Verification of Hybrid Systems in the Manna-Pnueli Framework Lecture Notes Washington University at St Louis, November 16, 2006

Henny Sipma
Stanford University

November 20, 2006

Contents

1 Introduction **2**

2 Computational Models **3**

3 Specification **13**

4 Verification **14**

5 Well-behaved Systems **17**

6 Invariant Generation **21**

7 Summary **24**

1 Introduction

The Manna-Pnueli Verification Framework

The Manna-Pnueli framework for verification contains the following components:

- Computational Model: fair transition systems;
- Specification Language: linear-time temporal logic
- Proof rules to establish that a system satisfies its temporal specification;
- Algorithmic techniques:
  - Model checking for decidable subclasses;
  - Automatic invariant generation to support deductive verification
  - Automatic generation of ranking functions to prove termination

The basic model of time is the natural numbers.

The computational model, proof rules, as well as the automatic invariant generation have been implemented in the verification tool, STeP, the Stanford Temporal Prover.

Verification of Hybrid Systems

The objectives in the verification of hybrid systems are

- Model of time: $\mathcal{R}^{\geq 0} \cup \{0\}$
- Reuse of verification methods and tools for discrete systems
- No change in specification language

The approach we have followed is to represent *continuous* $\mathcal{R}^{\geq 0} \mapsto \Sigma$ behaviors by infinitely many *discrete* $\mathcal{N} \mapsto \Sigma$ behaviors such that

- every time point in the continuous behavior is represented in *at least one* discrete behavior, and
- some time points in the continuous behavior are represented in *all* discrete behaviors.

2 Computational Models

Fair Transition Systems

A fair transition system $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{J} \rangle$ contains the following components:

- V, a finite set of variables (of any type). A *state* is a type-consistent interpretation of the system variables. The set of all states is designated by Σ.
- Θ, the *initial condition*: an assertion (first-order formula) over V that characterizes the initial states.
- $\mathcal{T}$, a finite set of *transitions*. Each transition $\tau \in \mathcal{T}$ is a function

$$\tau : \Sigma \mapsto 2^{\Sigma}$$

  mapping each state $s \in \Sigma$ into a (possibly empty) set of τ-successor states, $\tau(s) \subseteq \Sigma$. Each transition τ is defined by a *transition relation* $\rho_\tau(V, V')$, a first-order formula in which the unprimed variables refer to the values in the current state s, and the primed variables refer to the values in the next state s'.
- $\mathcal{J} \subseteq \mathcal{T}$: the set of *just* (weakly fair) transitions.

A *run* of a fair transition system $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{J} \rangle$ is an infinite sequence of states $\sigma : s_0, s_1, \ldots$ such that the following holds

- **Initiation:** $s_0 \models \Theta$;
- **Consecution:** for all $i \geq 0$:

$$(s_i, s_{i+1}) \models \rho_\tau \qquad \text{for some } \tau \in \mathcal{T}$$

A *computation* is a run such that the following holds

- **Fairness:** a transition $\tau \in \mathcal{J}$ is not continuously enabled without being taken.

Fair Transition System: Example

V	$\{x\}$		
Θ	$x = 0$		
$\mathcal{T}$	$\{\tau_1, \tau_2, \tau_I\}$	with	ρ_{τ_1}: $x \geq 0 \wedge x' = x - 1$
			ρ_{τ_2}: $x' = x + 1$
			ρ_{τ_I}: $x' = x$
$\mathcal{J}$	$\{\tau_1, \tau_2\}$		

Some computations:

$$\sigma_1 : \langle 0 \rangle , \langle 1 \rangle , \langle 2 \rangle , \langle 1 \rangle , \langle 2 \rangle , \ldots$$

$$\sigma_2 : \langle 0 \rangle , \langle -1 \rangle , \langle 0 \rangle , \langle 0 \rangle , \langle 1 \rangle , \langle 2 \rangle , \ldots$$

Some runs that are *not* computations:

$$\sigma_x : \langle 0 \rangle , \langle 0 \rangle , \langle 0 \rangle , \langle 0 \rangle , \ldots$$

$$\sigma_y : \langle 0 \rangle , \langle 1 \rangle , \langle 2 \rangle , \langle 3 \rangle , \ldots$$

The run σ_x violates the fairness of both transitions τ_1 and τ_2; the run σ_y violates the fairness of transition τ_1.

Clocked Transition System

A clocked transition system (CTS) $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{J}, \Pi \rangle$ has the following components:

- V: A finite set of typed *system variables* partitioned in a set C of real-valued clocks and a set D of discrete variables that can be of any type. We assume that C contains a variable T called the *masterclock*. The purpose of the masterclock is to keep track of global time.
- Θ: The *initial condition*, an assertion over V characterizing the initial states. We assume that Θ implies $T = 0$, that is, global time starts at 0.
- $\mathcal{T}$: A finite set of *transitions*. We assume that for each transition $\tau \in \mathcal{T}$, ρ_τ implies $T' = T$, that is, we assume that discrete transitions happen in zero time.
- $\mathcal{J} \subseteq \mathcal{T}$: A finite set of just (weakly fair) transitions.
- Π: The *time-progress condition*. An assertion over V to specify a global restriction on the progress of time.

The (continuous) *behaviors* of a CTS are *multi-valued* functions over the nonnegative reals:

$$(\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma$$

In STeP the masterclock T is automatically added, as well as its constraints in the initial condition and the transition relation.

Some Definitions

- **Height:** For a behavior $\sigma_T : (\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma$ and $t \geq 0$,

$$ht(\sigma_T, t) = \mathbf{min}\{\ k \mid \forall j > k \ .\ \sigma_T(t, k) = \sigma_T(t, j)\}$$

- **Atomic interval:** $[\ell, h]$ with $\ell, h \geq 0$ and $\ell < h$ is an atomic interval in σ_T if

  $$ht(\sigma_T, \ell) > 0 \ \text{ and } \ ht(\sigma_T, h) > 0 \ \text{ and } \ \forall t \ . \ \ell < t < h \ . \ ht(\sigma_T, t) = 0$$

  $[\ell, -)$ is an atomic interval if

  $$ht(\sigma_T, \ell) > 0 \ \text{ and } \ \forall t > \ell \ . \ ht(\sigma_T, t) = 0$$

- **Countable Variability:** A behavior $\sigma_T : (\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma$ is countably variable if
  - for all $t \geq 0$, $ht(\sigma_T, t)$ exists, and
  - except for countably many $t \geq 0$, $ht(\sigma_T, t) = 0$.

Behavior of CTS

A mapping $\sigma_T : \ (\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma$ is a behavior of a CTS $\Phi : \ \langle V, \Theta, \mathcal{T}, \mathcal{J}, \Pi \rangle$ if

- Countable variability: σ_T is countably variable;
- Initiation: $\sigma_T(0, 0) \models \Theta$;
- Discrete Consecution: for all $t \geq 0$, for all $j \geq 0$,

  $$\langle \sigma_T(t, j), \sigma(t, j+1) \rangle \models \rho_\tau \qquad \text{for some } \tau \in \mathcal{T}$$

- Continuous evolution: for every atomic interval $[\ell, h]$ $([\ell, -))$, for all $t : \ell \leq t \leq h$,

  for all $c \in C$: $\quad \sigma_T(t, 0)[c] = \sigma_T(\ell, ht(\ell))[c] + t - \ell$
  for all $d \in D$: $\quad \sigma_T(t, 0)[d] = \sigma_T(\ell, ht(\ell))[d]$

- Time Progress: for all $t \geq 0$, $\sigma_T(t, 0) \models \Pi$.
- Fairness: it is not the case that a just transition is continuously enabled without being taken.

Clocked Transition System: Example

Clocked transition system Φ:

C	$\{c, T\}$	(clocks)
D	$\{x\}$	(discrete variables)
Θ	$c = 0 \ \wedge \ x = 0 \ \wedge \ T = 0$	
$\mathcal{T}$	$\{\tau_1, \tau_I\}$ with	
	$\rho_{\tau_1} : \ c \geq 1 \ \wedge \ c' = 0 \ \wedge \ x' = x + 1 \ \wedge \ T' = T$	
Π:	$c \leq 2$	
$\mathcal{J}$:	$\{\tau_1\}$	

Figure 2 shows two behaviors of this system. Note that c can be anywhere between 1 and 2 when a discrete transition is taken. It is at least 1 because of the enabling condition of τ_1 and at most 2 because of the progress condition.

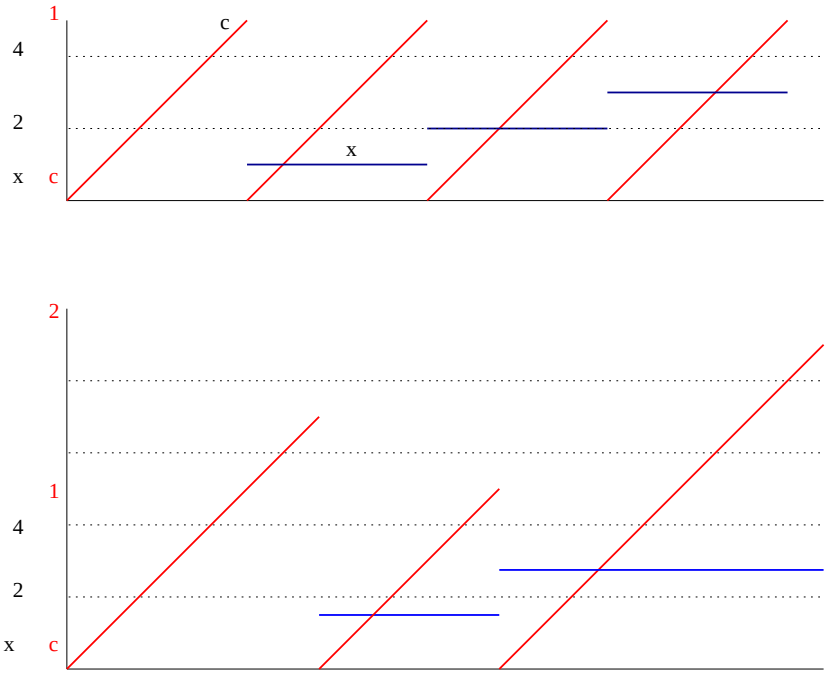

Figure 1: Two behaviors of Φ

Associated Transition System

Given a CTS $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{J}, \Pi \rangle$, its associated fair transition system is:

$$\Phi_T : \langle V, \Theta, \mathcal{T} \cup \{tick\}, \mathcal{J} \cup \{tick\} \rangle$$

with

$$\rho_{tick}[\Delta] : \left(\begin{array}{c} D' = D \ \wedge \ C' = C + \Delta \\ \wedge \\ \Delta > 0 \\ \wedge \\ \forall t \in [0 \ldots \Delta] \ . \ \Pi(D, C + t) \end{array} \right)$$

Notes:

- ρ_{tick} is parameterized by Δ, the amount that time is advanced. It represents an infinite number of transitions, one for each value of Δ.
- $D' = D$ stands for $d' = d$, for all $d \in D$. Thus, discrete variables cannot change value during a time step.
- $C' = C + \Delta$ stands for $c' = c + \Delta$, for all $c \in C$. Thus all clocks are incremented uniformly by the time duration of the interval.
- The progress condition must hold throughout the interval.

Computation of a CTS

An infinite sequence of states $\sigma : s_0, s_1, s_2, \ldots$ is a computation of a CTS Φ if

- σ is a computation of the associated fair transition system Φ_T, and
- σ is time divergent: the sequence $s_0[T], s_1[T], s_2[T], \ldots$ grows beyond any bound.

Clocked Transition System: Example

The following clocked transition system

C	$\{c, T\}$	(clocks)
D	$\{x\}$	(discrete variables)
Θ	$c = 0 \ \wedge \ x = 0 \ \wedge \ T = 0$	
$\mathcal{T}$	$\{\tau_1, \tau_I\}$ with	
	$\rho_{\tau_1} : \ c \geq 1 \ \wedge \ c' = 0 \ \wedge \ x' = x + 1 \ \wedge \ T' = T$	
Π:	$c \leq 2$	
$\mathcal{J}$:	$\{\tau_1\}$	

has as associated transition system:

V $\quad \{x, c, T\}$

Θ $\quad c = 0 \ \wedge \ x = 0 \ \wedge \ T = 0$

$\mathcal{T}$ $\quad \{\tau_1, \tau_I, tick\}$ with

$\rho_{\tau_1} : \ c \geq 1 \ \wedge \ c' = 0 \ \wedge \ x' = x + 1 \ \wedge \ T' = T$

$$\rho_{tick}[\Delta] = \left(\begin{array}{c} x' = x \ \wedge \ c' = c + \Delta \ \wedge \ T' = T + \Delta \\ \wedge \\ \Delta > 0 \\ \wedge \\ \forall t \in [0 \ldots \Delta] \ . \ c + t \leq 2 \end{array} \right)$$

$\mathcal{J}$: $\quad \{\tau_1, tick\}$

Note: It is easy to see that the conjunct $\forall t \in [0 \ldots \Delta] \ . \ c + t \leq 2$ in the tick transition can be simplified to $c + \Delta \leq 2$. This is generally true for convex progress conditions.

Some sample computations representing the behavior shown earlier

$$\sigma_1 : \ \langle 0, 0, 0 \rangle \overset{tick[1]}{\rightarrow} \langle 1, 0, 1 \rangle \overset{\tau_1}{\rightarrow} \langle 0, 1, 1 \rangle \overset{tick[1]}{\rightarrow} \langle 1, 1, 2 \rangle \rightarrow \ldots$$

$$\sigma_2 : \ \langle 0, 0, 0 \rangle \overset{tick[0.5]}{\rightarrow} \langle 0.5, 0, 0.5 \rangle \overset{tick[0.1]}{\rightarrow} \langle 0.6, 0, 0.6 \rangle \overset{tick[0.4]}{\rightarrow} \langle 1, 0, 1 \rangle \rightarrow \ldots$$

Not computations:

$$\sigma_x : \ \langle 0, 0, 0 \rangle \overset{tick[0.5]}{\rightarrow} \langle 0.5, 0, 0.5 \rangle \overset{tick[0.25]}{\rightarrow} \langle 0.75, 0, 0.75 \rangle \overset{tick[0.125]}{\rightarrow} \langle 0.875, 0, 0.875 \rangle \rightarrow \ldots$$

$$\sigma_y : \ \langle 0, 0, 0 \rangle \overset{\tau_I}{\rightarrow} \langle 0, 0, 0 \rangle \overset{\tau_I}{\rightarrow} \langle 0, 0, 0 \rangle \overset{\tau_I}{\rightarrow} \ldots$$

Run σ_x will never advance beyond $T = 1$ (in fact it will never even reach $T = 1$) and thus violates the time divergence condition. Run σ_y violates both the fairness condition and the time divergence condition.

Behaviors versus Computations

Notation: For a CTS Φ we write

- $\mathcal{L}_B(\Phi)$: the set of *continuous* behaviors $((\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma)$ of Φ.
- $\mathcal{L}(\Phi)$: the set of *discrete* behaviors $(\mathcal{N} \mapsto \Sigma)$ of the associated fair transition system of Φ.

Given a CTS Φ, for every $\sigma_T \in \mathcal{L}_B(\Phi)$ there exists a set of computations $S \subseteq \mathcal{L}(\Phi)$ such that

- **Initial state:** for all $\sigma : s_0, s_1, \ldots \in S$, $s_0 = \sigma_T(0, 0)$.
- σ **is a sampling of** σ_T: for all $\sigma : s_0, s_1, \ldots \in S$ there exists an infinite sequence (t_0, k_0), (t_1, k_1), (t_2, k_2), ..., with $(t_{i+1}, k_{i+1}) > (t_i, k_i)$, such for all $i \geq 0$, $s_i = \sigma_T(t_i, k_i)$.

- **Discrete transitions are represented in all:** for all $\sigma : s_0, s_1, \ldots \in S$, for all $t \geq 0$ such that $ht(\sigma_T, t) > 0$, for all $j : 0 \ldots ht(\sigma_T, t)$, there exists $i \geq 0$ such that $s_i = \sigma(t, j)$.
- **All time points are represented in some:** for all $t \geq 0$ there exists $\sigma : s_0, s_1, \ldots \in \sigma_T$ such that $s_i = \sigma_T(t, 0)$ for some $i \geq 0$.

Hybrid Transition System

A hybrid transition system HTS $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{A}, \mathcal{J}, \Pi \rangle$ has the following components:

- V: A finite set of typed *system variables* partitioned in a set I of real-valued integrators (continuous) variables, a set C of real-valued clocks and a set D of discrete variables that can be of any type. We assume that C contains a variable T called the *masterclock*.
- Θ: The *initial condition*, an assertion over V characterizing the initial states. We assume that Θ implies $T = 0$.
- $\mathcal{T}$: A finite set of *transitions*. We assume that for each transition $\tau \in \mathcal{T}$, ρ_τ implies $T' = T$.
- $\mathcal{A}$: A finite set of *activities*. Each activity $\alpha \in \mathcal{A}$ is described by an activity relation
$$\rho_\alpha : \ p_\alpha \rightarrow \dot{I}^\alpha = F^\alpha(V)$$
where $p_\alpha(D)$ is a predicate over discrete variables only, and F^α is a function over all system variables that can be integrated. We assume that the conditions p^α are exhaustive, that is, $\bigvee_{\alpha \in \mathcal{A}} p_\alpha$ is valid, and that they are mutually exclusive, that is, $p_{\alpha_1} \wedge p_{\alpha_2}$ is unsatisfiable for $\alpha_1 \neq \alpha_2$.
- $\mathcal{J} \subseteq \mathcal{T}$: A finite set of just (weakly fair) transitions.
- Π: The *time-progress condition*. An assertion over V to specify a global restriction on the progress of time.

Behavior of Hybrid Transition System

A mapping $\sigma_T : \ (\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma$ is a behavior of a HTS $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{A}, \mathcal{J}, \Pi \rangle$ if

- Countable variability: σ_T is countably variable;
- Initiation: $\sigma_T(0, 0) \models \Theta$;
- Discrete Consecution: for all $t \geq 0$, for all $j \geq 0$,
$$\langle \sigma_T(t, j), \sigma(t, j + 1) \rangle \models \rho_\tau \qquad \text{for some } \tau \in \mathcal{T}$$
- Continuous evolution: for every atomic interval $[\ell, h]$ $([\ell, -))$, for all $\delta : \ell \leq \delta \leq h$,

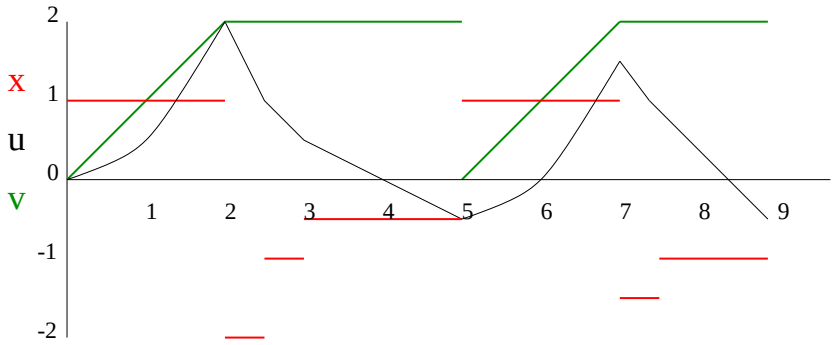

Figure 2: A possible behavior of HTS ACC

$$\text{for all } i \in I: \qquad \sigma_T(t,0)[i] = \sigma_T(\ell, ht(\ell))[i] + \int_0^{\delta-\ell} F^\alpha dt$$

$$\text{for all } c \in C: \qquad \sigma_T(t,0)[c] = \sigma_T(\ell, ht(\ell))[c] + \delta - \ell$$

$$\text{for all } d \in D: \qquad \sigma_T(t,0)[d] = \sigma_T(\ell, ht(\ell))[d]$$

- Time Progress: for all $t \geq 0$, $\sigma_T(t,0) \models \Pi$.
- Fairness: it is not the case that a just transition is continuously enabled without being taken.

Note. Clocks can be considered part of the integrators. The reason we keep them separate in STeP is for convenience. It saves the user from having to specify the evolution function for all clock variables.

Example: Program ACC

V $\quad \{x, u, v, T\}$ with $D = \{x\}$, $C = \{T\}$, $I = \{u, v\}$

Θ $\quad x = 1 \ \wedge \ u = 0 \ \wedge \ v = 0$

$\mathcal{T}$ $\quad \{\tau_1, \tau_2, \tau_I\}$ with

$\quad \rho_{\tau_1} : \ u \geq 0 \ \wedge \ x' = -u$

$\quad \rho_{\tau_2} : \ u < 0 \ \wedge \ x' = 1 \ \wedge \ v' = 0$

$\mathcal{A}$ $\quad \{\alpha_1, \alpha_2\}$ with

$$\rho_{\alpha_1} : \ x > 0 \ \rightarrow \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ x \end{pmatrix}$$

$$\rho_{\alpha_2} : \ x \leq 0 \ \rightarrow \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} x \\ 0 \end{pmatrix}$$

Π $\quad u \leq 2 \ \wedge \ u \geq -2$

A possible behavior of this HTS is shown in figure 2.

Associated Fair Transition System of HTS

Given an HTS $\Phi : \langle V, \Theta, \mathcal{T}, \mathcal{A}, \mathcal{J}, \Pi \rangle$, its associated fair transition system is

$$\Phi_T : \langle V, \Theta, \mathcal{T} \cup \{\tau_\alpha \mid \alpha \in \mathcal{A}\}, \mathcal{J} \cup \{\tau_\alpha \mid \alpha \in \mathcal{A}\} \rangle$$

with

$$\rho_{\tau_\alpha}[\Delta] : \left(\begin{array}{c} D' = D \ \wedge \ C' = C + \Delta \ \wedge \ \Delta > 0 \ \wedge \ p_\alpha \\ \wedge \\ I' = I + \int_0^\Delta F^\alpha dt \\ \wedge \\ \forall \delta \in [0 \ldots \Delta] \ . \ \Pi(D, C + \delta, I + \int_0^\delta F^\alpha dt) \end{array} \right)$$

Differential Inclusions

Sometimes derivatives of continuous variables are not known exactly, but only known to lie in a certain range. This can be described by *differential inclusions*.

Activity

In this case the activity can be described by the triple

$$\langle p_\alpha(D), \ \dot{I_d^\alpha} = F^\alpha(V), \ \dot{I_b^\alpha} \in [F_\ell^\alpha(V), F_u^\alpha(V)] \rangle$$

where I_d is the set of variables described by a single set of differential equations (they are deterministic) and I_b^α is the set of variables described by an inclusion (their derivative is bounded).

Continuous Evolution

The continuous evolution in atomic intervals (ℓ, h), $(\ell, -)$ now becomes
for all $i \in I_d^\alpha$, for all $\delta \in (\ell, h)$

$$\sigma_T(\delta, 0)[i] = \sigma_T(\ell, ht(\ell))[i] + \int_0^{\delta - \ell} F^\alpha dt$$

for all $i \in I_b^\alpha$, for all $\delta \in (\ell, h)$

$$\sigma_T(\delta, 0)[i] \ \geq \ \sigma_T(\ell, ht(\ell)[i] + \int_0^{\delta - \ell} F_\ell^\alpha dt$$

and

$$\sigma_T(\delta, 0)[i] \ \leq \ \sigma_T(\ell, ht(\ell)[i] + \int_0^{\delta - \ell} F_u^\alpha dt$$

Activity Transition Relation

The activity transition relation now becomes

$$\rho_{\tau_\alpha}[\Delta] : \left(\begin{array}{c} D' = D \ \wedge \ C' = C + \Delta \ \wedge \ \Delta > 0 \\ \wedge \\ I_d^{\alpha\prime} = I + \int_0^\Delta F^\alpha dt \\ \wedge \\ I + \int_0^\Delta F_\ell^\alpha dt \ \ \leq \ \ I_b^{\alpha\prime} \ \ \leq I + \int_0^\Delta F_u^\alpha dt \\ \wedge \\ \forall E : \bar{\mathcal{R}}. \forall \delta \in [0 \ldots \Delta] \ . \\ \left(\begin{array}{c} I_b^\alpha + \int_0^\delta F_\ell^\alpha \ \leq E \ \leq I_b^\alpha + \int_0^\delta F_u^\alpha \\ \rightarrow \\ \Pi(D, \ C + \delta, \ I_d^\alpha + \int_0^\delta F^\alpha dt, \ E) \end{array} \right) \end{array} \right)$$

3 Specification

Specification Language

As specification language we can use regular LTL (Linear-time temporal logic) with arbitrary references to clocks (including the masterclock T) and continuous variables.

Examples

- Bounded response: (within d time units)

$$\forall x \;.\; \Box((p \;\wedge\; T = x) \rightarrow \Diamond(q \;\wedge\; T \leq x + d))$$

  if p holds now (save the current time in variable x), then eventually there will be a time when q is true and the then current time will be less than or equal to the time saved plus the response time.

- Minimum separation: (by at least d time units)

$$\forall x \;.\; \Box((p \;\wedge\; T = x) \;\rightarrow\; \Box(q \;\rightarrow\; T \geq x + d))$$

LTL operators

$\Box\,\varphi$	always φ
$\Diamond\,\varphi$	eventually φ
$\varphi\mathcal{U}\psi$	φ until ψ
$\varphi\mathcal{W}\psi$	φ waitfor ψ

Specification Language on Continuous Behaviors

Interpretation of LTL formulas over continuous $(\mathcal{R}^{\geq 0}, \mathcal{N}) \mapsto \Sigma$ behaviors.

$$\sigma_T(t,i) \vDash \Box\,\varphi \quad \text{iff} \quad \begin{array}{l} \forall j \geq i \;.\; \sigma_T(t,j) \vDash \varphi \text{ and} \\ \forall \delta > t \;.\; \forall j \geq 0 \;.\sigma(t,j) \vDash \varphi \end{array}$$

$$\sigma_T(t,i) \vDash \Diamond\,\varphi \quad \text{iff} \quad \begin{array}{l} \exists j \geq i \;.\; \sigma_T(t,j) \vDash \varphi \text{ or} \\ \exists \delta > t \;.\; \exists j \geq 0 \;.\sigma(t,j) \vDash \varphi \end{array}$$

$$\sigma_T(t,i) \vDash \varphi\mathcal{U}\psi \quad \text{iff} \quad \begin{array}{l} \exists j \geq i \;.\; \sigma_T(t,j) \vDash \psi \text{ and} \\ \forall k \;.\; i \leq k < j \;.\; \sigma_T(t,k) \vDash \varphi \\ \textit{or} \\ \exists \delta > t \;.\; \exists j \geq 0 \;.\; \sigma_T(\delta,j) \vDash \psi \text{ and} \\ \forall k \geq i \;.\; \sigma_T(t,k) \vDash \varphi \text{ and} \\ \forall \zeta \;.\; t < \zeta\delta \;.\; \forall k \geq 0 \;.\; \sigma_T(\zeta,k) \vDash \varphi \text{ and} \\ \forall k \;.\; 0 \leq k < j \;.\; \sigma_T(\delta,k) \vDash \varphi \end{array}$$

$$\sigma_T(t,i) \vDash \varphi\mathcal{W}\psi \quad \text{iff} \quad \sigma_T(t,i) \vDash \varphi\mathcal{U}\psi \text{ or } \sigma_T(t,i) \vDash \Box\,\varphi$$

4 Verification

Does a real-time/hybrid system satisfy its temporal specification?

$$\Phi \models \varphi$$

Semantics:

$$\mathcal{L}_B(\Phi) \subseteq \mathcal{L}_B(\varphi) \qquad (\mathcal{R}^{\geq 0}, \mathcal{N}) - models$$

where $\mathcal{L}_B(\Phi)$ is the set of continuous behaviors of the system Φ and $\mathcal{L}_B(\varphi)$ is the set of continuous behaviors that satisfy the formula φ in the continuous semantics.

We would like to apply verification methods and tools for discrete systems, that is, methods that establish

$$\Phi \models \varphi$$

in the discrete semantics:

$$\mathcal{L}(\Phi) \subseteq \mathcal{L}(\varphi) \qquad \mathcal{N} - models$$

Let Φ be a real-time or hybrid system.
For "universal" properties:

$$\mathcal{L}_B(\Phi) \subseteq \mathcal{L}_B(\varphi) \quad \text{iff} \quad \mathcal{L}(\Phi) \subseteq \mathcal{L}(\varphi)$$

For "existential" properties:

$$\mathcal{L}_B(\Phi) \subseteq \mathcal{L}_B(\varphi) \quad \text{if} \quad \mathcal{L}(\Phi) \subseteq \mathcal{L}(\varphi)$$

where "universal" properties are considered properties that have only universal quantifiers in their definition of continuous semantics, and "existential" properties refer to all other properties.

Therefore, we can reuse our verification framework for discrete systems to verify real-time and hybrid systems:

- Verification Rules
- Verification Diagrams

Example: Program ACC

Consider the HTS ACC shown before:

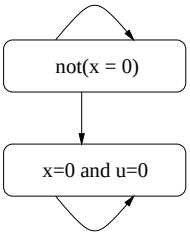

Figure 3: Verification diagram for $\Box(x = 0 \ \rightarrow \ \Box(u = 0 \ \wedge \ x = 0))$

V $\quad \{x, u, v, T\}$ with $D = \{x\}$, $C = \{T\}$, $I = \{u, v\}$

Θ $\quad x = 1 \ \wedge \ u = 0 \ \wedge \ v = 0$

$\mathcal{T}$ $\quad \{\tau_1, \tau_2, \tau_I\}$ with

$\quad \rho_{\tau_1} : \ u \geq 0 \ \wedge \ x' = -u$

$\quad \rho_{\tau_2} : \ u < 0 \ \wedge \ x' = 1 \ \wedge \ v' = 0$

$\mathcal{A}$ $\quad \{\alpha_1, \alpha_2\}$ with

$$\rho_{\alpha_1} : \ x > 0 \ \rightarrow \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ x \end{pmatrix}$$

$$\rho_{\alpha_2} : \ x \leq 0 \ \rightarrow \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} x \\ 0 \end{pmatrix}$$

Π $\quad u \leq 2 \ \wedge \ u \geq -2$

We want to prove that ACC satisfies

$$\Box(x = 0 \ \rightarrow \ \Box(u = 0 \ \wedge \ x = 0)$$

or in words, if x ever becomes zero, it will stay zero forever, that is, it gets trapped.

Figure 3 shows a verification diagram for this property: it represents the first-order proof obligations for proving this property.

Verification conditions:
From the diagram we can extract that we have to show that $x = 0 \wedge u = 0$ is preserved by all transitions:

$$\{x = 0 \ \wedge \ u = 0\} \ \mathcal{T} \ \{x = 0 \ \wedge \ u = 0\}$$

or

$$\tau_1 : \ x = 0 \ \wedge \ u = 0 \ \wedge \ \underbrace{u' = u \ \wedge \ u \geq 0 \ \wedge \ x' = -u}_{\rho_{\tau_1}} \ \rightarrow \ x' = 0 \ \wedge \ u' = 0$$

$$\tau_2 : \ x = 0 \ \wedge \ u = 0 \ \wedge \ \underbrace{u' = u \ \wedge \ u < 0 \ \wedge \ x' = 1}_{\rho_{\tau_2}} \ \rightarrow \ x' = 0 \ \wedge \ u' = 0$$

$$\tau_{\alpha_1} : \forall \Delta \; (x = 0 \; \wedge \; u = 0 \; \wedge \; \underbrace{(\ldots \wedge x > 0 \ldots)}_{\rho_{\tau_{\alpha_1}}} \; \rightarrow \; x' = 0 \; \wedge \; u' = 0)$$

$$\tau_{\alpha_2} : \forall \Delta \; (x = 0 \; \wedge \; u = 0 \; \wedge \; \underbrace{\left(\begin{array}{c} \Delta > 0 \; \wedge \; x' = x \\ \wedge \\ u' = u + x.t \; \wedge \; v' = v + 0.t \\ \wedge \\ \forall t \in [0 \ldots \Delta] \; . \; \left(\begin{array}{c} u + xt \leq 2 \\ \wedge \\ u + xt \geq -2 \end{array} \right) \end{array} \right)}_{\rho_{\tau_{\alpha_2}}} \rightarrow x' = 0 \wedge u' = 0)$$

It is easy to see that all verification conditions are valid, which completes the proof.

Where did we pay the price?

Where did we pay the price for the increased complexity of verifying hybrid systems?

1. Introduction of quantifiers in verification condition

   - Universal force quantifier:

     $$\forall \Delta \; . \; (\text{precond} \; \wedge \; \rho_{tick}[\Delta] \; \rightarrow \; \text{postcond}')$$

   - Existential force quantifier:

     $$\left(\begin{array}{c} \ldots \\ \wedge \\ \forall \delta \in [0 \ldots \Delta] \; . \; \Pi(D, C + \delta, I + \int_0^\delta F^\alpha dt) \end{array} \right)$$

   - Differential inclusions: one existential quantifier per variable.

   Thus we need more sophisticated automatic theorem proving, or interactive theorem proving to check validity of the verification conditions.

2. Increased complexity of ground verification conditions: nonlinear arithmetic with functions such as $e^x, \lg x, \sin x$ etc.

3. We assume we have been given the integral of the functions F^α.

4. Increased complexity of invariant generation.

5. Well-behavedness of systems

5 Well-behaved Systems

A fair/real-time/hybrid system is *well-behaved* if

1. Its set of computations is nonempty, and
2. Every run prefix of the system can be extended to a computation.

WHY?

1. if $\mathcal{L}(S) = \emptyset$ then $S \models \varphi$ for any φ, since $S \models \varphi$ iff $\mathcal{L}(S) \subseteq \mathcal{L}(\varphi)$.
2. parts of the state space are reachable, but do not appear in any computation.

$\Rightarrow$ Verification results for an ill-behaved system can be misleading!

$\Rightarrow$ An ill-behaved system is usually indicative of a modeling error.

Example: Achilles and the Tortoise

The Greek philosopher Zeno (495-430 BC) "proved" that a tortoise, given some headstart, cannot be overtaken by the fastest runner (Achilles). We can model his argument by the following transition system. The positions of Achilles and the tortoise are represented by the variables a and s, respectively. There is one transition, advancing Achilles to the (old) position of the tortoise, and advancing the tortoise a distance consistent with the time taken by Achilles.

V	$\{a, s\}$
Θ	$a = 0 \ \wedge \ s = 50$
$\mathcal{T}$	$\{\tau, \tau_I\}$ with
	$\rho_\tau : \ a' = s \ \wedge \ s' = s + \gamma(s - a)$
J	$\{\tau\}$

where γ is a constant less than 1.

Φ is a perfectly well behaved fair transition system. It has one computation (assume $\gamma = 0.01$):

$$\langle 0, 50\rangle, \ \langle 50, 50.5\rangle \ \langle 50.5, 50.505\rangle, \ \ldots$$

and it is easy to prove that it satisfies:

$$\varphi : \ \Box(a \ < \ s)$$

that is, Achilles will never overtake the tortoise. Of course, fair transition systems do not pretend to model real time.

We can also model it is as a CTS (which do pretend to model real time):

V	$\{a, s, T\}$
Θ	$a = 0 \ \wedge \ s = 50 \ \wedge \ T = 0$
$\mathcal{T}$	$\{\tau, \tau_I\}$ with
	$\rho_\tau : \ a' = s \ \wedge \ s' = s + \gamma(s - a) \ \wedge \ a \geq T$
J	$\{\tau\}$
Π	$T \leq a$

Here we allow Achilles to take a step, and then let time catch up:

$$\rho_{tick}[\Delta] : \left(\begin{array}{c} \Delta > 0 \ \wedge \ a' = a \ \wedge \ s' = s \ \wedge \ T' = T + \Delta \\ \wedge \ \forall t \in [0 \ldots \Delta] \ . \ T + t \leq a \end{array} \right)$$

A run of this system is

$$\langle 0, 50, 0 \rangle \ \xrightarrow{\tau} \ \langle 50, 50.5, 0 \rangle \ \xrightarrow{tick[3]} \ \langle 50, 50.5, 3 \rangle \ \xrightarrow{tick[47]} \ \langle 50, 50.5, 50 \rangle \ \xrightarrow{\tau} \ \langle 50.5, 50.505, 50 \rangle \ \ldots$$

And again it is easy to prove that this system satisfies:

$$\varphi : \ \Box(a \ < \ s)$$

But now because

$$\mathcal{L}_B(S) \ = \ \emptyset$$

In honor of Zeno, systems that have runs that cannot be extended to (infinite) computations are called *Zeno systems*.

Example: Water tanks

Figure 4 shows another example of a Zeno system: it consists of a plant comprising three tanks. The two top tanks alternate in providing water for the lower tank. The objective is to avoid to have any of the tanks run dry. A control system measuring the level in the two top tanks and controlling (open-closed) the two valves has been installed to achieve this objective.

It is easy to see that the two tanks must eventually run dry, since more water is going out than is coming in. However, for the system description below, it is also easy to prove that the two top tanks never will run dry, that is, the proposed controller achieves its objective.

From figure 5 it is clear that the objective is achieved only by eventually switching infinitely often in a finite time, which in practice is not possible. In the modeled behavior time stops just before the condition to be achieved is violated. Again the property holds because the system has no behaviors.

Well-behaved Systems

Fair transition systems can be "made" well-behaved by

- ensuring that Θ is satisfiable, and
- adding an idling transition τ_I with transition relation $\rho_{\tau_I} : \ V' = V$.

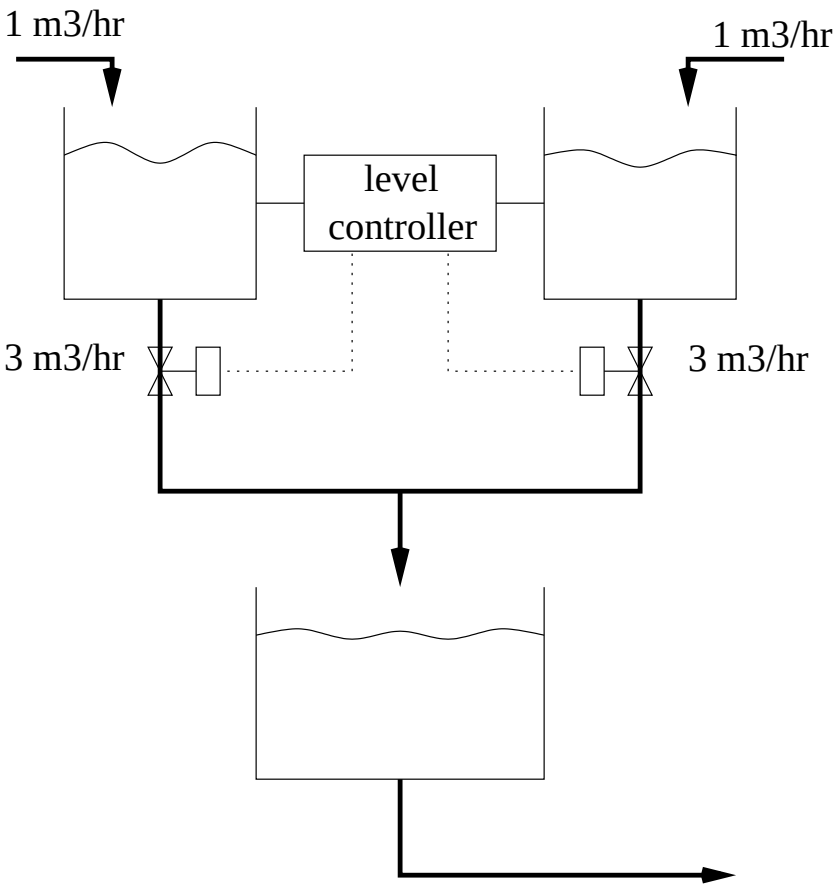

Figure 4: Three water tanks system

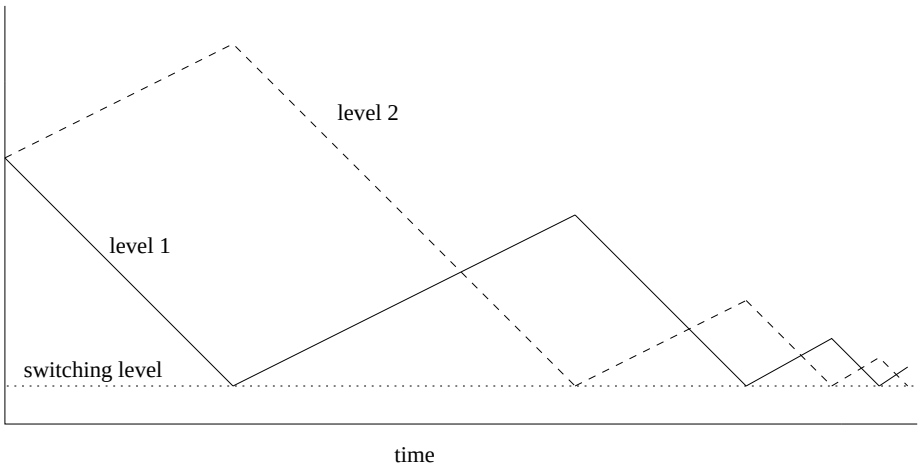

Figure 5: Behavior of three water tanks system

In particular:

- Adding fairness constraints (in the form of just and compassionate transitions) cannot make a well-behaved system ill-behaved.

However .....

This is not sufficient for real-time and hybrid system, because of the additional condition:

- **Time divergence:** T (the master clock) grows beyond any bound

$\Rightarrow$ We have to show that the tick transition is enabled sufficiently often for sufficiently large values of Δ.

Proving NonZenoness

A system is nonZeno if every run prefix can be extended to an infinite computation:

in CTL:

$$\forall \epsilon > 0. \forall t. A \,\Box (T = t \;\rightarrow\; E \,\Diamond (T \geq t + \epsilon))$$

which is equivalent to

$$\exists \epsilon > 0. \forall t. A \,\Box (T = t \;\rightarrow\; E \,\Diamond (T \geq t + \epsilon))$$

6 Invariant Generation

Forward propagation from false

Forward propagation from the initial states until convergence.

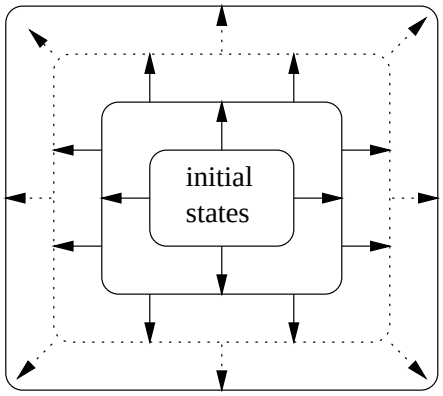

Figure 6: Forward propagation from the initial states

Define the forward propagator (predicate transformer):

$$\mathcal{F}(X) : \Theta \ \vee \ Post(X)$$

where X denotes a set of states, and $Post(X)$ the disjunction of the postconditions of all transitions starting from a state in X:

$$Post(X) = \bigvee_{\tau \in \mathcal{T}} post(\tau, X)$$

with $post(\tau, X)$ the postcondition of τ

$$post(\tau, X) \ = \ \exists V^0 \ . \ X(V^0) \wedge \ \rho_\tau(V^0, V)$$

To compute the reachable states start with the empty set of states (denoted by *false*) and iterate until convergence, that is, $\mathcal{F}(X) \subseteq X$:

$false$	Θ	$\Theta \vee Post(\Theta)$	$\Theta \vee Post(\Theta \vee Post(\Theta))$	$\ldots$
$\emptyset$	initial states	0 or 1 step	0 or 1 or 2 steps	

Problems:

- Quantifier elimination
- Obtaining/detecting convergence

Solutions:

- Perform propagation in abstract finite domain (quantifier elimination and detecting convergence are decidable, and obtaining convergence is guaranteed).
- Perform propagation in abstract domain that allows quantifier elimination (and perform widening to obtain convergence).

Forward propagation from true

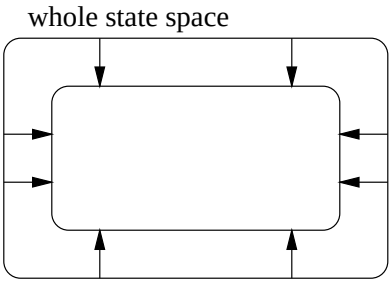

Figure 7: Forward propagation starting from the whole state space

When propagating starting from *true* the result of every propagation step is an invariant (*true* is an invariant of every system):

$$true \quad Post(true) \vee \Theta \quad Post(Post(true) \vee \Theta) \vee \Theta \quad \ldots$$

For real-time and hybrid systems we can separate the propagation of the discrete and time-step transitions:

$$\Theta \vee Post(true)$$

can be written as

$$\Theta \quad \vee \quad Post_D \quad \vee \quad Post_A$$

disjunction of all discrete transitions

disjunction of all time-step transitions

$$\bigvee_{\alpha \in \mathcal{A}} post(\tau_\alpha, true)$$

Invariant Generation: Simplification

The postcondition of a time-step transition can be simplified to $\Pi \wedge p_\alpha$ (or Π in case of a CTS).

$$
\begin{aligned}
post(\tau_\alpha, true) &= \exists \Delta. D^0. C^0. I^0. \rho_{\tau_\alpha}[\Delta](D^0, C^0, I^0) \\
&= \exists \Delta. D^0. C^0. I^0. \left(\begin{array}{c} \Delta > 0 \ \wedge \ \underbrace{D = D^0} \ \wedge \ \underbrace{C = C^0 + \Delta} \ \wedge p_\alpha(D^0) \\ \wedge \\ \underbrace{I = I^0 + \int_0^\Delta F^\alpha dt} \\ \wedge \\ \forall \delta \in [0 \ldots \Delta] \ . \ \Pi(D^0, C^0 + t, I^0 + \int_0^\delta F^\alpha dt) \end{array} \right)
\end{aligned}
$$

Instantiate:

$$
\begin{array}{lll}
D^0 & \text{with} & D \\
C^0 & \text{with} & C - \Delta \\
I^0 & \text{with} & I - \int_0^\Delta F^\alpha dt
\end{array}
$$

to obtain

$$
\exists \Delta. \left(\begin{array}{c} \Delta > 0 \ \wedge \ p_\alpha(D) \\ \wedge \\ \forall \delta \in [0 \ldots \Delta]. \Pi(D, C + t - \Delta, I + \int_0^\delta F^\alpha dt - \int_0^\Delta F^\alpha dt) \end{array} \right)
$$

Take $t = \Delta$ (over approximation) to obtain

$$
\Pi(D, C, I) \ \wedge \ p_\alpha
$$

and thus

$$
post(\tau_\alpha, true) \ \rightarrow \ \Pi \ \wedge \ p_\alpha
$$

and therefore

$$
Post_A(true) \ = \bigvee_{\alpha \in \mathcal{A}} post(\tau_\alpha, true) \quad \rightarrow \quad \Pi
$$

by the exhaustiveness of the p_α's

1-step Forward propagation:

$$
Inv_1 : \ \ \Theta \ \vee \ Post_D(true) \ \vee \ \Pi
$$

2-step Forward propataion:

$$
Inv_2 : \ \ \Theta \ \vee \ Post_D(true) \ \vee \ Post_A(Post_D(true) \ \vee \ \Theta)
$$

These two invariants (which are not too hard to compute) are useful in many applications.

7 Summary

- Reduction from verification of real-time and hybrid systems to verification of discrete systems.
- Part of the complexity has been relegated to
  - theorem prover (quantifier elimination)
  - decision procedures
  - real analysis
- We still have to deal with
  - proving that the system is well-behaved (nonZeno)
  - invariant generation