\qquad \textit{(Lemma 1)}\\
&\quad \textit{iff}\ \ \exists\Im\ \Im \models L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}}\ \&\ \forall N,\phi'\ (N,\phi') \in \Im \Rightarrow \mathcal{H},(N,\phi') \models \Psi\\
&\quad \textit{iff}\ \ \exists\Im\ \Im \models L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}}\ \&\ \forall N,\phi'\ (N,\phi') \in \Im \Rightarrow\ \models \phi'\left(\lceil\Psi\rceil_{\mathcal{H}}^{N(X)}\right)\\
&\qquad\qquad \textit{(induction hypothesis)}\\
&\quad \textit{iff}\ \ \exists\Im\ \Im \models L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}}\ \&\ \Im \models \textstyle\bigwedge_{N\in\mathcal{L}} \forall X\ N(X) \rightarrow \lceil\Psi\rceil_{\mathcal{H}}^{N(X)}\\
&\quad \textit{iff}\ \ \exists\Im\ \Im \models L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}} \wedge \textstyle\bigwedge_{N\in\mathcal{L}} \forall X\ N(X) \rightarrow \lceil\Psi\rceil_{\mathcal{H}}^{N(X)}\\
&\quad \textit{iff}\ \ \models \exists L_1,\ldots,L_n\ L(\phi(X)) \wedge \mathcal{R}_{\mathcal{H}} \wedge \textstyle\bigwedge_{N\in\mathcal{L}} \forall X\ N(X) \rightarrow \lceil\Psi\rceil_{\mathcal{H}}^{N(X)}\\
&\quad \textit{iff}\ \ \models \phi\left(\lceil AG\,\Psi\rceil_{\mathcal{H}}^{L(X)}\right)
\end{aligned}$$

3.2 Eliminating Locations

Theorem 1 tells us that we can solve a hybrid system verification problem by proving the satisfiability of some suitable first-order theory, or equivalently, by showing the validity of some corresponding second-order formula. The *Elimination Theorem* below helps us in this respect, for it allows us to transform a given second-order formula into an equivalent first-order formula (if this is at all possible).[2]

[2] In general, second-order formulas do not necessarily have a (finite or infinite) first-order equivalent. However, in case the second-order formula is of the form $\exists P\ \Phi$, where Φ is a first-order formula that is Horn in P, then we know that there exists a first-order equivalent (which may be infinite, though). Note that proving Safety properties, i.e. proving the validity of second-order formulas involving reachability theories, are just of this form.

Theorem 2 (Elimination Theorem). *Let Φ and Ψ be two first-order formulas that are positive with respect to the predicate symbol L. Then*

$$\exists L\ \big(\forall \overline{x}\,(L(\overline{x}) \to \Phi) \wedge \Psi\big) \qquad \equiv \qquad \Psi\left[L(\overline{\alpha})/\left(\nu L(\overline{x}).\Phi\right)\frac{\overline{x}}{\overline{\alpha}}\right]$$

$$\textit{where} \quad \nu L(\overline{x}).\Phi(L) = \bigwedge_{i \leq \omega} \Phi^i(\top) \qquad \textit{with} \quad \Phi^0(\top) = \top, \Phi^{n+1}(\top) = \Phi(\Phi^n(\top))$$

The proof of this Theorem can be found in [NS95] (but also see [NS99] and [NOS99]). There also some generalizations and dual forms are examined. For the purpose of this paper, however, the above form suffices.

The Elimination Theorem tells us that any second-order formula of the form $\exists L\ \big(\forall \overline{x}\,(L(\overline{x}) \to \Phi) \wedge \Psi\big)$ is equivalent to Ψ with every occurrence of L (with actual argument list $\overline{\alpha}$) within Ψ replaced by the greatest fixpoint of Φ (after instantiating the abstract parameters with the actual arguments). Notice that the second-order formulas we are dealing with are indeed of the form required. Therefore, with each application of the Elimination Theorem we get rid of one of the existentially quantified predicate symbols. Now, since these predicate symbols are just the location names of the hybrid system under consideration, each application of the Elimination Theorem also eliminates one of the locations.

Evidently, it cannot be guaranteed that the fixpoint computation will terminate in general. However, it can easily be shown (see [Non99]) that in case we are about to eliminate a location which has no outgoing edge leading to itself, the fixpoint computation will definitely terminate after two iterations.

Coming back to the example on page 2 where we wanted to examine the effect of eliminating location L_2, we now know that we have to compute – in fact, find a first-order equivalent for – the second-order formula

$$\exists L_2 \left[\begin{array}{l} \forall x, y\ L_1(x,y) \to x \leq y \to L_2(x,y)\ \wedge \\ \forall x, y\ L_2(x,y) \to \left\{\begin{array}{l} x \leq y\ \wedge \\ x + y \leq 10\ \wedge \\ \forall \delta\ (\delta \geq 0 \wedge x + 2\delta \leq y + \delta \to L_2(x+2\delta, y+\delta))\ \wedge \\ x = y \to L_3(0,0) \end{array}\right\} \end{array}\right.$$

The five conjuncts of the above second-order formula describe the transition from L_1 to L_2, the location invariant for L_2, the property to be proved, the time transition for location L_2, and the edge transition from L_2 to L_3 respectively. In order to apply the Elimination Theorem to this second-order formula, let $\Psi = \forall x, y\ L_1(x,y) \to x \leq y \to L_2(x,y)$ and $\Phi = x \leq y \wedge x + y \leq 10 \wedge \forall \delta\ (\delta \geq 0 \wedge x + 2\delta \leq y + \delta \to L_2(x+2\delta, y+\delta)) \wedge x = y \to L_3(0,0)$. Then $\Phi^0(\top) = \top$, $\Phi^1(\top) = x \leq y \wedge x + y \leq 10 \wedge x = y \to L_3(0,0)$, $\Phi^2(\top) = \Phi^1(\top) \wedge x \leq y \to 2y \leq 5 + x$, and $\Phi^3(\top) = \Phi^2(\top)$ as can easily be checked by the reader. We thus have found the fixpoint and substitute it for L_2 in Ψ resulting in

$$\begin{array}{l} \forall x, y\ L_1(x,y) \to x \leq y \to L_3(0,0) \\ \forall x, y\ L_1(x,y) \to x \leq y \to 2y \leq x + 5 \end{array}$$

The first formula describes just the new edge to be introduced. The second formula, on the other hand, tells us about the necessary and sufficient condition on the data variables for location L_1 such that it would be impossible to violate $x + y \leq 10$ in location L_2.

3.3 Examples and Experimental Results

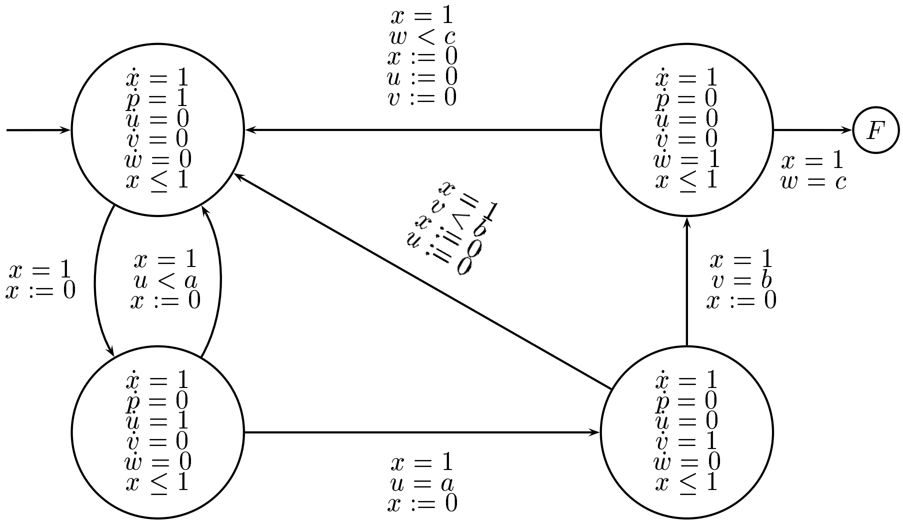
There exists a prototype implementation of the Elimination Approach (for proving safety-properties) written in Sicstus-Prolog with the CLP(Q,R)-library for constraint handling. It has been tested on a lot of examples known from the literature (or taken from hybrid system verifier distributions). The experimental results can briefly be summarized as follows: as already expected in the introduction, standard forward reachability (if it at all terminates) can hardly be beaten in case we are about to prove safety properties for non-trivial systems that only require a single pass through the reachable locations. This is the case for instance for the famous "audio-protocol"-example.[3] For other, unfortunately still trivial systems like the "Leaking Gas Burner" or the "Billiards"-example, the Elimination Approach showed a slightly better behavior than standard reachability analysis.[4] However, in such cases, where safety properties can be proved in milliseconds anyway, this can hardly be called evidence. The lack of non-trivial hybrid system in the literature that require several passes through some of their locations made us compose our own examples. They are designed as simple as possible such that they may serve to illustrate the effect of the Elimination Approach compared to reachability analysis methods. Two such examples are given below.

A Silly Multiplier This is an example within which three numbers a, b, and c are to be multiplied and the final product is stored in the data variable p. The multiplication is performed by successively adding 1 to p, similar to the nested for-loop

$$\begin{array}{l} \text{for } (w = 0; w < c; w{+}{+}) \\ \quad \text{for } (v = 0; v < b; v{+}{+}) \\ \quad\quad \text{for } (u = 0; u < a; u{+}{+}) \\ \quad\quad\quad p := p + 1 \end{array}$$

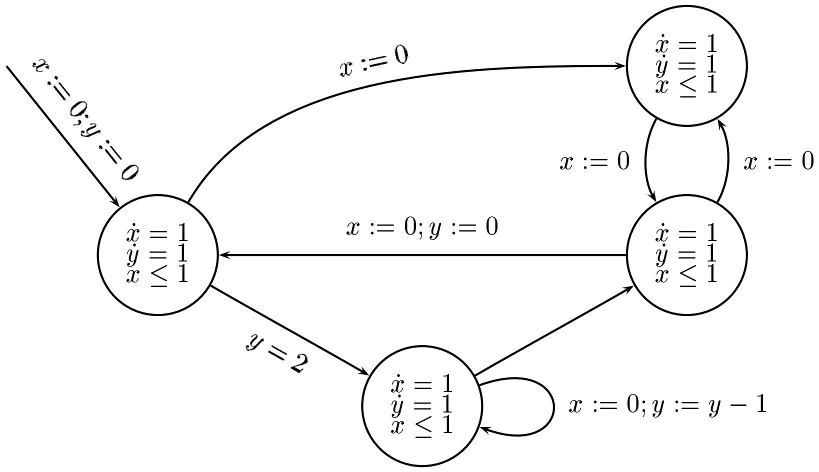
[3] In case of more trivial such examples like the "Water-Level-Monitor" or the "Railroad-Gate-Controller" there is not so much of a difference.

[4] It should be noted here that it is in fact very easy to compare the Elimination Approach with standard reachability analysis methods for, in a sense, reachability analysis can be viewed as a special case of the Elimination Approach: we just have to move the location names to the argument list as a further additional argument (leaving a single unique dummy predicate as the only remaining predicate symbol). This then leads to backward reachability, whereas, by using a dual form of the Elimination Theorem, we can also get forward reachability (see also [Non99]).



It is to be shown that the location F can be reached – after all, as soon as F is reached, the data variable p contains the multiplication result we are interested in. (Backward or forward) reachability analysis in a sense simulates the behavior of the multiplier. I.e., since this system is fully deterministic, it takes a walk through the whole computation. Evidently, this is very time consuming even if we only attempted to compute $10 \times 10 \times 10$. Now, compare this with the Elimination Approach. It takes two steps to eliminate the top left location, another $2a$ iterations for the bottom left location, some $2b$ iterations for the bottom right location, and finally, $2c$ iterations for the top right location. We then end up with a single location, namely F, for which the data variable p is initialized with the product $a \times b \times c$. Thus, even for big numbers a, b, and c the elimination approach is able to fulfill its task in a very short amount of time, whereas any kind of reachability analysis requires much more effort (essentially $a + b + c$ versus $a \times b \times c$).

An Example where Reachability Fails The particularity about the following example is that it contains an "impossible" transition, i.e., one of the locations – the one at the bottom – is unreachable because the guard ($y = 2$) of the transition that may lead to this very location will never be enabled. In a sense, forward reachability analysis detects this, though rather indirectly, for it never tries to compute states involving this location. However, forward reachability nevertheless does not terminate, since the data variable y may become arbitrarily big and therefore the fixpoint computation finds new possible states in each iteration. At the first glance, backward reachability might have a better chance. Suppose we were about to prove that $x \leq y$ is an overall invariant of the system. If there were not the bottom location, backward reachability would have no problem to detect that the invariant indeed holds. But it this very location (or actually the transition that leads to itself) that prevents backward reachability from termination.



As for the Elimination Approach, recall that the real purpose of the Elimination Theorem is to provide us with a model for the given first-order theory. The Elimination Theorem is just one of the possible means we can imagine that may help us finding such a model. There are, however, cases for which we can (even by mere syntactic examinations) find out that a given first-order formula indeed has a (trivial) model. This is the case, for instance, if we reach a situation (maybe after having performed some eliminations) where every conjunct of the intermediate result contains a negative location predicate literal. In this case we can conclude that all locations that have not yet been eliminated are unreachable. This is in fact very common. The famous "Railroad-Gate-Controller" for example, has 22 (compound) locations. But only 7 of them are in fact reachable. Our implementation of the Elimination Approach first eliminates these seven reachable locations and then detects that all others are unreachable and terminates with success. The same happens for the example above: first the top three locations get eliminated and then the unreachability of the final (bottom) location is detected. Thus the system terminates with success.

4 Summary and Conclusion

We presented a hybrid systems verification methodology that is based on a successive location elimination procedure that allows us to simplify the system until it becomes trivial. This approach turns out to be particularly useful in case standard forward or backward reachability analysis methods would require to pass some of the locations several times (long lasting loops). There are even examples where a reachability analysis fails, whereas the Elimination Approach terminates successfully. However, there also are some non-trivial examples where forward reachability seems unbeatable, namely systems which contain quite a huge amount of locations, yet require merely a single pass in order to compute the set of reachable states. This observation leads to the conjecture that it might make sense to suitably combine the two methodologies: in case a non-trivial system (with many locations) contains long-lasting loops that prevent a reachability analysis from termination within some reasonable amount of time, it certainly makes sense to try and eliminate such loops with the help of the Elimination Approach and maybe to proceed with the reachability analysis after that.

References

[ACD90] R. Alur, C. Courcoubetis, and D. L. Dill. Model checking for real-time systems. In *Proceedings of the 5th Annual Symposium on Logic in Computer Science*, pages 414–425, 1990.

[ACH+95] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifaksi, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[ACHH93] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, pages 209–229. Springer Verlag, Lecture Notes in Computer Science, vol. 736, 1993.

[AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[AH92] 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*, pages 74–106. Springer Verlag, New York, LNCS 600, 1992.

[AHH96] Rajeev Alur, Thomas A. Henzinger, and Pei-Hsin Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22(3):181–201, 1996.

[AHS96] R. Alur, T. A. Henzinger, and E. Sontag, editors. *Hybrid Systems III.* Lecture Notes in Computer Science, Springer Verlag, 1996.

[ANKS95] P. Antsaklis, A. Nerode, W. Kohn, and S. Sastry, editors. *Hybrid Systems II.* Lecture Notes in Computer Science, vol. 999, Springer Verlag, 1995.

[GNRR93] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors. *Hybrid Systems.* Springer Verlag, Lecture Notes in Computer Science, vol. 736, 1993.

[Hen96] T. A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th LICS*, pages 278–292. IEEE Comp. Soc. Press, 1996.

[HNSY92] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. In *Proceedings of the 7th Annual Symposium on Logic in Computer Science*, pages 394–406. IEEE Computer Society Press, New York, 1992.

[Non99] Andreas Nonnengart. A deductive model checking approach for hybrid systems. Technical Report MPI-I-1999-2-006, Max-Planck-Institute for Computer Science, Saarbrücken, Germany, November 1999. Available via http://www.mpi-sb.mpg.de/.

[NOS99] Andreas Nonnengart, Hans Jürgen Ohlbach, and Andrzej Szałas. Elimination of predicate quantifiers. In Hans Jürgen Ohlbach and Uwe Reyle, editors, *Logic, Language and Reasoning – Essays in Honour of Dov Gabbay.* Kluwer, Dordrecht, Netherlands, 1999. ISBN: 0-7923-5687-X.

[NS95] Andreas Nonnengart and Andrzej Szałas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory.

Technical Report MPI-I-95-2-007, Max-Planck-Institute for Computer Science, Saarbrücken, Germany, March 1995. Available via: http://www.mpi-sb.mpg.de/.

[NS99] Andreas Nonnengart and Andrzej Szalas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. in: [Orł99], 1999.

[Orł99] Ewa Orłowska, editor. *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, volume 24 of *Studies in Fuzziness and Soft Computing*. Physica-Verlag, c/o Springer Verlag, 1999. ISBN: 3-7908-1164-5.