Robustness of Temporal Logic Specifications for Systems

Georgios Fainekos dissertation series - Part II

Akshay Rajhans

ECE Department, CMU

SVC Seminar: Sep 12, 2008

Outline

1. Preliminaries and Problem Formulation
2. Why we could get stuck
3. A short detour: Approximate bisimulation relations and bisimulation functions
4. How bisimulation function comes to our rescue here
5. Verification algorithm
6. Other contributions and related work

Outline

Continuous Time Dynamical System

Definition

A continuous time dynamical system is a tuple $\Sigma = (N, P, f, g, I, AP, \mathcal{O})$, where:

- N and P are dimensions of state space and observation space
- $f : \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}^P$ are continuous maps
- I is a compact subset of $\mathbb{R}^N$ that is the set of initial states
- AP is the set of atomic propositions
- $\mathcal{O} : AP \to \mathcal{P}(\mathbb{R}^P)$ is a predicate mapping

Continuous Time Dynamical System

Definition

A continuous time dynamical system is a tuple $\Sigma = (N, P, f, g, I, AP, \mathcal{O})$, where:

- N and P are dimensions of state space and observation space
- $f : \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}^P$ are continuous maps
- I is a compact subset of $\mathbb{R}^N$ that is the set of initial states
- AP is the set of atomic propositions
- $\mathcal{O} : AP \to \mathcal{P}(\mathbb{R}^P)$ is a predicate mapping

Trajectory

A trajectory is a pair of functions $(x(t), y(t))$ such that $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ and $y : \mathbb{R}_{\geq 0} \to \mathbb{R}^P$ satisfy:

- $x(0) \in I$ and
- $\forall t \in \mathbb{R}_{\geq 0}$
  - $\dot{x}(t) = f(x(t))$ and
  - $y(t) = g(x(t))$

(Finite) Timed State Sequence

Definition: (Finite) TSS

A timed state sequence $\mathcal{T}$ in a space Q is a tuple $(\sigma, \tau, \mathcal{O})$, where, for some $n \in \mathbb{N}$

- $\sigma = \sigma_0, \sigma_1, \cdots, \sigma_n$ is a sequence of states
- $\tau = \tau_0, \tau_1, \cdots, \tau_n$ is a sequence of time stamps
- $\mathcal{O} : AP \to \mathcal{P}(Q)$ is a predicate mapping

such that, the following conditions are satisfied:

- $\forall i \in \{0, 1, \cdots, n\}$ we have $\sigma_i \in Q$ and $\tau_i \in \mathcal{R}_{\geq 0}$
- τ is a strictly monotonically increasing sequence

Example: Numerically simulated trajectory by Matlab's ODE integrator

(Finite) Timed State Sequence

Definition: (Finite) TSS

A timed state sequence $\mathcal{T}$ in a space Q is a tuple $(\sigma, \tau, \mathcal{O})$, where, for some $n \in \mathbb{N}$

- $\sigma = \sigma_0, \sigma_1, \cdots, \sigma_n$ is a sequence of states
- $\tau = \tau_0, \tau_1, \cdots, \tau_n$ is a sequence of time stamps
- $\mathcal{O} : AP \rightarrow \mathcal{P}(Q)$ is a predicate mapping

such that, the following conditions are satisfied:

- $\forall i \in \{0, 1, \cdots, n\}$ we have $\sigma_i \in Q$ and $\tau_i \in \mathcal{R}_{\geq 0}$
- τ is a strictly monotonically increasing sequence

Example: Numerically simulated trajectory by Matlab's ODE integrator

Suffix operator

- $\sigma \uparrow_i = \sigma_i, \sigma_{i+1}, \sigma_{i+2}, \cdots, \sigma_n$
- $\tau \uparrow_i = 0, \tau_{i+1} - \tau_i, \tau_{i+2} - \tau_i, \cdots, \tau_n - \tau_i$
- $\mathcal{T} \uparrow_i = (\sigma \uparrow_i, \tau \uparrow_i, \mathcal{O})$

Notation

Set of all possible finite TSS

Denoted by TS

Set of all possible TS given $\mathcal{T}$

Denoted by $TS_{\mathcal{T}}$.
The only requirement is that they have the same time stamps as $\mathcal{T}$.

Notation

Set of all possible finite TSS

Denoted by TS

Set of all possible TS given $\mathcal{T}$

Denoted by $TS_{\mathcal{T}}$.
The only requirement is that they have the same time stamps as $\mathcal{T}$.

Trace

Informally, 'trace' = sampled form of 'trajectory'
Formally: Given a sequence of time stamps τ of length $|\tau|$, trace of a dynamical system Σ is a TSS $\mathcal{T} = (\sigma, \tau, \mathcal{O})$ such that $\exists$ a trajectory (x, y) of Σ satisfying $\sigma_i = y(\tau_i) = g(x(\tau_i))$ for every $i = 0, 1, \cdots, |\tau| - 1$.

Set of all possible traces given τ

Denoted by $\mathcal{L}_\tau(\Sigma)$

(Boolean) MTL semantics for systems

Inductive Grammar

Given $\pi \in AP$, $\mathcal{I} \subset \mathbb{R}_{\geq 0}$ with rational endpoints, an MTL formula ϕ is defined according to the inductive grammar:

$$\phi \ ::= \ \top \mid \pi \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \phi_1 \, \mathcal{U}_{\mathcal{I}} \phi_2$$

Semantics

Let $\mathcal{T} = (\sigma, \tau, \mathcal{O}) \in TS$, $\pi \in AP$, $i, j, \in \mathbb{N}$ and ψ, ϕ_1, ϕ_2 be well formed MTL formulas. Then the semantics can be recursively defined as:

$$\begin{aligned}
\langle\langle \top \rangle\rangle(\mathcal{T}) &:= \top \\
\langle\langle \pi \rangle\rangle(\mathcal{T}) &:= \sigma_0 \in \mathcal{O}(\pi) \\
\langle\langle \neg\psi \rangle\rangle(\mathcal{T}) &:= \neg\langle\langle \psi \rangle\rangle(\mathcal{T}) \\
\langle\langle \phi_1 \vee \phi_2 \rangle\rangle(\mathcal{T}) &:= \langle\langle \phi_1 \rangle\rangle(\mathcal{T}) \vee \langle\langle \phi_2 \rangle\rangle(\mathcal{T}) \\
\langle\langle \phi_1 \, \mathcal{U}_{\mathcal{I}} \phi_2 \rangle\rangle(\mathcal{T}) &:= \bigvee\nolimits_{i=0}^{|\mathcal{T}|-1} \left((\tau_i \in \mathcal{I}) \wedge \langle\langle \phi_2 \rangle\rangle(\mathcal{T}\uparrow_i) \wedge \bigwedge\nolimits_{j=0}^{i-1} \langle\langle \phi_1 \rangle\rangle(\mathcal{T}\uparrow_j) \right)
\end{aligned}$$

Problem statement

Statement

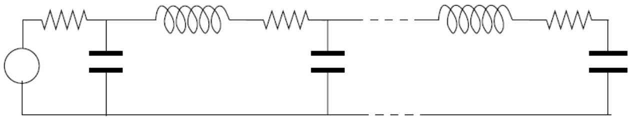
Given ϕ, Σ and τ, verify that $\mathcal{L}_\tau(\Sigma) \subseteq \mathcal{L}(\phi)$, where $\mathcal{L}(\phi)$ is the set of all models of ϕ i.e. $\mathcal{L}(\phi) = \{\mathcal{T} \in TS | \ll \phi \gg (\mathcal{T}) = \mathrm{T}\}$
Or, in other words, verify that $\mathcal{L}_\tau(\Sigma) \cap \mathcal{L}(\neg\phi) = \emptyset$

Problem statement

Statement

Given ϕ, Σ and τ, verify that $\mathcal{L}_\tau(\Sigma) \subseteq \mathcal{L}(\phi)$, where $\mathcal{L}(\phi)$ is the set of all models of ϕ i.e. $\mathcal{L}(\phi) = \{\mathcal{T} \in TS | \ll \phi \gg (\mathcal{T}) = \mathrm{T}\}$
Or, in other words, verify that $\mathcal{L}_\tau(\Sigma) \cap \mathcal{L}(\neg\phi) = \emptyset$

Example: TL verification of a transmission line

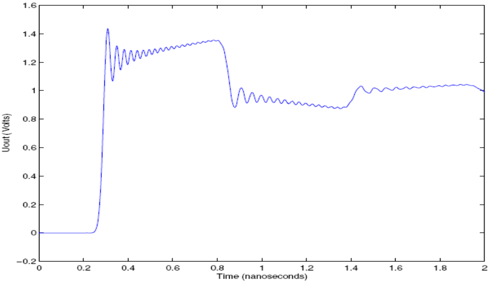


Dynamics of the line given by:

$$\dot{x}(t) = Ax(t) + bU_{in}(t) \text{ and } U_{out}(t) = Cx(t)$$

Example continued...

Given that $U_{in}(0) \in [-0.2, 0.2]$, a sample trace could be



We want to verify that $U_{out}(t)$ stabilizes between 0.8V to 1.2V within T nS, and that overshoot is bounded by θ V. Here, $T \in [0, 2]$ and $\theta \geq 0$ are design parameters. This specification can be written in MTL as $\phi = \Box\pi_1 \wedge \Diamond_{[0,2]}\Box\pi_2$, where $\mathcal{O}(\pi_1) = [-\theta, \theta]$ and $\mathcal{O}(\pi_2) = [0.8, 1.2]$.

Robust satisfaction

Boolean result is not robust. We need something beyond result $\in \{\text{true,false}\}$.

Robust satisfaction

Boolean result is not robust. We need something beyond result $\in$ {true,false}.
So we define

Robustness degree

$\varepsilon := \mathbf{Dist}_\rho(\sigma, P_{\mathcal{T}}^{\phi})$, i.e. signed distance of σ from $P_{\mathcal{T}}^{\phi}$

where,

Set of all TSS that satisfy ϕ

$$P_{\mathcal{T}}^{\phi} = \{\sigma' | (\sigma', \tau, \mathcal{O}) \in TS_{\mathcal{T}} \cap \mathcal{L}(\phi)\}$$

Metric

Let $d(y_1, y_2) = \sqrt{(y_1 - y_2)^T (y_1 - y_2)}$.
Then $\rho(\sigma, \sigma') = max\{d(\sigma_0, \sigma'_0), d(\sigma_1, \sigma'_1), \cdots, d(\sigma_{|\tau|-1}, \sigma'_{|\tau|-1})\}$ is a well defined metric on $(\mathbb{R}^P)^{|\mathcal{T}|}$.

Robustness degree

Computation of $P_{\mathcal{T}}^{\phi}$ is expensive. So we don't do that.
Instead, we define multi-valued a.k.a. robust semantics.

Robust semantics

Let $\mathcal{T} = (\sigma, \tau, \mathcal{O}) \in TS$, $\pi \in AP$, $i, j, \in \mathbb{N}$ and ψ, ϕ_1, ϕ_2 be well formed MTL formulas. Then the robust semantics can be recursively defined as:

$$
\begin{aligned}
[\![\top]\!](\mathcal{T}) &:= +\infty \\
[\![\pi]\!](\mathcal{T}) &:= \mathbf{Dist}_d(\sigma_0, \mathcal{O}(\pi)) \\
[\![\neg\psi]\!](\mathcal{T}) &:= -[\![\psi]\!](\mathcal{T}) \\
[\![\phi_1 \vee \phi_2]\!](\mathcal{T}) &:= [\![\phi_1]\!](\mathcal{T}) \sqcup [\![\phi_2]\!](\mathcal{T}) \\
[\![\phi_1\, \mathcal{U}_{\mathcal{I}} \phi_2]\!](\mathcal{T}) &:= \bigsqcup_{i=0}^{|\mathcal{T}|-1} \left([\![\tau_i \in \mathcal{I}]\!](\mathcal{T}) \sqcap [\![\phi_2]\!](\mathcal{T}\uparrow_i) \sqcap \bigsqcap_{j=0}^{i-1} [\![\phi_1]\!](\mathcal{T}\uparrow_j) \right)
\end{aligned}
$$

Remember that: By construction, robust TL semantics give a conservative approximation of the robustness degree.

Outline

Why we might get stuck

In order to proceed with the verification

We will need to check the robust valuation of the given specification for every trace.

- If $[[\phi]](\mathcal{T}_i) > 0 \ \forall \mathcal{T}_i$, then the system satisfies the formula
- If $\exists \mathcal{T}_j$ such that $[[\phi]](\mathcal{T}_i) < 0$, then the system does not satisfy the formula as $\exists$ a counterexample

But...

In case of continuous systems, $\exists$ infinite traces that can arise out of a finite non-empty non-singleton set of initial conditions.

Outline

Definitions

Labeled transition system

A labeled transition system with observations is a tuple $T = (\mathcal{Q}, \Sigma, \rightarrow, \mathcal{Q}^0, \Pi, \ll \cdot \gg)$ that consists of

- a (possibly infinite) set $\mathcal{Q}$ of states
- a (possibly infinite) set Σ of labels
- a transition relation $\rightarrow \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q}$
- a (possibly infinite) set $\mathcal{Q}^0 \subseteq \mathcal{Q}$ of initial states
- a (possibly infinite) set Π of of observations
- an observation map $\ll \cdot \gg : \mathcal{Q} \rightarrow \Pi$

σ-successor

A set valued map given $\forall q \in \mathcal{Q}$ by: $Post^\sigma(q) = \{q' \in \mathcal{Q} | q \xrightarrow{\sigma} q'\}$

Trajectory $q^0 \xrightarrow{\sigma^0} q^1 \xrightarrow{\sigma^1} q^2 \xrightarrow{\sigma^2} \ldots$, where $q^0 \in \mathcal{Q}^0$

Approximate bisimulation relations

Given labeled transition systems
$T_1 = (\mathcal{Q}_1, \Sigma, \rightarrow_1, \mathcal{Q}_1^0, \Pi, \ll \cdot \gg)$ and $T_2 = (\mathcal{Q}_2, \Sigma, \rightarrow_2, \mathcal{Q}_2^0, \Pi, \ll \cdot \gg)$,
assuming $\mathcal{Q}_1$, $\mathcal{Q}_2$ and Π are metric spaces,
assuming $\mathcal{Q}_1^0$ and $\mathcal{Q}_2^0$ and $Post_1^\sigma(q_1)$ and $Post_2^\sigma(q_2)$ are compact sets,

Definition: δ-approximate Bisimulation relation

A relation $\mathcal{B}_\delta \subseteq \mathcal{Q}_1 \times \mathcal{Q}_2$ is a δ-approximate bisimulation relation between T_1 and T_2 if $\forall (q_1, q_2) \in \mathcal{B}_\delta$

- $d_\Pi(\ll q_1 \gg_1, \ll q_2 \gg_2) \leq \delta$
- $\forall q_1 \xrightarrow{\sigma}_1 q_1', \exists q_2 \xrightarrow{\sigma}_2 q_2'$ such that $(q_1', q_2') \in \mathcal{B}_\delta$
- $\forall q_2 \xrightarrow{\sigma}_2 q_2', \exists q_1 \xrightarrow{\sigma}_1 q_1'$ such that $(q_1', q_2') \in \mathcal{B}_\delta$

T_1 and T_2 are said to be approximately bisimilar with precision δ, if there exists such $\mathcal{B}_\delta$, written as $T_1 \sim_\delta T_2$.

Note: When δ=0, we have the usual notion of exact bisimulation.

Bisimulation functions

Definition: Bisimulation function

A continuous function $V_B : \mathcal{Q}_1 \times \mathcal{Q}_2 \rightarrow \mathbb{R}_{\geq 0}$ is a bisimulation function for the dynamical system Σ if its level sets are closed sets and $\forall (q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$, we have:

- $V_B(q_1, q_2) \geq d_\Pi(\ll q_1 \gg_1, \ll q_2 \gg_2)$
- $V_B(q_1, q_2) \geq \max_{q_2 \xrightarrow{\sigma}_2 q_2'} \min_{q_1 \xrightarrow{\sigma}_1 q_1'} V_B(q_1', q_2')$
- $V_B(q_1, q_2) \geq \max_{q_1 \xrightarrow{\sigma}_1 q_1'} \min_{q_2 \xrightarrow{\sigma}_2 q_2'} V_B(q_1', q_2')$

Theorem

If V_B is a bisimulation function, then $\forall \delta \geq 0$, the set $B_\delta = \{(q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2 | V_B(q_1, q_2) \leq \delta\}$ is a δ-approximate bisimulation relation between T_1 and T_2.

Analogous concepts for continuous systems

Pappas

"Bisimilar Linear Systems" - Automatica, '03

- Explains how the notion of exact bisimulations can be developed for continuous systems

Girard and Pappas

Some good papers that talk about approximate bisimulations

- "Approximate Bisimulations for Constrained Linear Systems" - CDC '05
- "Approximate Bisimulations for Nonlinear Dynamical Systems" - CDC '05
- "Approximation Metrics for Discrete and Continuous Systems" - IEEE Transactions on Automatic Control, '07

Constructing bisimulation functions

For linear systems

For autonomous linear systems $\dot{x} = Ax$, $y = Cx$, we can get a valid bisimulation function of the type $V(x_1, x_2) = \sqrt{(x_1 - x_2)^T M(x_1 - x_2)}$ if we can get M that satisfies

- $M \geq C^T C$ and
- $A^T M + MA \leq 0$

Note: Here we need to solve a semidefinite program.

For nonlinear systems

For autonomous nonlinear systems $\dot{x} = f(x)$, $y = g(x)$, there exists a valid bisimulation function of the form $V(x_1, x_2) = \sqrt{q(x_1, x_2)}$ if

- $q(x_1, x_2) - ||g_1(x_1) - g_2(x_2)||^2$ is sum of squares, and
- $-\frac{\partial q(x_1, x_2)}{\partial x_1} f_1(x_1) - \frac{\partial q(x_1, x_2)}{\partial x_2} f_2(x_2)$ is sum of squares.

Note: Here we need to solve a sum of sqares program.

What we need to remember

Take-away messages

- For stable systems, bisimulation functions exist and can be computed (possibly with some effort).
- The level sets of bisimulation functions are invariant sets.
- If two trajectories start with some value of bisimulation function, it cannot increase as time increases.
- Which means, the distance between any two trajectories cannot increase as time increases.

Outline

How does a bisimulation function apply here?

A smart idea

Consider an approximate bisimulation relation between a system and itself. Then any trajectory that starts with the value of the bisimulation function $=$ r from another trajectory, remains within this 'robustness tube' of radius r.

How does a bisimulation function apply here?

A smart idea

Consider an approximate bisimulation relation between a system and itself. Then any trajectory that starts with the value of the bisimulation function $=$ r from another trajectory, remains within this 'robustness tube' of radius r.

The connecting piece

Theorem: Given V a bisimulation function, (x_1, y_1) and (x_2, y_2) two trajectories whose traces are $\mathcal{T}_1$ and $\mathcal{T}_2$,
if $\exists i \in \{1, 2\}$ such that $|[[\phi]](\mathcal{T}_i)| > V(x_1(0), x_2(0))$
then that guarantees that $\ll \phi \gg (\mathcal{T}_1) = \ll \phi \gg (\mathcal{T}_2)$

How does a bisimulation function apply here?

A smart idea

Consider an approximate bisimulation relation between a system and itself. Then any trajectory that starts with the value of the bisimulation function = r from another trajectory, remains within this 'robustness tube' of radius r.

The connecting piece

Theorem: Given V a bisimulation function, (x_1, y_1) and (x_2, y_2) two trajectories whose traces are $\mathcal{T}_1$ and $\mathcal{T}_2$,
if $\exists i \in \{1, 2\}$ such that $|[[\phi]](\mathcal{T}_i)| > V(x_1(0), x_2(0))$
then that guarantees that $\ll \phi \gg (\mathcal{T}_1) = \ll \phi \gg (\mathcal{T}_2)$

Which means that we can...

- reason about a neighborhood of initial conditions by reasoning about only one (central) initial condition
- reason about an entire initial condition set by reasoning about finitely many initial conditions

Outline

1. Preliminaries and Problem Formulation
2. Why we could get stuck
3. A short detour: Approximate bisimulation relations and bisimulation functions
4. How bisimulation function comes to our rescue here
5. Verification algorithm
6. Other contributions and related work

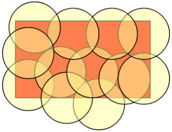
Setting up the algo. - sampling the initial conditions set

Theorem: In words

Let V be a bisimulation function, $I \subset \mathbb{R}^N$ a compact set of initial conditions. Then, $\forall \delta > 0$, $\exists$ a finite set of points $\{x_1, x_2, \cdots, x_r\} \subset I$ such that,

$$\forall x \in I, \exists x_i, i \in \{1, 2, \cdots, r\} s.t. V(x, x_i) \le \delta \tag{1}$$

Pictorially:



Centers of these balls will form the set $\{x_1, x_2, \cdots, x_r\}$, while radius will be δ.

Discretization operator

Given $\delta > 0$, $\mathbf{Disc}(I, \delta)$ gives out a set $\{x_1, x_2, \cdots, x_r\}$ that satisfies (1).

Verification algorithm

Start

1. Choose δ, get points $x_i(0)$ by doing $\mathbf{Disc}(I, \delta)$
2. Simulate (x_i, y_i) and get $\mathcal{T}_i$.
3. Find out the value of $[[\phi]](\mathcal{T}_i)$.

Case 1: Satisfaction

If $[[\phi]](\mathcal{T}_i) > \delta$, $\forall i \in \{1, 2, \cdots, r\}$, we have: $\ll \phi \gg = \mathrm{T}\ \forall \mathcal{T} \in \mathcal{L}_\tau(\Sigma)$
Report that formula is satisfied and stop.

Case 2: Counterexample

If $\exists i \in \{1, \cdots, r\}$ such that $[[\phi]](\mathcal{T}_i) < 0$, then that trace is a counterexample.
Report counterexample and stop.

Case 3: Recursive refinement

In case of inconclusive results, i.e. $0 < [[\phi]](\mathcal{T}_i) < \delta$, for some i, refine locally the initial conditions (i.e. reduce δ, get more centers) and re-run the tests.

Possible outcomes of the algorithm

One of the three outcomes possible:

- Formula holds for every trajectory from the entire initial condition set
- Formula holds for a subset of the initial condition set
- Formula does not hold, algorithm returns a counterexample
- Less likely but possible: Result inconclusive, if $[[\phi]] = 0$.

An observation

The more robust the system, the less number of individual tests needed.

Summary

Take-away messages from part-I talk

- Multi-valued TL semantics make the use of TL more robust in testing
- We can analyze CT signals by corresponding DT TSSs after 'strengthening' the specifications

Take-away messages from today's part-II talk

- Multi-valued TL semantics can be extended from signals to systems
- Bounded time verification can be approached using multi-valued TL testing
- The more robust the system, the easier the verification

Outline

Other contributions of the thesis and related work

Robust testing of hybrid systems - An interesting application

"Robust Test Generation and Coverage for Hybrid Systems" - Julius, Fainekos, Anand, Lee, Pappas, HSCC '07

- Being within an observation map of a proposition is analogous to being in a discrete location.
- Unsafe regions are also analogous to observation maps.
- Temporal constraints are induced by the switching times e.g. you have to or cannot switch within a given interval of time.

Other contributions

TL Motion Planning:

- Reachability, collision avoidance, seqencing, coverage and liveness requirements of a motion planning problem can be cast as a TL problem.
- Robustness techniques can be applied to those problems too.
- Controller synthesis: Given an LTL formula ϕ, generate a controller for a system such that it satisfies ϕ

References

Relevant papers

- "Temporal Logic Verification Using Simulation" - Fainekos, Girard, Pappas
- "Approximate Bisimulations for Constrained Linear Systems" - Girard, Pappas
- "Approximate Bisimulations for Nonlinear Dynamical Systems" - Girard, Pappas
- "Approximation Metrics for Discrete and Continuous Systems" - Girard, Pappas
- "Bisimilar Linear Systems" - Pappas

Thank you

Thank you

- Thanks to Ed Clarke for hosting me
- Thanks to Alex Donzé for reviewing the slides
- Thank you all for attending