Analysis of Parameterised Timed Systems using Horn Constraints

Hossein Hojjat
Philipp Rümmer
Pavle Subotic
Wang Yi

SynCoP+PV
23 April 2017
Context

- Starting point: Work on general-purpose fixed-point solvers

- Application to timed systems possible?

- Yes, and some more:
  - Parameterisation: #processes, timing
  - Infinite data domains
Overview

- Timed automata
- BIP models
- Parameterised systems
- Software programs etc.

System/Program Property, Analysis rules

Set of Horn constraints

Horn solver (off-the-shelf)

Solvable (timeout)

Unsolvable + Counterexample

E.g., equivalent to: *No error states are reachable*

Floyd-Hoare
Design by contract
Owicki-Gries
Rely Guarantee etc.

HSF
Z3
Eldarica
Duality
Example: ERR unreachable?

Constraints:
\[
\begin{align*}
\forall x, y. \; \text{true} & \rightarrow I_0(x, y) \\
\forall x, y. \; I_0(x, y) & \rightarrow I_1(0, 0) \\
\forall x, y. \; I_1(x, y) & \rightarrow I_2(x + 1, y) \\
\forall x, y. \; I_2(x, y) & \rightarrow I_1(x, x + 2) \\
\forall x, y. \; I_1(x, y) \land y < x & \rightarrow false
\end{align*}
\]

Solution:
\[
\begin{align*}
I_0(x, y) & \equiv true \\
I_1(x, y) & \equiv x \leq y \\
I_2(x, y) & \equiv true
\end{align*}
\]
Outline

- Parameterised analysis by example
- Model of execution
- Encoding
- Experiments
Timed systems

Controller

Trains

[FORTE'94]
Timed systems

System invariant:

\[ g \in G \]

Constraints:

Local transitions:

\[ T_1(g, l_1, g', l'_1) \land Inv_1(g, l_1) \rightarrow Inv_1(g', l'_1) \]

Non-interference:

\[ T_1(g, l_1, g', l'_1) \land Inv_1(g, l_1) \land Inv_2(g, l_2) \rightarrow Inv_2(g', l_2) \]

+ clauses for *time elapse*, *synchronisation*, *initiation*, *assertions*.

\[ Inv_{1,2,3}(g, l_1, l_2, l_3) \]
Invariant schemata

**Modular**
Separate invariant for each process

**Monolithic**
Single invariant for whole system

**Weaker**
Smaller invariants

**Stronger**
Detailed invariants
Parameterised systems

After refinements:

Invariant schema that enables verification
(at most one train crosses bridge at a time)

\[ \forall n_1, n_2, n_3. \ dist(n_1, n_2, n_3) \rightarrow \]
\[ Inv(g, l_2, n_1, l_1[n_1], n_2, l_1[n_2], n_3, l_1[n_3]) \]

\( k \)-indexed invariant
Ashcroft invariant
Basic Systems

- Processes $P$
- Global state space $G$
- Local state space $L_p$ (for each $p \in P$)
- Initial states $Init_p \subseteq G \times L_p$
- Transition relation $(g, l) \xrightarrow{p} (g', l')$ ($g \in G, l \in L_p$)

- Error states $(\langle p_1, E_1 \rangle, \ldots, \langle p_m, E_m \rangle)$

where $p_1, \ldots, p_m \in P$, $E_i \subseteq G \times L_{p_i}$
Execution Model

- System state space
  \[ S = G \times \prod_{p \in P} L_p \]

- Initial system states
  \[ S_0 = \{ (g, \bar{l}) \mid \forall p \in P. (g, \bar{l}[p]) \in Init_p \} \subseteq S \]

- System transition relation
  \[ p \in P \quad (g, \bar{l}[p]) \xrightarrow{p} (g', \bar{l}') \]
  \[ (g, \bar{l}) \rightarrow (g', \bar{l}[p/l']) \]

- System error states
  \[ Err = \{ (g, \bar{l}) \in S \mid \forall i. (g, \bar{l}[p_i]) \in E_i \} \subseteq S \]

- Safety: \[ S_0 \rightarrow^* Err ? \]
Owicki-Gries-style Horn Encoding

- **Finite case:** $P$ is finite

- **Unbounded homogeneous case:** $P = \mathbb{N}$
  One process replicated infinitely often

- **Unbounded heterogeneous case:**
  $$P = \bigcup_{i=1, \ldots, n} (\{i\} \times P_i)$$
  Infinitely many processes of different types
Invariant Schemata (finite case)

- Assuming \( P = \{1, 2, \ldots, n\} \)

- Invariant schema:
  - an anti-chain \( A \subseteq \{0, 1\}^n \)
  - (component-wise comparison)

- Every element of \( A \) represents one invariant
Invariant Schemata
(finite case)

- Given schema \( A \subseteq \{0, 1\}^n \)
  choose relation symbols \( \{R_{\bar{a}} \mid \bar{a} \in A\} \)
to represent invariant

- System invariant

\[
Inv(g, \bar{l}) = \bigwedge_{\bar{a} \in A} R_{\bar{a}}(g, \bar{l}[\bar{a}])
\]
Invariant Schemata (finite case)

For instance \( P = \{1, 2, 3\} \)

\[
A_1 = \{ (1, 0, 0), (0, 1, 0), (0, 0, 1) \}
\]

\[
A_2 = \{ (1, 1, 0), (0, 1, 1) \}
\]

\[
A_3 = \{ (1, 1, 1) \}
\]

Modular
Separate invariant for each process

Monolithic
Single invariant for whole system
Clauses (finite case)

- **Initiation**
  \[
  \left\{ R_{\bar{a}}(g, l_1, \ldots, l_k) \leftarrow Init_{i_1}(g, l_1) \land \cdots \land Init_{i_k}(g, l_k) \right\}_{\bar{a} \in A}
  \]
  \((i_1, \ldots, i_k \text{ non-zero entries in } \bar{a})\)

- **Consecution**
  \[
  \left\{ R_{\bar{a}}(g', \bar{l}[p/l'][\bar{a}]) \leftarrow \langle (g, \bar{l}[p]) \xrightarrow{p} (g', l') \rangle \land R_{\bar{a}}(g, \bar{l}[\bar{a}]) \land Ctxt(\{p\}, g, \bar{l}) \right\}_{p=1, \ldots, n \atop \bar{a} \in A}
  \]

- **Absence of errors**
  \((\langle p_1, E_1 \rangle, \ldots, \langle p_m, E_m \rangle)\)
  \[
  false \leftarrow \left( \bigwedge_{j=1, \ldots, m} (g, \bar{l}[p_j]) \in E_j \right) \land Ctxt(\{p_1, \ldots, p_m\}, g, \bar{l})
  \]

- **Context predicates:**
  \[
  Ctxt(Q, g, \bar{l}) = \bigwedge \left\{ R_{\bar{c}}(g, \bar{l}[\bar{c}]) \mid \bar{c} \in A \text{ and } \exists q \in Q. \bar{c}[q] > 0 \right\}
  \]
Unbounded Invariant Schemata

- Assuming \( P = \bigcup_{i=1,\ldots,n} (\{i\} \times P_i) \)

  with \( P_i = \{0\} \) or \( P_i = \mathbb{N} \)

- Invariant schema: \( A \subseteq \prod_{i=1,\ldots,n} (\{0,1\} \cup P_i) \)

- For instance \( P = \{(0, 0)\} \cup (\{1\} \times \mathbb{N}) \)
  \[ A = \{(1, 3)\} \]
Clauses for (Homogeneous) Unbounded Case

- Assuming $P = \mathbb{N}$ and $A = \{(k)\}$

\[
\left\{ \begin{array}{l}
R(g, p_{\sigma(1)}, l_{\sigma(1)}, \ldots, p_{\sigma(k)}, l_{\sigma(k)}) \leftarrow \text{dist}(p_1, \ldots, p_k) \land R(g, p_1, l_1, \ldots, p_k, l_k) \\
R(g, p_1, l_1, \ldots, p_k, l_k) \leftarrow \text{dist}(p_1, \ldots, p_k) \land \text{Init}(g, l_1) \land \cdots \land \text{Init}(g, l_k)
\end{array} \right\}_{\sigma \in S_k} \tag{6}
\]

\[
R(g', p_1, l'_1, \ldots, p_k, l_k) \leftarrow \text{dist}(p_1, \ldots, p_k) \land ((g, l_1) \overset{p_1}{\rightarrow} (g', l'_1)) \land R(g, p_1, l_1, \ldots, p_k, l_k) \tag{7}
\]

\[
R(g', p_1, l_1, \ldots, p_k, l_k) \leftarrow \text{dist}(p_0, p_1, \ldots, p_k) \land ((g, l_0) \overset{p_0}{\rightarrow} (g', l'_0)) \land R\text{Conj}(0, \ldots, k) \tag{8}
\]

\[
\text{false} \leftarrow \text{dist}(p_1, \ldots, p_r) \land \left( \bigwedge_{j=1, \ldots, m} (p_j = p_j \land (g, l_j) \in E_j) \right) \land R\text{Conj}(1, \ldots, r) \tag{9}
\]

\[
R\text{Conj}(a, \ldots, b) = \bigwedge_{i_1, \ldots, i_k \in \{a, \ldots, b\}, i_1 < i_2 < \cdots < i_k} R(g, p_{i_1}, l_{i_1}, \ldots, p_{i_k}, l_{i_k})
\]
CEX-Guided Refinement of Schemata

- Start with weakest (fully modular) invariant schema $A_0$

- If Horn constraints are not solvable, check whether CEX is genuine
  - Yes → System CEX
  - No → Choose stronger schema $A_{i+1}$

- Example: trains model
Extensions of the Basic Model

- **Physical time** (discrete, dense)
  - Add a global clock (global variable)
  - Delay transitions

- **Communication channels** (UPPAAL-style)
  - Clauses encoding simultaneous transition of two processes

- **Barriers**

- **BIP-style interactions**
Protocol X (link)

Protocol is correct assuming timing parameters $A \leq B$
On Completeness

Concurrent System

Owicki-Gries-style encoding

Set of Horn constraints

Horn solver (off-the-shelf)

Relatively complete?
- Finite case (+ time) ✔
- General infinite case ❌
- Well-structured TS [FSE'16] ✔

Usually incomplete
→ “Best-effort”
# Experiments

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>#Ci</th>
<th>#Cf</th>
<th>(N^\text{th}) (sec)</th>
<th>Inv Schema</th>
<th>Total (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature Control System (unsafe)</td>
<td>48</td>
<td>110</td>
<td>0.37</td>
<td>(1,1,1)</td>
<td>3.86</td>
</tr>
<tr>
<td>Temperature Control System</td>
<td>48</td>
<td>110</td>
<td>1.12</td>
<td>(1,1,1)</td>
<td>4.31</td>
</tr>
<tr>
<td>Fischer</td>
<td>47</td>
<td>221</td>
<td>5.62</td>
<td>(2,1)</td>
<td>12.21</td>
</tr>
<tr>
<td>Fischer (unsafe)</td>
<td>47</td>
<td>221</td>
<td>2.84</td>
<td>(2,1)</td>
<td>8.91</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>50</td>
<td>162</td>
<td>3.60</td>
<td>(2,1)</td>
<td>8.22</td>
</tr>
<tr>
<td>CSMA/CD (unsafe)</td>
<td>50</td>
<td>793</td>
<td>2.91</td>
<td>(4,1)</td>
<td>13.81</td>
</tr>
<tr>
<td>Lynch-Shavit</td>
<td>50</td>
<td>299</td>
<td>66.11</td>
<td>(2)</td>
<td>70.10</td>
</tr>
<tr>
<td>Lynch-Shavit (unsafe)</td>
<td>50</td>
<td>299</td>
<td>2.58</td>
<td>(2)</td>
<td>5.35</td>
</tr>
<tr>
<td>Train Crossing</td>
<td>28</td>
<td>686</td>
<td>2.51</td>
<td>(1,3)</td>
<td>8.84</td>
</tr>
<tr>
<td>Train Crossing (unsafe)</td>
<td>28</td>
<td>240</td>
<td>1.53</td>
<td>(1,2)</td>
<td>3.91</td>
</tr>
</tbody>
</table>

- **Horn solver:** Eldarica  
- **Machine:** Intel Core i7 Duo 2.9 GHz, 8GB  
- **Details:** [HCVS'14]
Conclusions

- Framework for encoding parameterised systems as Horn clauses
- Support for various features: Time, data, communication, parameters
- Feasible for real models? initial experiments promising
- https://github.com/uuverifiers/eldarica
Appendix
(Constrained) Horn clauses

**Definition**
Suppose
- \( \mathcal{L} \) is some constraint language (e.g., Presburger A.);
- \( \mathcal{R} \) is a set of relation symbols;
- \( \mathcal{X} \) is a set of first-order variables.

Then a *Horn clause* is a formula \( C \land B_1, \ldots, B_n \rightarrow H \) where
- \( C \) is a constraint in \( \mathcal{L} \) (without symbols from \( \mathcal{R} \ ));
- each \( B_i \) is a literal of the form \( r(t_1, \ldots, t_m) \);
- \( H \) is either \textit{false}, or of the same form as the \( B_i \).
Solvability

Definition
A set $\mathcal{C}$ of Horn clauses is syntactically/symbolically solvable if the $\mathcal{R}$-symbols can be replaced with constraints such that all clauses become valid.