Formal Design Theories and Tools for Safety-Critical Cyber-Physical Systems

Naijun Zhan

School of Computer Science, Peking University

2024 Summer School on Trustworthy Software

SEI of ECNU, Shanghai, July 9-12

Part 2 : Verification

Hybrid Hoare Logic [APLAS'10, CoRR abs/2303.15020]

Model-Checking vs Theorem-Proving

Model-checking

- CPS are modelled as hybrid automata, and verification is conducted by computing reachable sets.
- Advantages :
  - ⇒ Models are very intuitive
  - ⇒ Automatic
  - ⇒ Counter-examples
- Disadvantages :
  - ⇒ Lackness of compositionality
  - ⇒ High complexity and undecidability
  - ⇒ Bounded time
  - ⇒ Scalability
- **How to compute reachable set is challenging**, many techniques based on abstraction, approximation, numeric computation and so on have been proposed

Theorem-proving

- HS are modelled as process algebras-like formalisms, and verification is conducted by theorem proving based on an underlined **specification logic** together with **invariant generation**
- Advantages :
  - ⇒ Compositionality
  - ⇒ Scalability
  - ⇒ Unbounded verification
- Disadvantages :
  - ⇒ Specification logics are normally complicated, only with weak relative completeness
  - ⇒ Normally interactive, non-automatic, at most semi-automatic
  - ⇒ Low efficiency
  - ⇒ No counter-example
- **How to synthesize more expressive invariants is challenging**, particular, for complicated behavior like DDE, SDE and so on

Related Work on Deductive Verification of HSs

- Differential dynamic logic (dL)
  - Due to Platzer, extension of dynamic logic to hybrid systems modelled by differential programs ([Platzer 08, Platzer 10])
  - Its continuous relative completeness and discrete relative completeness ([Platzer 12])
  - Some variants, e.g., stochastic differential dynamic logic ([Platzer 12]), differential game logic ([Platzer 15]), differential refinement logic ([Loos&Platzer 16]), etc.
  - Tool support, **KeYmaera** ([Platzer&Quesel 08])
  - **Cannot cope with concurrency and communication in an explicit and compositional way**
- Hybrid Event-B
  - Due to Su, Abrial and Zhu, extension of Event-B to hybrid systems ([Su&Abrial&Zhu 14])
  - More concerning how to design a correct hybrid system based on refinement theory similar to action system
  - Tool support, **Roddin**
  - **Weak on verification**
- Others, e.g. **Extended Duration Calculus** (needs oracle for continuous reasoning)

☹ **A powerful specification logic for HSs that can deal with concurrency and communication in a compositional way is desirable!**

A Brief History of Hoare logic

- Hoare Logic for sequential programs
  - **Hoare logic** due to Floyd and Hoare, the cornerstone of **program verification** ([Floyd 67, Hoare 69])
  - Hoare triple : $\{Pre\}\ P\ \{Post\}$
  - The relative completeness wrt the completeness of the assertional logic due to Cook ([Cook 78])
  - Not a complete Hoare proof system for some program constructs ([Clarke 79])
- Extending Hoare Logic to concurrent programs
  - **Hoare logic** for concurrent programs with shared variable, due to Owicki, Gries **et al.** ([Owicki& Gries 76]), **non-interference** for parallel rule, its relative completeness in Cook's sense ([Owicki 76])
  - **Hoare logic** for CSP due to Apt, Francez and de Roever ([Apt& Francez & de Rover 80]), **cooperative** for communication rule, and its relative completeness in Cook's sense ([Apt 83])
  - **Generalized Hoare logic** for concurrent programs with different models due to Lamport ([Lamport 77, Lamport 80, Lamport& Schneider 84]), and its relative completeness (Cousot& Cousot 89])
- Extending Hoare Logic to mutable data structure
  - **Separation logic** due to Reynolds (Reynolds 02])
  - O'Hearn **et al.** developed the logic, even extended it to concurrency ([O'Hearn at al. 07, O'Hearn et al. 09, O'Hearn et al. 11]).

A Brief History of Hoare logic

- Extending Hoare logic to real-time and hybrid systems
  - **Real-time Hoare logic** due to Hooman ([Hooman 94])
  - A DC-based **Hybrid Hoare Logic** (HHL) due to Liu et al. ([Liu et al. 11])
    - A **hybrid Hoare assertion** is of the form $\{Pre\}\ P\ \{Post;\ HF\}$, Pre and $Post$ are FOL formulas, P is an HCSP process, HF is a DC formula, specifying invariants
    - The proof system is not compositional
    - [Wang&Zhan&Guelev 12] proposed an assume-guarantee style compositional proof system, and [Guelev&Wang&Zhan 17] proved its relative completeness wrt DC
    - An **Isabelle/HOL-based theorem prover** ([Wang&Zhan&Zou 15])
  - Recently, we simplified HHL with the following improvements
    - The assertion logic is the first-order theory of differential equations, together with some traces predicates
    - Its continuous relative completeness and discrete relative completeness
    - An implementation based on Isabelle/HOL
- Other important extensions of Hoare logic
  - **A Hoare-like proof system for partial correctness of probabilistic programs** (Hartog& Vink 02])
  - **Relational Hoare logic** (RHL) for comparing two programs ([Benton 04])
  - **Probabilistic RHL** (pRHL) for specifying and reasoning about security ([Barthe et al. 13a]), its proof assistant EasyCrypt ([Barthe et al. 13b])
  - **Quantum Hoare logic** (qHL) ([Ying 11]) and its proof assistant ([Liu et al. 19])
  - **Incorrect logic** ([O'Hear 11]) and **incorrect separation logic** ([Raad et al. 20]) for detecting bugs
- More details about the history of Hoare logic pls refer to [Apt&Olderog 19]

Small and Big-step Semantics

Small-step semantics

- Transitions of the form $(c, s) \xrightarrow{e} (c', s')$, denotes that one step of computation transforms process c and state s into process c' and state s', resulting in event e.
- The transitive closure of small-step semantics is $(c, s) \xrightarrow{tr}{}^{*} (c', s')$, where tr is the trace of events.

Big-step semantics

- Transitions of the form $(c, s) \Rightarrow (s', tr)$, denotes that the process c carries initial state s to final state s' with resulting trace tr.

Theorem (Equivalence Between Two Semantics)

i) *For any big-step relation* $(c, s) \Rightarrow (s', tr)$*, we have the small-step relation* $(c, s) \xrightarrow{tr}{}^{*} (skip, s')$.

ii) *For any small-step relation* $(c, s) \xrightarrow{tr}{}^{*} (skip, s')$*, there exists* tr' *such that* $tr \rightsquigarrow_r tr'$ *and* $(c, s) \Rightarrow (s', tr')$.

Small Step Semantics for HCSP

$$\frac{\begin{array}{c}\mathbf{p} \text{ is a solution of the ODE } \dot{\mathbf{x}} = \mathbf{e} \\ \mathbf{p}(0) = s(\mathbf{x}) \quad \forall t \in [0, d).\ s[\mathbf{x} \mapsto \mathbf{p}(t)](B)\end{array}}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle, s) \xrightarrow{\langle d, \mathbf{p}, \{\} \rangle} (\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle, s[\mathbf{x} \mapsto \mathbf{p}(d)])} \qquad \frac{\neg s(B)}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle, s) \xrightarrow{\tau} (\text{skip}, s)}$$

$$\frac{\begin{array}{c}\mathbf{p} \text{ is a solution of the ODE } \dot{\mathbf{x}} = \mathbf{e} \\ \mathbf{p}(0) = s(\mathbf{x}) \quad \forall t \in [0, d).\ s[\mathbf{x} \mapsto \mathbf{p}(t)](B)\end{array}}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle \trianglerighteq []_{i \in I}(ch_i * \to c_i), s) \xrightarrow{\langle d, \mathbf{p}, \mathbf{Rdy}(\cup_{i \in I} ch_i *) \rangle} (\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle \trianglerighteq []_{i \in I}(ch_i * \to c_i), s[\mathbf{x} \mapsto \mathbf{p}(d)])}$$

$$\frac{\neg s(B)}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle \trianglerighteq []_{i \in I}(ch_i * \to c_i), s) \xrightarrow{\tau} (\text{skip}, s)} \qquad \frac{i \in I \quad ch_i * = ch!e}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B \rangle \trianglerighteq []_{i \in I}(ch_i * \to c_i), s) \xrightarrow{\langle ch!, s(e) \rangle} (c_i, s)}$$

$$\frac{(c_1, s_1) \xrightarrow{\tau} (c_1', s_1')}{(c_1 \|_{cs} c_2, s_1 \uplus s_2) \xrightarrow{\tau} (c_1' \|_{cs} c_2, s_1' \uplus s_2)} \qquad \frac{ch \notin cs \quad (c_1, s_1) \xrightarrow{\langle ch \triangleright, v \rangle} (c_1', s_1')}{(c_1 \|_{cs} c_2, s_1 \uplus s_2) \xrightarrow{\langle ch \triangleright, v \rangle} (c_1' \|_{cs} c_2, s_1' \uplus s_2)}$$

$$\frac{\begin{array}{c}\mathrm{compat}(\mathbf{Rdy}_1, \mathbf{Rdy}_2) \\ (c_1, s_1) \xrightarrow{\langle d, \mathbf{p}_1, \mathbf{Rdy}_1 \rangle} (c_1', s_1') \quad (c_2, s_2) \xrightarrow{\langle d, \mathbf{p}_2, \mathbf{Rdy}_2 \rangle} (c_2', s_2')\end{array}}{(c_1 \|_{cs} c_2, s_1 \uplus s_2) \xrightarrow{\langle d, \mathbf{p}_1 \uplus \mathbf{p}_2, \mathbf{Rdy}_1 \cup \mathbf{Rdy}_2 \rangle} (c_1' \|_{cs} c_2', s_1' \uplus s_2')}$$

$$\frac{ch \in cs \quad (c_1, s_1) \xrightarrow{\langle ch!, v \rangle} (c_1', s_1') \quad (c_2, s_2) \xrightarrow{\langle ch?, v \rangle} (c_2', s_2')}{(c_1 \|_{cs} c_2, s_1 \uplus s_2) \xrightarrow{\langle ch, v \rangle} (c_1' \|_{cs} c_2', s_1' \uplus s_2')}$$

Big Step Semantics for HCSP

$$\frac{}{(ch!e, s) \Rightarrow (s, \langle ch!, s(e)\rangle)} \quad \frac{}{(ch!e, s) \Rightarrow (s, \langle d, I_s, \{ch!\}\rangle^\frown \langle ch!, s(e)\rangle)} \quad \frac{}{(ch!e, s) \Rightarrow (s, \langle \infty, I_s, \{ch!\}\rangle)}$$

$$\frac{\neg s(B)}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B\rangle, s) \Rightarrow (s, \epsilon)} \quad \frac{\begin{array}{c}\mathbf{p} \text{ is a solution of the ODE } \dot{\mathbf{x}} = \mathbf{e} \\ \mathbf{p}(0) = s(\mathbf{x}) \quad \forall t \in [0, d).\, s[\mathbf{x} \mapsto \mathbf{p}(t)](B) \quad \neg s[\mathbf{x} \mapsto \mathbf{p}(d)](B)\end{array}}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B\rangle, s) \Rightarrow (s[\mathbf{x} \mapsto \mathbf{p}(d)], \langle d, \mathbf{p}, \emptyset\rangle)}$$

$$\frac{\begin{array}{c}\mathbf{p} \text{ is a solution of the ODE } \dot{\mathbf{x}} = \mathbf{e} \quad \mathbf{p}(0) = s_1(\mathbf{x}) \\ \forall t \in [0, d).\, s_1[\mathbf{x} \mapsto \mathbf{p}(t)](B) \quad i \in I \quad ch_i* = ch!e \quad (c_i, s_1[\mathbf{x} \mapsto \mathbf{p}(d)]) \Rightarrow (s_2, tr)\end{array}}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B\rangle \unrhd \|_{i\in I}(ch_i* \to c_i), s_1) \Rightarrow (s_2, \langle d, \mathbf{p}, \mathbf{Rdy}(\cup_{i\in I} ch_i*)\rangle^\frown \langle ch!, s_1[\mathbf{x} \mapsto \mathbf{p}(t)](e)\rangle^\frown tr)}$$

$$\frac{\begin{array}{c}\mathbf{p} \text{ is a solution of the ODE } \dot{\mathbf{x}} = \mathbf{e} \\ \mathbf{p}(0) = s_1(\mathbf{x}) \quad \forall t \in [0, d).\, s_1[\mathbf{x} \mapsto \mathbf{p}(t)](B) \quad \neg s_1[\mathbf{x} \mapsto \mathbf{p}(d)](B)\end{array}}{(\langle \dot{\mathbf{x}} = \mathbf{e} \& B\rangle \unrhd \|_{i\in I}(ch_i* \to c_i), s_1) \Rightarrow (s_1[\mathbf{x} \mapsto \mathbf{p}(d)], \langle d, \mathbf{p}, \mathbf{Rdy}(\cup_{i\in I} ch_i*)\rangle)}$$

$$\frac{(c_1, s_1) \Rightarrow (s_1', tr_1) \quad (c_2, s_2) \Rightarrow (s_2', tr_2) \quad tr_1 \|_{cs} tr_2 \Downarrow tr}{(c_1 \|_{cs} c_2, s_1 \uplus s_2) \Rightarrow (s_1' \uplus s_2', tr)}$$

Hoare Logic

Assertion Logic FOD_Γ

$$
\begin{array}{rcl}
val & := & x_i \mid c \mid v + w \mid v \cdot w \mid \cdots \\
time & := & d \mid \infty \mid d_1 + d_2 \mid d_1 - d_2 \mid \cdots \\
vector & := & (x_1, \ldots, x_n) \mid \mathbf{x} \mid \mathbf{p}(t) \\
state\_traj & := & I_{\mathbf{x}_0} \mid \mathbf{p}_{\mathbf{x}_0, \mathbf{e}} \mid \mathbf{p}(\cdot + d) \mid \mathbf{p}_1 \uplus \mathbf{p}_2 \\
generalized\_event & := & \langle ch\triangleright, val \rangle \mid \langle time, state\_traj, rdy \rangle \\
trace & := & \epsilon \mid generalized\_event \mid \gamma \mid trace_1 {}^\frown trace_2
\end{array}
$$

Hoare triple

Hoare triple is of the form $\{P\}\ c\ \{Q\}$, where P and Q are assertions on state and trace. It means starting from initial state and trace satisfying P, after executing c, the final state and trace satisfies Q.

The proof system consists four parts :

1. **Axioms and inference rules for first-order theory of differential equations** (FOD), omitted
2. **Axioms and inference rules for timed traces and readiness**
3. **Axioms and inference rules for HCSP constructs**
4. **General rules** such as **consequence rule**, omitted

Part II : Axioms and Inference Rules for Timed Traces and Readiness

$$\frac{\langle ch_1 \triangleright_1, v_1\rangle^\frown tr_1 \|_{cs} \langle ch_2 \triangleright_2, v_2\rangle^\frown tr_2 \Downarrow tr \quad ch_1 \in cs \quad ch_2 \in cs}{\exists tr'.\ ch_1 = ch_2 \wedge v_1 = v_2 \wedge (\triangleright_1, \triangleright_2) \in \{(!, ?), (?, !)\} \wedge tr = \langle ch, v\rangle^\frown tr' \wedge tr_1 \|_{cs} tr_2 \Downarrow tr'}\ \text{SyncPairE}$$

$$\frac{\langle ch_1 \triangleright_1, v_1\rangle^\frown tr_1 \|_{cs} \langle ch_2 \triangleright_2, v_2\rangle^\frown tr_2 \Downarrow tr \quad ch_1 \notin cs \quad ch_2 \in cs}{\exists tr'.\ tr = \langle ch_1 \triangleright_1, v_1\rangle^\frown tr' \wedge tr_1 \|_{cs} \langle ch_2 \triangleright_2, v_2\rangle^\frown tr_2 \Downarrow tr'}\ \text{SyncUnpairE1}$$

$$\frac{ch \in cs}{\langle ch \triangleright, v\rangle^\frown tr_1 \|_{cs} \langle d, \mathbf{p}, \mathbf{Rdy}\rangle^\frown tr_2 \not\Downarrow tr}\ \text{SyncPairE2}$$

$$\frac{\neg\, \mathrm{compat}(\mathbf{Rdy}_1, \mathbf{Rdy}_2)}{\langle d_1, \mathbf{p}_1, \mathbf{Rdy}_1\rangle^\frown tr_1 \|_{cs} \langle d_2, \mathbf{p}_2, \mathbf{Rdy}_2\rangle^\frown tr_2 \not\Downarrow tr}\ \text{SyncWaitE1}$$

$$\frac{\langle d, \mathbf{p}_1, \mathbf{Rdy}_1\rangle^\frown tr_1 \|_{cs} \langle d, \mathbf{p}_2, \mathbf{Rdy}_2\rangle^\frown tr_2 \Downarrow tr \quad \mathrm{compat}(\mathbf{Rdy}_1, \mathbf{Rdy}_2)}{\exists tr'.\ tr = \langle d, \mathbf{p}_1 \uplus \mathbf{p}_2, \mathbf{Rdy}_1 \cup \mathbf{Rdy}_2\rangle^\frown tr' \wedge tr_1 \|_{cs} tr_2 \Downarrow tr'}\ \text{SyncWaitE2}$$

Part III : Axioms and inference rules for HCSP constructs

$$\dfrac{}{\left\{\begin{array}{l} Q[tr^\frown\langle ch!, e\rangle/tr] \wedge \\ \forall d > 0.\ Q[tr^\frown\langle d, I_{\mathbf{x}_0}, \{ch!\}\rangle^\frown\langle ch!, e\rangle/tr] \wedge \\ Q[tr^\frown\langle \infty, I_{\mathbf{x}_0}, \{ch!\}\rangle/tr] \end{array}\right\}\ ch!e\ \{Q\}}\ \text{Output}$$

$$\dfrac{}{\left\{\begin{array}{l} \forall v.\ Q[v/x, tr^\frown\langle ch?, v\rangle/tr] \wedge \\ \forall d > 0.\ \forall v.\ Q[v/x, tr^\frown\langle d, I_{\mathbf{x}_0}, \{ch?\}\rangle^\frown\langle ch?, v\rangle/tr] \wedge \\ Q[tr^\frown\langle \infty, I_{\mathbf{x}_0}, \{ch?\}\rangle/tr] \end{array}\right\}\ ch?x\ \{Q\}}\ \text{Input}$$

$$\dfrac{}{\left\{\begin{array}{l} (\neg B \rightarrow Q) \wedge \\ \forall d > 0.\ (\forall t \in [0, d).\ B[\mathbf{p}_{\mathbf{x}_0}(t)/\mathbf{x}]) \wedge \neg B[\mathbf{p}_{\mathbf{x}_0}(d)/\mathbf{x}] \rightarrow \\ \qquad Q[\mathbf{p}_{\mathbf{x}_0}(d)/\mathbf{x}, tr^\frown\langle d, \mathbf{p}_{\mathbf{x}_0}, \emptyset\rangle/tr] \end{array}\right\}\ \langle \dot{\mathbf{x}} = \mathbf{e} \& B\rangle\ \{Q\}}\ \text{Cont}$$

$$\dfrac{\{P_1\}\ c_1\ \{Q_1\} \quad \{P_2\}\ c_2\ \{Q_2\}}{\{P_1[\epsilon/tr] \wedge P_2[\epsilon/tr]\}\ c_1\|_{cs}c_2\ \{\exists tr_1, tr_2.\ Q_1[tr_1/tr] \wedge Q_2[tr_2/tr] \wedge tr_1\|_{cs}tr_2 \Downarrow tr\}}\ \text{Par}$$

Continuous Relative Completeness

Theorem

Suppose all valid FOD formulas are provable, then if a Hoare triple $\{P\}\ c\ \{Q\}$ is valid, then it is provable in the above proof system.

- Define its **weakest pre-condition and strongest post-condition**
- Express all predicates appearing in weakest pre-condition and strongest post-condition in FOD, including traces, synchronization, and predicates

Discrete Relative Completeness

$$\forall d > 0.\,(\forall t \in [0,d).\,B[\mathbf{p}_{\mathbf{x}_0}(t)/\mathbf{x}]) \wedge \neg B[\mathbf{p}_{\mathbf{x}_0}(d)/\mathbf{x}] \rightarrow Q[\mathbf{p}_{\mathbf{x}_0}(d)/\mathbf{x}, tr^\frown\langle d, \mathbf{p}_{\mathbf{x}_0}, \emptyset\rangle/tr] \qquad \text{(CP)}$$

$$\begin{array}{l} \forall T > 0.\,(\forall 0 \leq t < T.\,\exists \epsilon_0 > 0.\,\forall 0 < \epsilon < \epsilon_0. \\ \quad \exists h_0 > 0.\,\forall 0 < h < h_0.\,\mathbf{f}_{\mathbf{x}_0,h}(t) \in \mathcal{U}_{-\epsilon}(B)) \rightarrow \\ \quad (\exists \epsilon_0 > 0.\,\forall 0 < \epsilon < \epsilon_0.\,\exists h_0 > 0.\,\forall 0 < h < h_0. \\ \quad\quad \mathbf{f}_{\mathbf{x}_0,h}(T) \in \neg\mathcal{U}_{-\epsilon}(B) \rightarrow (\mathbf{f}_{\mathbf{x}_0,h}(T), tr^\frown\langle T, \mathbf{f}_{\mathbf{x}_0,h}, \emptyset\rangle) \in \mathcal{U}_{-\epsilon}(Q)) \end{array} \qquad \text{(DP)}$$

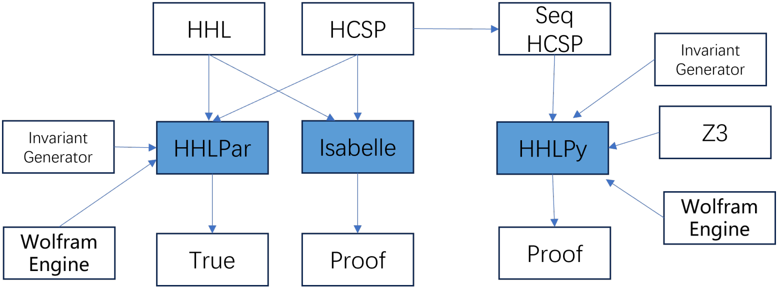
$$\forall d > 0.\,(\forall t \in [0,d).\,B[\mathbf{p}_{\mathbf{x}_0}(t)/\mathbf{x}]) \rightarrow Q[\mathbf{p}_{\mathbf{x}_0}(d)/\mathbf{x}, tr^\frown\langle d, \mathbf{p}_{\mathbf{x}_0}, \mathbf{Rdy}(\cup_{i\in I} ch_i*)\rangle^\frown\langle ch!, e[\mathbf{p}_{\mathbf{x}_0}(d)/\mathbf{x}]\rangle/tr] \qquad \text{(CI)}$$

$$\begin{array}{l} \forall T > 0.\,(\forall 0 \leq t < T.\,\exists \epsilon_0 > 0.\,\forall 0 < \epsilon < \epsilon_0. \\ \quad \exists h_0 > 0.\,\forall 0 < h < h_0.\,\mathbf{f}_{\mathbf{x}_0,h}(t) \in \mathcal{U}_{-\epsilon}(B)) \rightarrow \\ \exists \epsilon_0 > 0.\,\forall 0 < \epsilon < \epsilon_0.\,\exists h_0 > 0.\,\forall 0 < h < h_0.\,(\mathbf{f}_{\mathbf{x}_0,h}(T), \\ \quad tr^\frown\langle T, \mathbf{f}_{\mathbf{x}_0,h}, \mathbf{Rdy}(\cup_{i\in I} ch_i*)\rangle^\frown\langle ch!, e[\mathbf{f}_{\mathbf{x}_0,h}(T)/\mathbf{x}]\rangle) \in \mathcal{U}_{-\epsilon}(Q) \end{array} \qquad \text{(DI)}$$

Theorem (Discrete Relative Completeness)

- *The proof system plus CP $\leftrightarrow$ DP and CI $\leftrightarrow$ DI, are complete relative to the discrete fragment, without referring to solutions of differential equations.*
- *Moreover, the discrete fragment of the proof system is complete in Cook's sense.*

HHL Theorem Prover [ICFEM'15, FM'23]



Part 2 : Verification

Invariant Generation : Theoretical Aspects [EMSOFT'11]

Complete Method to DI : Differential Invariant

- Program

  x:=1;

  while (x<=1000000000)

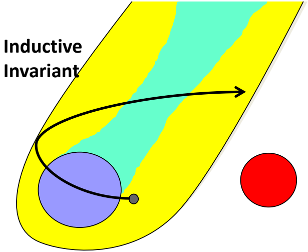
  { x:=x+1; }

  $x \leqq 0$

- Inductive Invariant
  - $x=1 \rightarrow x \geqq 1$
  - $x \geqq 1 \rightarrow x+1 \geqq 1$
  - $x \geqq 1 \rightarrow \neg(x \leqq 0)$

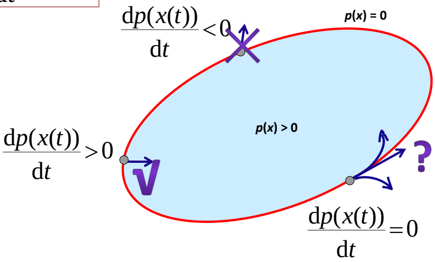
Continuous system

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x)$$



Complete Method to DI : Lie Derivatives

$$\boxed{\frac{\mathrm{d}x}{\mathrm{d}t} = f(x)}$$



Lie Derivative :

$$\mathcal{L}_{\mathbf{f}}^{k} p(\mathbf{x}) \widehat{=} \begin{cases} p(\mathbf{x}), & k = 0, \\ \left\langle \frac{\partial}{\partial \mathbf{x}} \mathcal{L}_{\mathbf{f}}^{k-1} p(\mathbf{x}), \mathbf{f}(\mathbf{x}) \right\rangle, & k > 0. \end{cases}$$

Complete Method to DI : Lie Derivatives

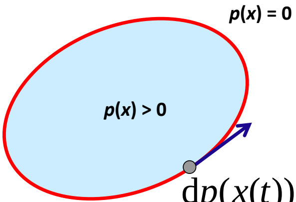
$$\frac{\mathrm{d}^1 p}{\mathrm{d}t^1} > 0$$

$$\vee \frac{\mathrm{d}^1 p}{\mathrm{d}t^1} = 0 \wedge \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} > 0$$

$$\vee \frac{\mathrm{d}^1 p}{\mathrm{d}t^1} = 0 \wedge \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} = 0 \wedge \frac{\mathrm{d}^3 p}{\mathrm{d}t^3} > 0$$

$$\vee \frac{\mathrm{d}^1 p}{\mathrm{d}t^1} = 0 \wedge \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} = 0 \wedge \frac{\mathrm{d}^3 p}{\mathrm{d}t^3} = 0 \wedge \cdots$$

?



$$\frac{\mathrm{d}p(x(t))}{\mathrm{d}t} = 0$$

Complete Method to DI : Necessary and Sufficient Conditioin

- $f(x)$ and $p(x)$ are polynomials
- Compute an upper bound N s.t.
- $p(x) \geq 0$ is an inductive invariant of $\frac{\mathrm{d}x}{\mathrm{d}t} = f(x)$

iff

$$p=0 \Rightarrow \left(\frac{\mathrm{d}^1 p}{\mathrm{d}t^1} > 0 \vee \right.$$

$$\frac{\mathrm{d}^1 p}{\mathrm{d}t^1} = 0 \wedge \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} > 0 \vee$$

$$\cdots \quad \cdots \quad \vee$$

$$\left. \frac{\mathrm{d}^1 p}{\mathrm{d}t^1} = 0 \wedge \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} = 0 \wedge \cdots \wedge \frac{\mathrm{d}^N p}{\mathrm{d}t^N} \geq 0 \right)$$

Complete Method to DI : Main Results

- Semi-algebraic set

$$\bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} p_{ij}(\mathbf{x}) \triangleright 0, \quad \triangleright \in \{\geq, >\}$$

- First-order theory of real numbers is decidable
  - Quantifier Elimination

Checking whether a semi-algebraic set is an inductive invariant of a polynomial continuous dynamical systems is decidable

1

Complete Method to DI : General Case

- **Problem :** Consider a PDS $(D, \mathbf{f})$ with

$$D = \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} p_{ij}(\mathbf{x}) \triangleright 0,$$

and $\mathbf{f} \in \mathbb{Q}^n[\mathbf{x}]$, where $\triangleright \in \{\geq, >\}$, to generate SAIs automatically with a general template

$$P = \bigvee_{k=1}^{K} \bigwedge_{l=1}^{L_k} p_{kl}(\mathbf{u}_{kl}, \mathbf{x}) \triangleright 0\,, \ \triangleright \in \{\geq, >\}$$

- **Basic idea** The procedure is essentially same as in the simple case, but have to sophisticatedly handle the complex combinations due to the complicated boundaries.

Complete Method to DI : General Case

Theorem (Main Result)

A semi-algebraic template $P(\mathbf{u},\mathbf{x})$ *defined by*

$$\bigvee_{k=1}^{K}\left(\bigwedge_{j=1}^{j_k} p_{kj}(\mathbf{u}_{kj},\mathbf{x})\geq 0 \quad\wedge \bigwedge_{j=j_k+1}^{J_k} p_{kj}(\mathbf{u}_{kj},\mathbf{x})>0\right)$$

is a CI of the PCCDS $(D,\mathbf{f})$ *with*

$$D\,\widehat{=}\bigvee_{m=1}^{M}\left(\bigwedge_{l=1}^{l_m} p_{ml}(\mathbf{x})\geq 0 \quad\wedge \bigwedge_{l=l_m+1}^{L_m} p_{ml}(\mathbf{x})>0\right),$$

iff $\mathbf{u}$ *satisfies*

$$\forall\mathbf{x}.\Big(\big(P\wedge D\wedge\Phi_D\rightarrow\Phi_P\big)\wedge\big(\neg P\wedge D\wedge\Phi_D^{Iv}\rightarrow\neg\Phi_P^{Iv}\big)\Big),$$

where

Complete Method to DI : General Case

Theorem (Main Result (Cont'd))

$$\Phi_D \widehat{=} \bigvee_{m=1}^{M} \left(\bigwedge_{l=1}^{l_m} \psi_0^+(p_{ml}, \mathrm{f}) \wedge \bigwedge_{l=l_m+1}^{L_m} \psi^+(p_{ml}, \mathrm{f}) \right),$$

$$\Phi_P \widehat{=} \bigvee_{k=1}^{K} \left(\bigwedge_{j=1}^{j_k} \psi_0^+(p_{kj}, \mathrm{f}) \wedge \bigwedge_{j=j_k+1}^{J_k} \psi^+(p_{kj}, \mathrm{f}) \right),$$

$$\Phi_D^{Iv} \widehat{=} \bigvee_{m=1}^{M} \left(\bigwedge_{l=1}^{l_m} \varphi_0^+(p_{ml}, \mathrm{f}) \wedge \bigwedge_{l=l_m+1}^{L_m} \varphi^+(p_{ml}, \mathrm{f}) \right),$$

$$\Phi_P^{Iv} \widehat{=} \bigvee_{k=1}^{K} \left(\bigwedge_{j=1}^{j_k} \varphi_0^+(p_{kj}, \mathrm{f}) \wedge \bigwedge_{j=j_k+1}^{J_k} \varphi^+(p_{kj}, \mathrm{f}) \right),$$

$$\psi^+(p, \mathrm{f}) \widehat{=} \bigvee_{0 \le i \le N_{p,\mathrm{f}}} \psi^{(i)}(p, \mathrm{f}) \textit{ with } \psi^{(i)}(p, \mathrm{f}) \widehat{=} \left(\bigwedge_{0 \le j < i} L_\mathrm{f}^j p = 0 \right) \wedge L_\mathrm{f}^i p > 0, \textit{ and}$$

$$\psi_0^+(p, \mathrm{f}) \widehat{=} \psi^+(p, \mathrm{f}) \vee \left(\bigwedge_{0 \le j \le N_{p,\mathrm{f}}} L_\mathrm{f}^j p = 0 \right)$$

$$\varphi^+(p, \mathrm{f}) \widehat{=} \bigvee_{0 \le i \le N_{p,\mathrm{f}}} \varphi^{(i)}(p, \mathrm{f}) \textit{ with } \varphi^{(i)}(p, \mathrm{f}) \widehat{=} \left(\bigwedge_{0 \le j < i} L_\mathrm{f}^j p = 0 \right) \wedge (-1)^i \cdot L_\mathrm{f}^i p > 0, \textit{ and}$$

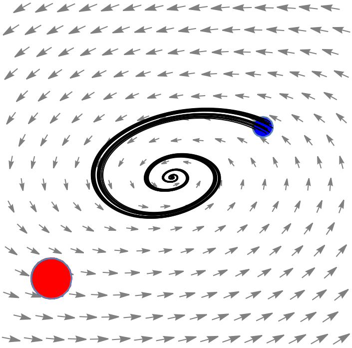
$$\varphi_0^+(p, \mathrm{f}) \widehat{=} \varphi^+(p, \mathrm{f}) \vee \left(\bigwedge_{0 \le j \le N_{p,\mathrm{f}}} L_\mathrm{f}^j p = 0 \right).$$

Part 2 : Verification

Invariant Generation : Practical Aspects [CAV'21, Inf.&Comp.'22, FM'24]

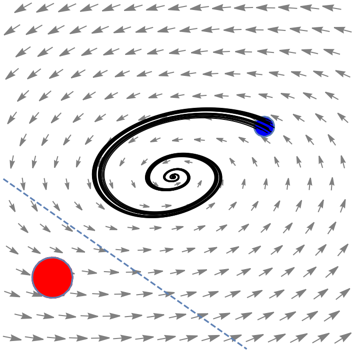
Barrier Certificates (BCs) vs. Inductive Invariants

$$\forall \mathbf{x}_0 \in \Theta. \forall t \in [0, T) : B(\xi_{\mathbf{x}_0}(t)) \leq 0,$$
$$\forall \mathbf{x} \in \mathcal{U} : B(\mathbf{x}) > 0.$$



Barrier Certificates (BCs) vs. Inductive Invariants

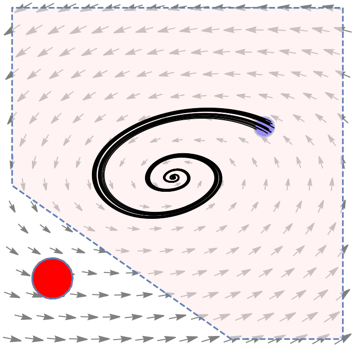
$$\forall \mathbf{x}_0 \in \Theta. \forall t \in [0, T): B(\xi_{\mathbf{x}_0}(t)) \leq 0,$$
$$\forall \mathbf{x} \in \mathcal{U}: B(\mathbf{x}) > 0.$$



Barrier Certificates (BCs) vs. Inductive Invariants

$$\forall \mathbf{x}_0 \in \Theta. \forall t \in [0, T) : B(\xi_{\mathbf{x}_0}(t)) \leq 0,$$
$$\forall \mathbf{x} \in \mathcal{U} : B(\mathbf{x}) > 0.$$

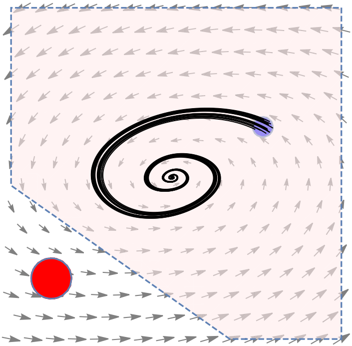
$$\forall \mathbf{x}_0 \in \Psi. \forall t \in [0, T) : \xi_{\mathbf{x}_0}(t) \in \Psi.$$

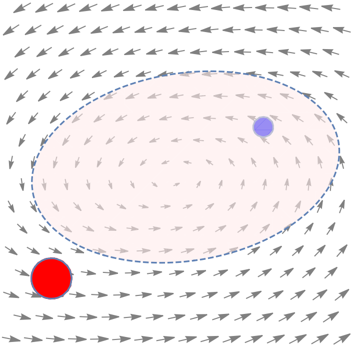


Barrier Certificates (BCs) vs. Inductive Invariants

$$\forall \mathbf{x}_0 \in \Theta. \forall t \in [0, T) : B(\xi_{\mathbf{x}_0}(t)) \leq 0,$$
$$\forall \mathbf{x} \in \mathcal{U} : B(\mathbf{x}) > 0.$$

$$\forall \mathbf{x}_0 \in \Psi. \forall t \in [0, T) : \xi_{\mathbf{x}_0}(t) \in \Psi.$$





BC Conditions vs. Inductive Invariance

- **BC Conditions**
  - original cond. **Prajna & Jadbabaie, 2004**
  - exponential-type cond. **Kong et al., 2013**
  - Darboux-type cond. **Zeng et al., 2016**
  - general cond. **Dai et al., 2017**
  - vector cond & semantics cond.. **Sogokon et al., 2018**
  - non-convex cond.'s **Yang et al., 2015, Zhang et al., 2018, Chen et al., 2020**

BC Conditions vs. Inductive Invariance

- **BC Conditions**
  - original cond. **Prajna & Jadbabaie, 2004**
  - exponential-type cond. **Kong et al., 2013**
  - Darboux-type cond. **Zeng et al., 2016**
  - general cond. **Dai et al., 2017**
  - vector cond & semantics cond.. **Sogokon et al., 2018**
  - non-convex cond.'s **Yang et al., 2015, Zhang et al., 2018, Chen et al., 2020**
- **Inductive Invariant Conditions**
  - Sufficient condition **Platzer & Clarke, 2008**
  - Necessary and sufficient condition **Liu et al., 2011**

BC Conditions vs. Inductive Invariance

- **BC Conditions**
  - original cond. **Prajna & Jadbabaie, 2004**
  - exponential-type cond. **Kong et al., 2013**
  - Darboux-type cond. **Zeng et al., 2016**
  - general cond. **Dai et al., 2017**
  - vector cond & semantics cond.. **Sogokon et al., 2018**
  - non-convex cond.'s **Yang et al., 2015, Zhang et al., 2018, Chen et al., 2020**
- **Inductive Invariant Conditions**
  - Sufficient condition **Platzer & Clarke, 2008**
  - Necessary and sufficient condition **Liu et al., 2011**

$\Rightarrow$ **Invariant BC cond.**

- Invariant BC is essentially same as inductive invariant
- **Q. Wang, M. Chen, B. Xue, N. Zhan, J.-P. Katoen :**
  Synthesizing Invariant Barrier Certificates via DCP. CAV '21.
  Encoding inductive invariants as barrier certificates : Synthesis via DCP. I&C '22.

Invariant Barrier Certificates

- **Invariant Barrier Certificates** : the counterpart of program invariants in continuous systems

Invariant Barrier Certificates

- **Invariant Barrier Certificates** : the counterpart of program invariants in continuous systems
  1. $\forall \mathbf{x} \in \Theta : B(\mathbf{x}) \leq 0;$ (initial)
  2. $\forall \mathbf{x} \in \mathcal{U} : B(\mathbf{x}) > 0;$ (separation)
  3. $\forall \mathbf{x} \in \mathbb{R}^n : \bigwedge_{i=1}^{N_{B,f}} \left(\left(\bigwedge_{j=0}^{i-1} \mathcal{L}_{\mathrm{f}}^{j} B(\mathbf{x}) = 0 \right) \implies \mathcal{L}_{\mathrm{f}}^{i} B(\mathbf{x}) \leq 0 \right).$ (consecution)

Invariant Barrier Certificates

- **Invariant Barrier Certificates** : the counterpart of program invariants in continuous systems
  1. $\forall \mathbf{x} \in \Theta : B(\mathbf{x}) \le 0;$ (initial)
  2. $\forall \mathbf{x} \in \mathcal{U} : B(\mathbf{x}) > 0;$ (separation)
  3. $\forall \mathbf{x} \in \mathbb{R}^n : \bigwedge_{i=1}^{N_{B,f}} \left(\left(\bigwedge_{j=0}^{i-1} \mathcal{L}_{\mathrm{f}}^{j} B(\mathbf{x}) = 0 \right) \implies \mathcal{L}_{\mathrm{f}}^{i} B(\mathbf{x}) \le 0 \right).$ (consecution)

Theorem (Inductive invariance)

If $B(\mathbf{x})$ is an invariant barrier certificate, then $\Psi = \{\mathbf{x} \mid B(\mathbf{x}) \le 0\}$ is an invariant. Conversely, if $\Psi = \{\mathbf{x} \mid B(\mathbf{x}) \le 0\}$ is an invariant satisfying $\Theta \subseteq \Psi$ and $\Psi \cap \mathcal{U} = \emptyset$, then $B(\mathbf{x})$ is an invariant barrier certificate.

Sufficient Condition for Invariant BCs

A polynomial $B \in \mathbb{R}[\mathbf{x}]$ is an invariant BC if for some $\epsilon \in \mathbb{R}^+$, there exist $v_{i,j} \in \mathbb{R}[\mathbf{x}]$ and sum-of-squares (SOS) polynomials $\sigma(\mathbf{x}), \sigma'(\mathbf{x})$ s.t.

1. $-B(\mathbf{x}) + \sigma(\mathbf{x})\mathcal{I}(\mathbf{x})$, (initial)
2. $B(\mathbf{x}) + \sigma'(\mathbf{x})\mathcal{U}(\mathbf{x}) - \epsilon$, (separation)
3. for all $1 \le i \le N_{B,\mathbf{f}}$, $-\mathcal{L}_{\mathbf{f}}^{i}B(\mathbf{x}) + \sum_{j=0}^{i-1} v_{i,j}(\mathbf{x})\mathcal{L}_{\mathbf{f}}^{j}B(\mathbf{x})$ (consecution)

are SOS polynomials.

Sufficient Condition for Invariant BCs

A polynomial $B \in \mathbb{R}[\mathbf{x}]$ is an invariant BC if for some $\epsilon \in \mathbb{R}^+$, there exist $v_{i,j} \in \mathbb{R}[\mathbf{x}]$ and sum-of-squares (SOS) polynomials $\sigma(\mathbf{x}), \sigma'(\mathbf{x})$ s.t.

1. $-B(\mathbf{x}) + \sigma(\mathbf{x})\mathcal{I}(\mathbf{x})$, (initial)
2. $B(\mathbf{x}) + \sigma'(\mathbf{x})\mathcal{U}(\mathbf{x}) - \epsilon$, (separation)
3. for all $1 \leq i \leq N_{B,\mathbf{f}}$, $-\mathcal{L}_{\mathbf{f}}^{i}B(\mathbf{x}) + \sum_{j=0}^{i-1} v_{i,j}(\mathbf{x})\mathcal{L}_{\mathbf{f}}^{j}B(\mathbf{x})$ (consecution)

are SOS polynomials.

Example (running)

Set a template $B(\mathbf{a}, \mathbf{x}) = ax_2$, we have $N_{B,\mathbf{f}} = 1$. We expect SOS polynomials :

$$-\underbrace{ax_2}_{B} + \sigma(\mathbf{x})\underbrace{\left(x_1^2 + (x_2 - 2)^2 - 1\right)}_{\mathcal{I}}, \qquad \text{(initial)}$$

$$\underbrace{ax_2}_{B} + \sigma'(\mathbf{x})\underbrace{(x_2 + 1)}_{\mathcal{U}} - 0.01, \qquad \text{(separation)}$$

$$-\underbrace{a\left(x_1x_2 - 0.5x_2^2 + 0.1\right)}_{\mathcal{L}_{\mathbf{f}}^{1}B} + v(\mathbf{x})\underbrace{ax_2}_{\mathcal{L}_{\mathbf{f}}^{0}B}. \qquad \text{(consecution)}$$

Necessary Condition for Invariant BCs

If $B \in \mathbb{R}[\mathbf{x}]$ is an invariant BC, then for some $\epsilon \in \mathbb{R}^+$, there exist $v_{i,j} \in \mathbb{R}[\mathbf{x}]$ and SOS polynomials $\sigma(\mathbf{x}), \sigma'(\mathbf{x}), \rho(\mathbf{x}), \rho'(\mathbf{x}), \rho_i''(\mathbf{x})$ s.t. for any $L \in \mathbb{R}^+$,

1. $-B(\mathbf{x}) + \rho(\mathbf{x})(\|\mathbf{x}\|^2 - L) + \sigma(\mathbf{x})\mathcal{I}(\mathbf{x}) + \epsilon$, (initial)
2. $B(\mathbf{x}) + \rho'(\mathbf{x})(\|\mathbf{x}\|^2 - L) + \sigma'(\mathbf{x})\mathcal{U}(\mathbf{x})$, (separation)
3. for $1 \le i \le N_{B,\mathbf{f}}$,
   $-\mathcal{L}_{\mathbf{f}}^i B(\mathbf{x}) + \rho_i''(\mathbf{x})(\|\mathbf{x}\|^2 - L) + \sum_{j=0}^{i-1} v_{i,j}(\mathbf{x})\mathcal{L}_{\mathbf{f}}^j B(\mathbf{x}) + \epsilon$ (consecution)

are SOS polynomials.

Necessary Condition for Invariant BCs

If $B \in \mathbb{R}[\mathbf{x}]$ is an invariant BC, then for some $\epsilon \in \mathbb{R}^+$, there exist $v_{i,j} \in \mathbb{R}[\mathbf{x}]$ and SOS polynomials $\sigma(\mathbf{x}), \sigma'(\mathbf{x}), \rho(\mathbf{x}), \rho'(\mathbf{x}), \rho_i''(\mathbf{x})$ s.t. for any $L \in \mathbb{R}^+$,

1. $-B(\mathbf{x}) + \rho(\mathbf{x})(\|\mathbf{x}\|^2 - L) + \sigma(\mathbf{x})\mathcal{I}(\mathbf{x}) + \epsilon$, (initial)
2. $B(\mathbf{x}) + \rho'(\mathbf{x})(\|\mathbf{x}\|^2 - L) + \sigma'(\mathbf{x})\mathcal{U}(\mathbf{x})$, (separation)
3. for $1 \leq i \leq N_{B,\mathbf{f}}$,
   $-\mathcal{L}_{\mathbf{f}}^i B(\mathbf{x}) + \rho_i''(\mathbf{x})(\|\mathbf{x}\|^2 - L) + \sum_{j=0}^{i-1} v_{i,j}(\mathbf{x})\mathcal{L}_{\mathbf{f}}^j B(\mathbf{x}) + \epsilon$ (consecution)

are SOS polynomials.

⇒ Consequence of Putinar's Positivstellensatz.

Bilinear Matrix Inequality (BMI) Feasibility

$$h(\mathbf{a}, \mathbf{s}, \mathbf{x}) \in \Sigma^{\leq 2d}[\mathbf{x}]$$

Bilinear Matrix Inequality (BMI) Feasibility

$$\begin{aligned} h(\mathbf{a},\mathbf{s},\mathbf{x}) &\in \Sigma^{\leq 2d}[\mathbf{x}] \\ &\Updownarrow \; h(\mathbf{a},\mathbf{s},\mathbf{x}) = \mathbf{b}^{\mathsf{T}} Q(\mathbf{a},\mathbf{s})\mathbf{b} \\ Q(\mathbf{a},\mathbf{s}) &\succeq 0 \end{aligned}$$

Bilinear Matrix Inequality (BMI) Feasibility

$$
\begin{aligned}
h(\mathbf{a}, \mathbf{s}, \mathbf{x}) &\in \Sigma^{\leq 2d}[\mathbf{x}] \\
&\Updownarrow\ h(\mathbf{a}, \mathbf{s}, \mathbf{x}) = \mathbf{b}^{\mathsf{T}} Q(\mathbf{a}, \mathbf{s}) \mathbf{b} \\
Q(\mathbf{a}, \mathbf{s}) &\succeq 0 \\
&\Updownarrow\ \mathcal{F}(\mathbf{a}, \mathbf{s}) = -Q(\mathbf{a}, \mathbf{s}) \\
\mathcal{F}(\mathbf{a}, \mathbf{s}) &\preceq 0
\end{aligned}
$$

Bilinear Matrix Inequality (BMI) Feasibility

$$
\begin{aligned}
h(\mathbf{a},\mathbf{s},\mathbf{x}) \;&\in\; \Sigma^{\leq 2d}[\mathbf{x}] \\
&\Big\Updownarrow\; h(\mathbf{a},\mathbf{s},\mathbf{x}) = \mathbf{b}^{\mathsf{T}} Q(\mathbf{a},\mathbf{s})\mathbf{b} \\
Q(\mathbf{a},\mathbf{s}) \;&\succeq\; 0 \\
&\Big\Updownarrow\; \mathcal{F}(\mathbf{a},\mathbf{s}) = -Q(\mathbf{a},\mathbf{s}) \\
\mathcal{F}(\mathbf{a},\mathbf{s}) \;&\preceq\; 0
\end{aligned}
$$

Example (running)

Assume $d = 1$ and $v(\mathrm{s},\mathrm{x}) = s_0 + s_1 x_1 + s_2 x_2$:

$$-a\left(x_1 x_2 - 0.5x_2^2 + 0.1\right) + v(\mathrm{x})ax_2 \in \Sigma^{\leq 2}[\mathrm{x}] \qquad \text{(consecution)}$$

Bilinear Matrix Inequality (BMI) Feasibility

$$\begin{aligned} h(\mathbf{a},\mathbf{s},\mathbf{x}) &\in \Sigma^{\leq 2d}[\mathbf{x}] \\ &\Updownarrow\ h(\mathbf{a},\mathbf{s},\mathbf{x}) = \mathbf{b}^{\mathsf{T}} Q(\mathbf{a},\mathbf{s})\mathbf{b} \\ Q(\mathbf{a},\mathbf{s}) &\succeq 0 \\ &\Updownarrow\ \mathcal{F}(\mathbf{a},\mathbf{s}) = -Q(\mathbf{a},\mathbf{s}) \\ \mathcal{F}(\mathbf{a},\mathbf{s}) &\preceq 0 \end{aligned}$$

Example (running)

Assume $d = 1$ and $v(\mathbf{s},\mathbf{x}) = s_0 + s_1x_1 + s_2x_2$:

$$-a\left(x_1x_2 - 0.5x_2^2 + 0.1\right) + v(\mathbf{x})ax_2 \in \Sigma^{\leq 2}[\mathbf{x}] \qquad \text{(consecution)}$$

$$\mathcal{F}(\mathbf{a},\mathbf{s}) = -\begin{pmatrix} -0.1a & 0 & 0.5as_0 \\ 0 & 0 & 0.5(as_1 - a) \\ 0.5as_0 & 0.5(as_1 - a) & as_2 + 0.5a \end{pmatrix} \preceq 0$$

BMI Optimization

$$\mathcal{F}_\iota(\mathbf{a}, \mathbf{s}) \preceq 0, \quad \iota = 1, 2, \ldots, l. \qquad (1)$$

BMI Optimization

$$\mathcal{F}_\iota(\mathbf{a},\mathbf{s}) \preceq 0, \quad \iota = 1,2,\ldots,l. \tag{1}$$

$$\Updownarrow$$

$$\begin{aligned} &\underset{\lambda,\mathbf{a},\mathbf{s}}{\text{maximize}} \quad \lambda \\ &\text{subject to} \quad \mathcal{B}_\iota(\lambda,\mathbf{a},\mathbf{s}) \widehat{=} \mathcal{F}_\iota(\mathbf{a},\mathbf{s}) + \lambda I \preceq 0, \quad \iota = 1,2,\ldots,l. \end{aligned} \tag{2}$$

BMI Optimization

$$\mathcal{F}_{\iota}(\mathbf{a},\mathbf{s}) \preceq 0, \quad \iota = 1,2,\ldots,l. \tag{1}$$

$\Updownarrow$ (1) is feasible iff $\lambda^* \geq \mathbf{0}$ in (2)

$$\begin{aligned} &\underset{\lambda,\mathbf{a},\mathbf{s}}{\text{maximize}} && \lambda \\ &\text{subject to} && \mathcal{B}_{\iota}(\lambda,\mathbf{a},\mathbf{s}) \widehat{=} \mathcal{F}_{\iota}(\mathbf{a},\mathbf{s}) + \lambda I \preceq 0, \quad \iota = 1,2,\ldots,l. \end{aligned} \tag{2}$$

General BMI Optimization

$$\begin{array}{ll} \underset{\mathbf{z}=(\mathbf{x},\mathbf{y})}{\text{maximize}} & g(\mathbf{z}) \\ \text{subject to} & \mathcal{B}(\mathbf{x},\mathbf{y}) \widehat{=} \sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j F_{i,j} + \sum_{i=1}^{m} x_i H_i + \sum_{j=1}^{n} y_j G_j + F \preceq 0 \end{array}$$

General BMI Optimization

$$\begin{array}{ll} \text{maximize} & g(\mathbf{z}) \\ \mathbf{z} = (\mathbf{x}, \mathbf{y}) & \\ \\ \text{subject to} & \mathcal{B}(\mathbf{x}, \mathbf{y}) \widehat{=} \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j F_{i,j} + \sum_{i=1}^{m} x_i H_i + \sum_{j=1}^{n} y_j G_j + F \preceq 0 \end{array}$$

$$\mathcal{B}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \mathbf{x} \otimes I \\ \mathbf{y} \otimes I \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} 0 & \Gamma \\ \Gamma^{\mathsf{T}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \otimes I \\ \mathbf{y} \otimes I \end{pmatrix} + \begin{pmatrix} \Omega_1 & \Omega_2 \end{pmatrix} \begin{pmatrix} \mathbf{x} \otimes I \\ \mathbf{y} \otimes I \end{pmatrix} + F$$

General BMI Optimization

$$\begin{array}{ll} \underset{\mathbf{z}=(\mathbf{x},\mathbf{y})}{\text{maximize}} & g(\mathbf{z}) \\ \text{subject to} & \mathcal{B}(\mathbf{x},\mathbf{y}) \widehat{=} \sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j F_{i,j} + \sum_{i=1}^{m} x_i H_i + \sum_{j=1}^{n} y_j G_j + F \preceq 0 \end{array}$$

$$\mathcal{B}(\mathbf{x},\mathbf{y}) = \begin{pmatrix} \mathbf{x}\otimes I \\ \mathbf{y}\otimes I \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} 0 & \Gamma \\ \Gamma^{\mathsf{T}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}\otimes I \\ \mathbf{y}\otimes I \end{pmatrix} + \begin{pmatrix} \Omega_1 & \Omega_2 \end{pmatrix} \begin{pmatrix} \mathbf{x}\otimes I \\ \mathbf{y}\otimes I \end{pmatrix} + F$$

Difference-of-Convex (DC) Decomposition

$$M = V^{\mathsf{T}} D V \qquad \text{(eigendecomposition)}$$

Difference-of-Convex (DC) Decomposition

$$\begin{aligned} M &= V^{\mathsf{T}} D V && \text{(eigendecomposition)} \\ &= \underbrace{V^{\mathsf{T}} D^{+} V}_{M_1 \succeq 0} - \underbrace{V^{\mathsf{T}} D^{-} V}_{M_2 \succeq 0} \end{aligned}$$

Difference-of-Convex (DC) Decomposition

$$\begin{aligned} M &= V^{\intercal} D V && \text{(eigendecomposition)} \\ &= \underbrace{V^{\intercal} D^{+} V}_{M_1 \succeq 0} - \underbrace{V^{\intercal} D^{-} V}_{M_2 \succeq 0} \end{aligned}$$

$$\mathcal{B}(\mathbf{x}, \mathbf{y}) = \underbrace{\mathcal{B}^{+}(\mathbf{x}, \mathbf{y})}_{\text{psd-convex}} - \underbrace{\mathcal{B}^{-}(\mathbf{x}, \mathbf{y})}_{\text{psd-convex}} \qquad \text{(DC decomposition)}$$

Difference-of-Convex (DC) Decomposition

$$\begin{aligned} M &= V^T D V & \text{(eigendecomposition)} \\ &= \underbrace{V^T D^+ V}_{M_1 \succeq 0} - \underbrace{V^T D^- V}_{M_2 \succeq 0} \end{aligned}$$

$$\mathcal{B}(\mathbf{x}, \mathbf{y}) = \underbrace{\mathcal{B}^+(\mathbf{x}, \mathbf{y})}_{\text{psd-convex}} - \underbrace{\mathcal{B}^-(\mathbf{x}, \mathbf{y})}_{\text{psd-convex}} \qquad \text{(DC decomposition)}$$

Example (running)

The decomposition of $\mathcal{B}(\lambda, \mathbf{a}, \mathbf{s})$ for consecution, for instance, gives

$$\mathcal{B}^+(\lambda, \mathbf{a}, \mathbf{s}) =$$

$$\frac{1}{8}\begin{pmatrix} 8\lambda + 0.08a + a^2 + 0.408s_0^2 & 0.408s_0s_1 & -2as_0 + 0.816s_0s_2 \\ 0.408s_0s_1 & 8\lambda + a^2 + 0.408s_1^2 & 4a - 2as_1 + 0.816s_1s_2 \\ -2as_0 + 0.816s_0s_2 & 4a - 2as_1 + 0.816s_1s_2 & 8\lambda - 4a + 2.449a^2 - 4as_2 + s_0^2 + s_1^2 + 1.632s_2^2 \end{pmatrix}$$

$$\mathcal{B}^-(\lambda, \mathbf{a}, \mathbf{s}) =$$

$$\frac{1}{8}\begin{pmatrix} a^2 + 0.408s_0^2 & 0.408s_0s_1 & 2as_0 + 0.816s_0s_2 \\ 0.408s_0s_1 & a^2 + 0.408s_1^2 & 2as_1 + 0.816s_1s_2 \\ 2as_0 + 0.816s_0s_2 & 2as_1 + 0.816s_1s_2 & 2.449a^2 + 4as_2 + s_0^2 + s_1^2 + 1.632s_2^2 \end{pmatrix}.$$

Reducing to a Series of Convex Programs

Linearize the "concave part" $-\mathcal{B}^{-}(\mathrm{x}, \mathrm{y})$ around a feasible solution $\mathbf{z}^{k}$:

$$\mathcal{B}^{+}(\mathbf{z}) - \mathcal{B}^{-}\left(\mathbf{z}^{k}\right) - \mathcal{D}\mathcal{B}^{-}\left(\mathbf{z}^{k}\right)\left(\mathbf{z} - \mathbf{z}^{k}\right) \preceq 0 \qquad \text{(QMIs)}$$

Reducing to a Series of Convex Programs

Linearize the "concave part" $-\mathcal{B}^{-}(\mathrm{x}, \mathrm{y})$ around a feasible solution $\mathbf{z}^k$:

$$\mathcal{B}^{+}(\mathbf{z}) - \mathcal{B}^{-}\left(\mathbf{z}^k\right) - \mathcal{D}\mathcal{B}^{-}\left(\mathbf{z}^k\right)\left(\mathbf{z} - \mathbf{z}^k\right) \preceq 0 \qquad \text{(QMIs)}$$

$\Updownarrow$ Schur complement

$$\begin{pmatrix} -I & N(\mathbf{z} \otimes I) \\ (\mathbf{z} \otimes I)^{\mathsf{T}} N^{\mathsf{T}} & -\mathcal{B}^{-}\left(\mathbf{z}^k\right) - \mathcal{D}\mathcal{B}^{-}\left(\mathbf{z}^k\right)\left(\mathbf{z} - \mathbf{z}^k\right) + \Omega(\mathbf{z} \otimes I) + F \end{pmatrix} \preceq 0 \qquad \text{(LMIs)}$$

Reducing to a Series of Convex Programs

Linearize the "concave part" $-\mathcal{B}^{-}(\mathrm{x}, \mathrm{y})$ around a feasible solution z^k :

$$\mathcal{B}^{+}(\mathbf{z}) - \mathcal{B}^{-}\left(\mathbf{z}^k\right) - \mathcal{D}\mathcal{B}^{-}\left(\mathbf{z}^k\right)\left(\mathbf{z} - \mathbf{z}^k\right) \preceq 0 \qquad \text{(QMIs)}$$

$\Updownarrow$ Schur complement

$$\begin{pmatrix} -I & N(\mathbf{z} \otimes I) \\ (\mathbf{z} \otimes I)^{\mathsf{T}} N^{\mathsf{T}} & -\mathcal{B}^{-}\left(\mathbf{z}^k\right) - \mathcal{D}\mathcal{B}^{-}\left(\mathbf{z}^k\right)\left(\mathbf{z} - \mathbf{z}^k\right) + \Omega(\mathbf{z} \otimes I) + F \end{pmatrix} \preceq 0 \qquad \text{(LMIs)}$$

⇒ DCP : an iterative procedure that solves **a series of convex programs**.

Finding the Initial Solution

$$\begin{aligned} \underset{\lambda, \mathbf{a}}{\text{maximize}} \quad & \lambda \\ \text{subject to} \quad & \mathcal{F}_\iota(\mathbf{a}, \mathbf{s})\big|_{\mathbf{s}=(\mathbf{c}_\iota, \mathbf{0}, \ldots, \mathbf{0})} + \lambda I \preceq 0, \quad \iota = 1, 2, \ldots, l. \end{aligned}$$

Here, $c_\iota \in \mathbb{R}_0^+$ encodes a non-negative constant multiplier polynomial.

Finding the Initial Solution

$$\begin{aligned} \underset{\lambda,\mathbf{a}}{\text{maximize}} \quad & \lambda \\ \text{subject to} \quad & \mathcal{F}_\iota(\mathbf{a},\mathbf{s})\Big|_{\mathbf{s}=(\mathbf{c}_\iota,\mathbf{0},\ldots,\mathbf{0})} + \lambda I \preceq 0, \quad \iota = 1,2,\ldots,l. \end{aligned}$$

Here, $c_\iota \in \mathbb{R}_0^+$ encodes a non-negative constant multiplier polynomial.

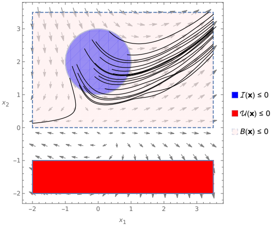
⇒ This LMI optimization always admits a strictly feasible solution $(\lambda, \mathbf{a})$ which induces also a strictly feasible solution $(\lambda, \mathbf{a}, (c_\iota, 0, \ldots, 0))$ to the original BMI optimization.

Properties of DCP

Example (running)

Our iterative procedure starts with a strictly feasible initial solution $\mathbf{z}^0$ and terminates with $\lambda^2 \geq 0$ (subject to numerical round-off) and $a^2 = -0.00363421$, yielding the barrier certificate

$$B\left(\mathbf{a}^2, \mathbf{x}\right) = -0.00363421 x_2 \leq 0.$$

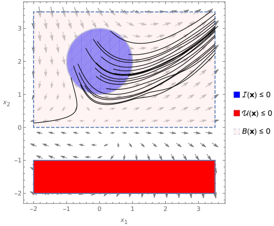


Properties of DCP

Example (running)

Our iterative procedure starts with a strictly feasible initial solution $\mathbf{z}^0$ and terminates with $\lambda^2 \geq 0$ (subject to numerical round-off) and $a^2 = -0.00363421$, yielding the barrier certificate

$$B\left(\mathbf{a}^2, \mathbf{x}\right) \ = \ -0.00363421 x_2 \leq 0.$$



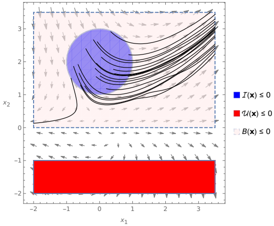
⇒ **Soundness :** every $\mathbf{z}^i$ is a feasible solution to the original BMI optimization ;

Properties of DCP

Example (running)

Our iterative procedure starts with a strictly feasible initial solution $\mathbf{z}^0$ and terminates with $\lambda^2 \geq 0$ (subject to numerical round-off) and $a^2 = -0.00363421$, yielding the barrier certificate

$$B\left(\mathbf{a}^2, \mathbf{x}\right) \; = \; -0.00363421 x_2 \leq 0.$$



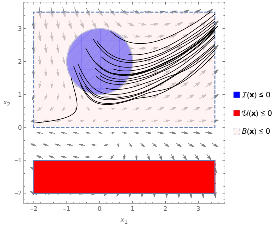
- ⇒ **Soundness :** every $\mathbf{z}^i$ is a feasible solution to the original BMI optimization;
- ⇒ **Convergence :** $\left\{\mathbf{z}^i\right\}_{i \in \mathbb{N}}$ converges to a KKT point (local optimum);

Properties of DCP

Example (running)

Our iterative procedure starts with a strictly feasible initial solution $\mathbf{z}^0$ and terminates with $\lambda^2 \geq 0$ (subject to numerical round-off) and $a^2 = -0.00363421$, yielding the barrier certificate

$$B\left(\mathbf{a}^2, \mathbf{x}\right) \;=\; -0.00363421 x_2 \leq 0.$$



- **Soundness :** every $\mathbf{z}^i$ is a feasible solution to the original BMI optimization ;
- **Convergence :** $\{\mathbf{z}^i\}_{i \in \mathbb{N}}$ converges to a KKT point (local optimum) ;
- **Weak completeness (via branch-and-bound) :** an invariant BC is guaranteed to be found (under mild assumptions) whenever there exists an inductive invariant (in the form of the given template).

Experiments

Table 1: Empirical results on benchmark examples (time in seconds)

Example name	n_{sys}	d_{flow}	d_{BC}	BMI-DC			PENLAB		SOSTOOLS	
				#iter.	time	verified	time	verified	time	verified
overview [11]	2	2	1	2	**0.03**	✓	0.31	✓	0.07	✓
contrived	2	1	2	0	**0.01**	✓	0.48	✓	0.75	✓
lie-der [37]	2	2	1	0	**0.01**	✓	0.22	✓	0.04	✓
lorenz [11]	3	2	2	8	**2.37**	✓	75.11	✗	1.47	✗
lti-stable [19]	2	1	2	0	**0.01**	✓	0.23	✓	0.14	✓
lotka-volterra [21]	3	2	1	3	**0.07**	✓	0.36	✓	0.21	✓
clock [44]	2	3	1	0	**0.01**	✓	0.88	✗	0.18	✗
lyapunov [45]	3	3	2	4	1.25	✓	56.98	✗	**0.35**	✓
arch1 [52]	2	5	2	0	**0.01**	✓	33.76	✗	0.31	✓
arch2 [52]	2	2	2	5	**0.37**	✓	0.38	✗	0.17	✗
arch3 [52]	2	3	2	1	**0.07**	✓	0.54	✓	0.18	✓
arch4 [52]	2	2	1	2	0.09	✓	0.49	✗	**0.06**	✓
barr-cert1 [42]	2	3	2	12	**0.85**	✓	2.53	✗	0.09	✗
barr-cert2 [11]	2	2	2	6	**1.57**	✓	1.16	✗	0.15	✓
barr-cert3 [65]	2	2	1	0	**0.01**	✓	0.20	✓	0.11	✗
barr-cert4 [65]	2	3	2	13	**0.96**	✓	0.89	✗	0.23	✗
fitzhugh-nagumo [48]	2	3	2	2	**0.16**	✓	1.24	✓	0.25	✗
stabilization [49]	3	2	2	9	2.88	✓	55.22	✓	**0.11**	✓
lie-high-order	2	1	2	32	**4.12**	✓	1.56	✗	0.25	✗
raychaudhuri [13]	4	2	2	34	**9.51**	✓	33.64	✗	0.14	✗
focus [43]	2	1	4	100	54.89	✗	0.95	✗	0.48	✗
sys-bio1 [28]	7	2	2	2	73.22	?	101.95	?	1.35	?
sys-bio2 [28]	9	2	1	1	1.03	?	15.54	?	0.16	?
quadcopter [19]	12	1	1	0	0.03	?	65.42	?	0.36	?

Generalization to Unbounded Domains

- **Archimedean condition** in Putinar's Positivstellensatz : there exists $N \in \mathbb{N}$ such that :

$$N - \|x\|^2 = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) \cdot g_i(x) \text{ for some } \sigma_i(x) \in \mathbb{R}[x]$$

  which requires that the system domain is bounded within a ball $N - \|x\|^2 \geq 0$

Generalization to Unbounded Domains

- **Archimedean condition** in Putinar's Positivstellensatz :
  there exists $N \in \mathbb{N}$ such that :

$$N - \|x\|^2 = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) \cdot g_i(x) \text{ for some } \sigma_i(x) \in \mathbb{R}[x]$$

  which requires that the system domain is bounded within a ball $N - \|x\|^2 \geq 0$
- For problems with unbounded domains, due to the violation of the Archimedean condition, existing invariant synthesis methods become **incomplete** (but still sound).

Generalization to Unbounded Domains

- **Archimedean condition** in Putinar's Positivstellensatz :
  there exists $N \in \mathbb{N}$ such that :

$$N - \|x\|^2 = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) \cdot g_i(x) \text{ for some } \sigma_i(x) \in \mathbb{R}[x]$$

  which requires that the system domain is bounded within a ball $N - \|x\|^2 \geq 0$
- For problems with unbounded domains, due to the violation of the Archimedean condition, existing invariant synthesis methods become **incomplete** (but still sound).
- Solution : **homogenization**, a recent advance in polynomial optimization field **Huang et al.,[MP'23]**

Homogenization

- Fix an auxiliary variable x_0. Given a polynomial $p(\mathrm{x})$ of degree d, its **homogenization** is

$$\tilde{p}(\tilde{\mathrm{x}}) = x_0^d p(x_1/x_0, \ldots, x_n/x_0)$$

$$p(x_1, x_2) = x_1^2 + x_2 + 1 \longrightarrow \tilde{p}(x_0, x_1, x_2) = x_1^2 + x_2 x_0 + x_0^2$$

Homogenization

- Fix an auxiliary variable x_0. Given a polynomial $p(\mathbf{x})$ of degree d, its **homogenization** is
$$\tilde{p}(\tilde{\mathbf{x}}) = x_0^d p(x_1/x_0, \ldots, x_n/x_0)$$
$$p(x_1, x_2) = x_1^2 + x_2 + 1 \longrightarrow \tilde{p}(x_0, x_1, x_2) = x_1^2 + x_2 x_0 + x_0^2$$
- For $\mathbb{K} = \{\mathbf{x} \mid p_i(\mathbf{x}) \geq 0,\ i = 1, \ldots, m\}$, we define
$$\begin{aligned} \tilde{\mathbb{K}}_{>0} &= \{\tilde{\mathbf{x}} \in \mathbb{R}^{n+1} \mid \tilde{p}_1(\tilde{\mathbf{x}}) \geq 0, \ldots, \tilde{p}_m(\tilde{\mathbf{x}}) \geq 0, \|\tilde{\mathbf{x}}\|^2 = 1, x_0 > 0\}, \\ \tilde{\mathbb{K}} &= \{\tilde{\mathbf{x}} \in \mathbb{R}^{n+1} \mid \tilde{p}_1(\tilde{\mathbf{x}}) \geq 0, \ldots, \tilde{p}_m(\tilde{\mathbf{x}}) \geq 0, \|\tilde{\mathbf{x}}\|^2 = 1, x_0 \geq 0\}, \end{aligned}$$

Homogenization

- Fix an auxiliary variable x_0. Given a polynomial $p(\mathrm{x})$ of degree d, its **homogenization** is
$$\tilde{p}(\tilde{\mathrm{x}}) = x_0^d p(x_1/x_0, \dots, x_n/x_0)$$
$$p(x_1, x_2) = x_1^2 + x_2 + 1 \longrightarrow \tilde{p}(x_0, x_1, x_2) = x_1^2 + x_2 x_0 + x_0^2$$
- For $\mathbb{K} = \{\mathrm{x} \mid p_i(\mathrm{x}) \geq 0,\ i = 1, \dots, m\}$, we define
$$\begin{aligned}\tilde{\mathbb{K}}_{>0} &= \{\tilde{\mathrm{x}} \in \mathbb{R}^{n+1} \mid \tilde{p_1}(\tilde{\mathrm{x}}) \geq 0, \dots, \tilde{p}_m(\tilde{\mathrm{x}}) \geq 0, \|\tilde{\mathrm{x}}\|^2 = 1, x_0 > 0\},\\ \tilde{\mathbb{K}} &= \{\tilde{\mathrm{x}} \in \mathbb{R}^{n+1} \mid \tilde{p_1}(\tilde{\mathrm{x}}) \geq 0, \dots, \tilde{p}_m(\tilde{\mathrm{x}}) \geq 0, \|\tilde{\mathrm{x}}\|^2 = 1, x_0 \geq 0\},\end{aligned}$$
- One to one mapping between $\tilde{\mathbb{K}}_{>0}$ and $\mathbb{K}$, $(x_0, \mathrm{x}) \mapsto \frac{\mathrm{x}}{x_0}$

Theorem (Huang et al., 2023)

When $\mathbb{K}$ is closed at ∞, i.e., closure$(\tilde{\mathbb{K}}_{>0}) = \tilde{\mathbb{K}}$

unbounded in $\mathbb{R}^n$ $\quad f(\mathrm{x}) \geq 0$ *over* $\mathbb{K} \iff \tilde{f}(\tilde{\mathrm{x}}) \geq 0$ *over* $\tilde{\mathbb{K}}$ $\quad$ *bounded in $\mathbb{R}^{n+1}$*

Synthesising BC over Unbounded Domains

- Fix ϵ and λ in the definition of exponential-type barrier certificate.
- **Sound Characterization** still works, but conservative.

1. $-B(\mathbf{x}) = \sigma_0(\mathbf{x}) + \sum_i \sigma_i(\mathbf{x}) g_i^{\mathcal{I}}(\mathbf{x})$, (initial)
2. $B(\mathbf{x}) - \epsilon = \sigma_0'(\mathbf{x}) + \sum_i \sigma_i'(\mathbf{x}) g_i^{\mathcal{U}}(\mathbf{x})$, (separation)
3. $\lambda B(\mathbf{x}) - \mathfrak{L}_f B(\mathbf{x}) = \sigma_0''(\mathbf{x}) + \sum_i \sigma_i''(\mathbf{x}) g_i^{\mathcal{X}}(\mathbf{x})$, (consecution)

- **Complete Characterization** A polynomial $B(\mathbf{x})$ of degree d is a BC **only if** for any $\epsilon_0 \in \mathbb{R}^+$, there exist some sum-of-squares (SOS) polynomials $\sigma_i(\tilde{\mathbf{x}}), \rho(\tilde{\mathbf{x}})$ and polynomial $\tau(\tilde{\mathbf{x}}) \in \mathbb{R}[\tilde{\mathbf{x}}]$ s.t.

1. $-\tilde{B}(\tilde{\mathbf{x}}) + \epsilon_0 = \sigma_0(\tilde{\mathbf{x}}) + \sum_i \sigma_i(\tilde{\mathbf{x}}) \tilde{g}_i^{\mathcal{I}}(\tilde{\mathbf{x}}) + \rho(\tilde{\mathbf{x}}) x_0 + \tau(\tilde{\mathbf{x}})(1 - \|\mathbf{x}\|^2)$,
2. $\tilde{B}(\tilde{\mathbf{x}}) - \epsilon x_0^d + \epsilon_0 = \sigma_0'(\tilde{\mathbf{x}}) + \sum_i \sigma_i'(\tilde{\mathbf{x}}) \tilde{g}^{\mathcal{U}}(\tilde{\mathbf{x}}) + \rho(\tilde{\mathbf{x}}) x_0 + \tau(\tilde{\mathbf{x}})(1 - \|\mathbf{x}\|^2)$,
3. $\tilde{h}(\tilde{\mathbf{x}}) + \epsilon_0 = \sigma_0''(\tilde{\mathbf{x}}) + \sum_i \sigma_i''(\tilde{\mathbf{x}}) \tilde{g}_i^{\mathcal{X}}(\tilde{\mathbf{x}}) + \rho(\tilde{\mathbf{x}}) x_0 + \tau(\tilde{\mathbf{x}})(1 - \|\mathbf{x}\|^2)$.

where $h(\mathbf{x}) = \lambda B(\mathbf{x}) - \mathfrak{L}_f B(\mathbf{x})$. Under some non-singular assumptions, the $\epsilon_0 = 0$ case is both sound and complete.

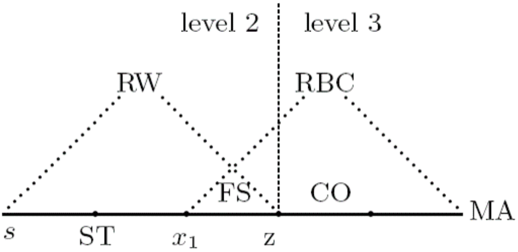
Part 2 : Verification

A Case Study : CTCS-3 [VSTTE'13, SCIS'15]

A Combined Scenario of CTCS-3

Informal Description

- CTCS-3 is an informal specification of Chinese high speed train that ensures safety and high throughput of trains.
- For historical reasons, CTCS currently contains two levels : level 2 and level 3.
- 14 scenarios
  - ⇒ Movement authority
  - ⇒ Level change (upgrade, degrade)
  - ⇒ Mode conversion (FS to CO)
  - ⇒ . . .



Formal Proof with HHLprover

```
definition Train :: proc where
  "Train =
    Rep (
      T ::= (λ_. 0);
      Cont (ODE ((λ_ _. 0)(S := (λs. s V),
                           V := (λs. s A), T := (λ_. 1))))
                 ((λs. s T < Period ∧ s V > 0));
      Wait (λs. Period - s T);
      Cm (''Train2Control''[!](λs. s V));
      Cm (''Train2Control''[!](λs. s S));
      Cm (''Control2Train''[?]A)
    )"
lemma Train_prop:
  "⊨
    {λs tr. s = (λ_. 0)(V := v0, S := s0, A := a0, T := t0) ∧ emp_t tr}
      Train
    {λs tr. ∃xs. s = Train_end_state (v0, s0, a0, t0) xs ∧
                 Train_inv (v0, s0, a0, t0) xs tr}"
```

Formal Proof with HHLprover

```
definition Control :: proc where
  "Control =
    Level ::= (λ_. 2.5);Next_seg_v ::= (λ_. 0);
    Rep(
      Cm (''Train2Control''[?]V);
      Cm (''Train2Control''[?]S);
      (IF (λs. s Level = 2.5 ∧ s S > Stop_point) THEN
          Level ::= (λ_. 3)
       ELSE Skip FI);
      (IF (λs. s Level = 3 ∧ s S < Stop_point / 2) THEN
          Next_seg_v ::= (λ_. Next_V_limit)
       ELSE Skip FI);
      Command_a ::= (λs. com_a_gen (s S) (s V) (s Next_seg_v));
      Cm (''Control2Train''[!](λs. s Command_a)))"
lemma Control_prop:
  "⊨
    {λs tr. s = ((λ_. 0)(V := v0, S := s0, Command_a := a0,
                         Level := l0, Next_seg_v := 0 )) ∧ empₜ tr}
      Control
    {λs tr. ∃xs. s = Control_end_state (v0, s0, a0, l0) xs
                 ∧ Control_blocks (v0, s0, a0, l0) xs tr}"
```

Formal Proof with HHLprover

```
definition system :: pproc where
  "system = Parallel (Single Train)
                     {''Train2Control'', ''Control2Train''}
                     (Single Control)"
lemma combine:
  "loop_invariant (s0, v0) ⟹
   combine_assn {''Train2Control'', ''Control2Train''}
    (Train_inv (v0, s0, com_a s0 v0, t0) as) (Control_blocks (v0, s0, com_a s0 v0, 2.5) vs)
     ⟹t  tot_block (s0,v0) (length as)"
lemma system_Prop:
  assumes "loop_invariant (s0, v0)"
  shows "⊨p
    {pair_assn (λs. s = ((λ_. 0)(V := v0, S := s0, A := com_a s0 v0, T := t0)))
               (λs. s = ((λ_. 0)(V := v0, S := s0, Command_a := com_a s0 v0,
                                 Level := 2.5, Next_seg_v := 0)))}
      system
    {∃g n. trace_gassn (tot_block (s0,v0) n)}"
```

Proof result

- It is proved with HHLprover that the train stops before the location of level transition and mode conversion.
- Details can be found in [Zou et al., VSTTE 2013].