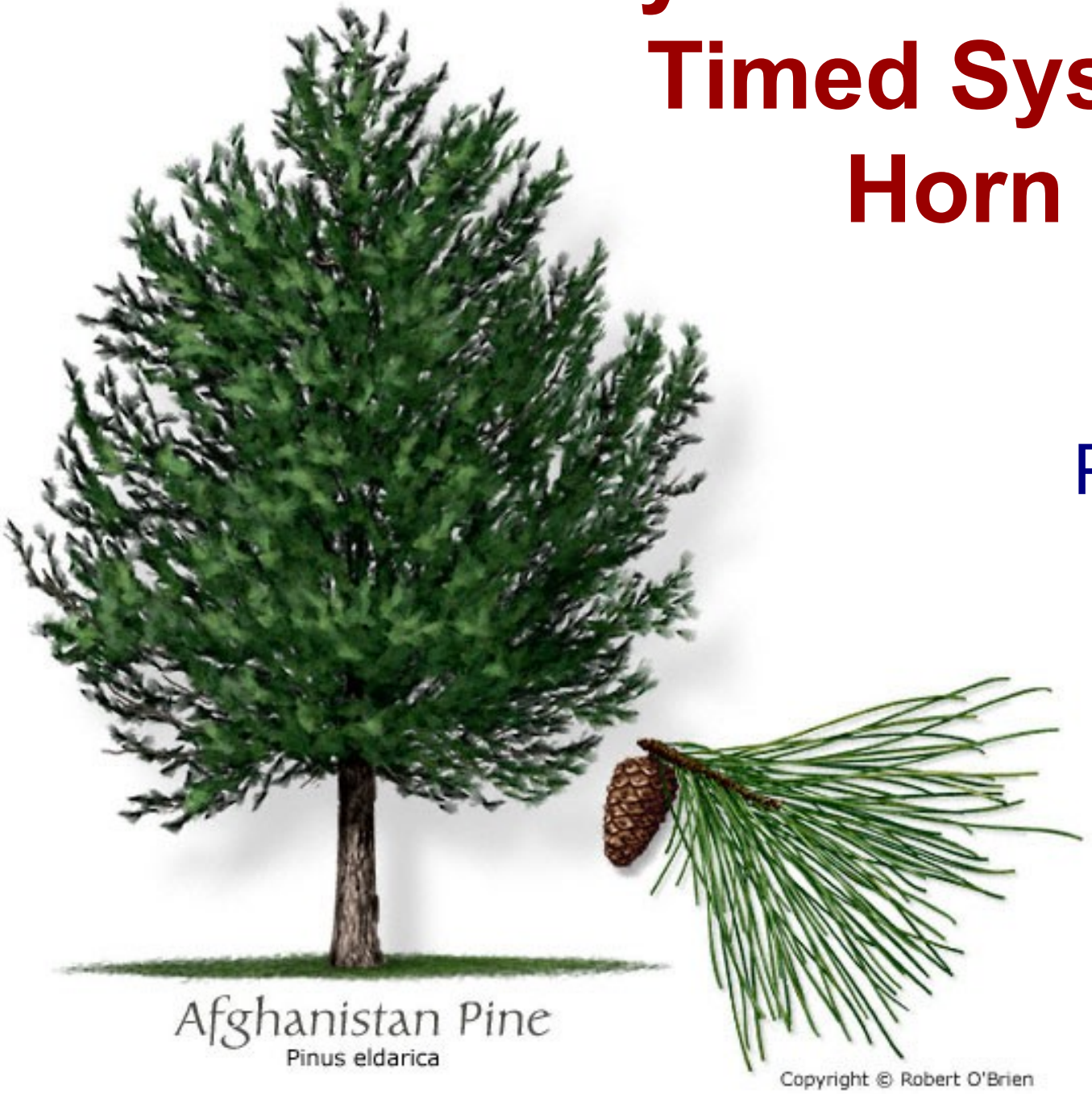
Analysis of Parameterised Timed Systems using Horn Constraints

Hossein Hojjat
Philipp Rümmer
Pavle Subotic
Wang Yi

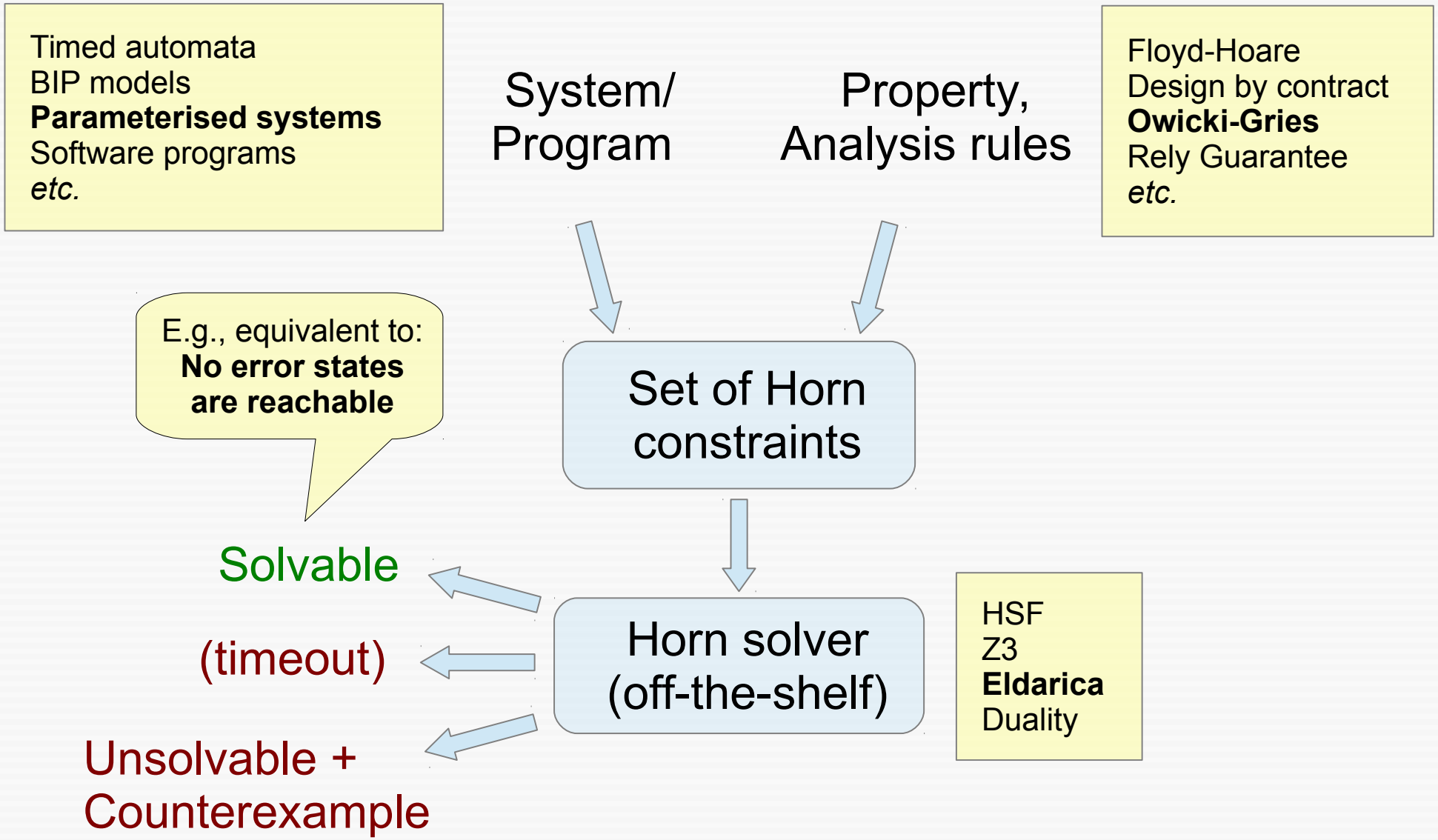


SynCoP+PV
23 April 2017

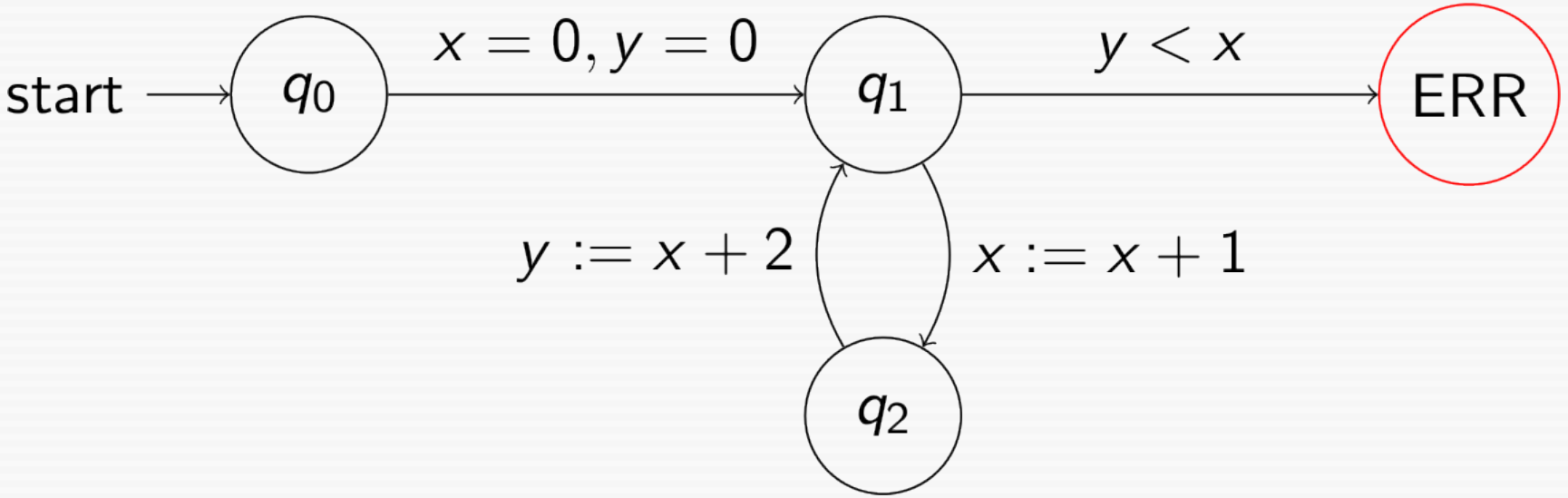
Context

- *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
Floyd-Hoare
Design by contract
Owicki-Gries
Rely Guarantee
etc.
E.g., equivalent to:
No error states are reachable
Set of Horn constraints
Solvable
(timeout)
Unsolvable +
Counterexample
Horn solver
(off-the-shelf)
HSF
Z3
Eldarica
Duality

Example: ERR unreachable?



Constraints:

$$\forall x, y.\ true \rightarrow I_0(x, y)$$
$$\forall x, y.\ I_0(x, y) \rightarrow I_1(0, 0)$$
$$\forall x, y.\ I_1(x, y) \rightarrow I_2(x + 1, y)$$
$$\forall x, y.\ I_2(x, y) \rightarrow I_1(x, x + 2)$$
$$\forall x, y.\ I_1(x, y) \wedge y < x \rightarrow false$$

Solution:

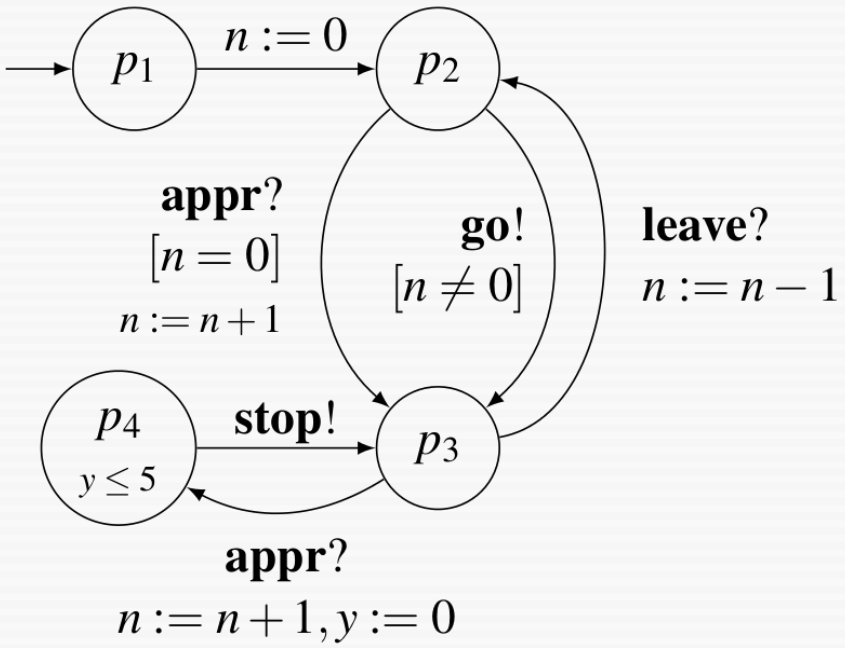
$$I_0(x, y) \equiv true$$
$$I_1(x, y) \equiv x \leq y$$
$$I_2(x, y) \equiv true$$

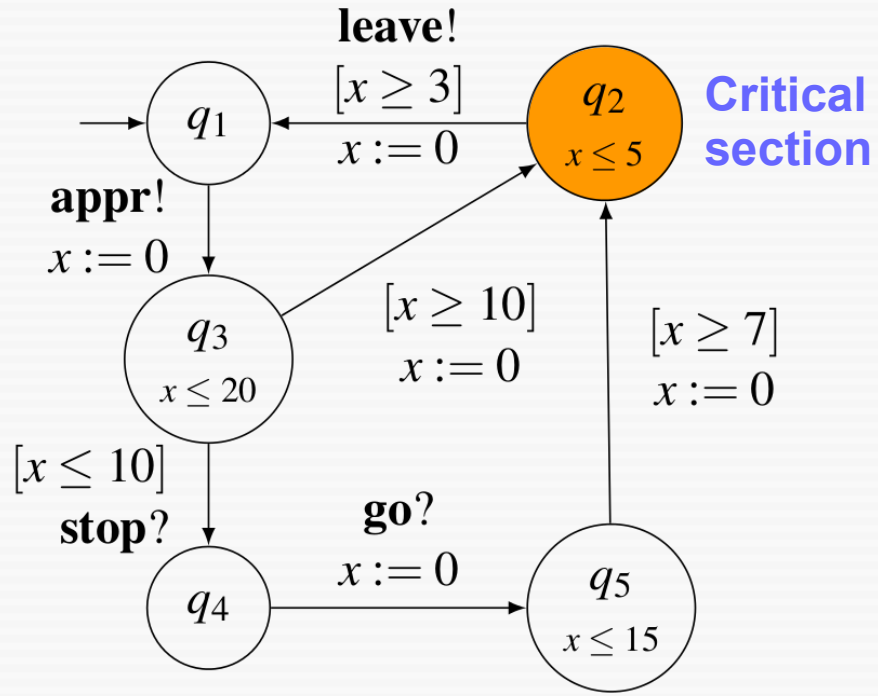
Outline

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

Timed systems





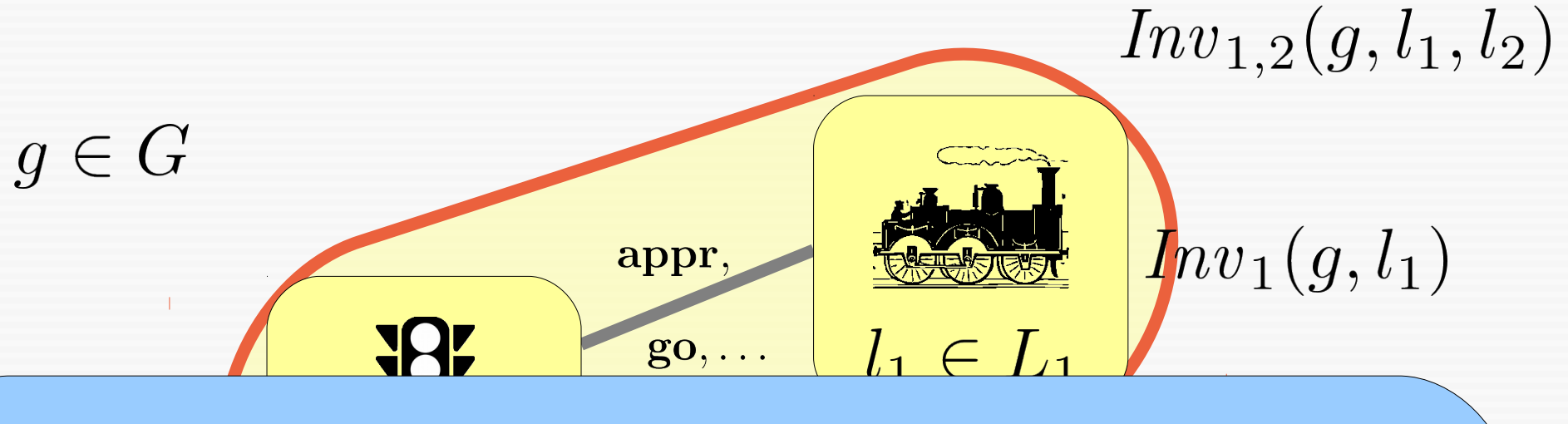


[FORTE'94]

Controller

Trains

Timed systems



Constraints:

Local transitions:

$$T_1(g, l_1, g', l_1') \wedge Inv_1(g, l_1) \rightarrow Inv_1(g', l_1')$$

Non-interference:

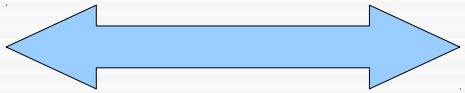
$$T_1(g, l_1, g', l_1') \wedge Inv_1(g, l_1) \wedge Inv_2(g, l_2) \rightarrow Inv_2(g', l_2)$$

+ 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



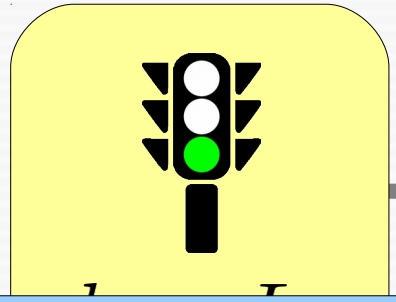
Monolithic
Single invariant
for whole system

Weaker
Smaller invariants

Stronger
Detailed invariants

Parameterised systems (link)

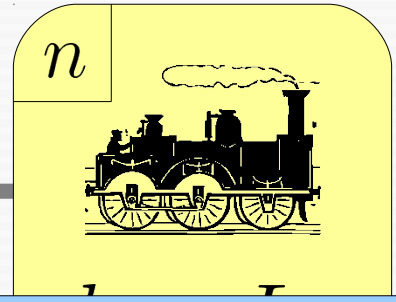
$g \in G$



$Inv_2(g, l_2)$

appr,

go, . . .



$Inv_1(g, n, l_1)$

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 $L_p \quad (\text{for each } p \in P)$
- Initial states $Init_p \subseteq G \times L_p$
- Transition relation $(g, l) \xrightarrow{p} (g', l') \quad (g \in G, l \in L_p)$

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

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

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

$$\frac{p \in P \qquad (g, \bar{l}[p]) \xrightarrow{p} (g', l')}{(g, \bar{l}) \rightarrow (g', \bar{l}[p/l'])}$$

- System error states

$$Err = \left\{(g, \bar{l}) \in S \mid \forall i.\ (g, \bar{l}[p_i]) \in E_i\right\} \subseteq S$$

- 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 $P = \{1, 2, 3\}$

$$A_1 = \left\{ \begin{matrix} (1,0,0), \\ (0,1,0), \\ (0,0,1) \end{matrix} \right\}$$

$$A_3 = \{(1,1,1)\}$$

$$A_2 = \{(1,1,0), (0,1,1)\}$$

Modular
Separate invariant
for each process



Monolithic
Single invariant
for whole system

Clauses *(finite case)*

- Initiation

$$\left\{ R_{\bar{a}}(\mathsf{g}, \mathsf{l}_1, \ldots, \mathsf{l}_k) \;\leftarrow\; Init_{i_1}(\mathsf{g}, \mathsf{l}_1) \wedge \cdots \wedge Init_{i_k}(\mathsf{g}, \mathsf{l}_k) \right\}_{\bar{a} \in A}$$

($i_1, \ldots, i_k$ non-zero entries in $\bar{a}$)

- Consecution

$$\left\{ R_{\bar{a}}(\mathsf{g}', \bar{\mathsf{l}}[p/\mathsf{l}'][\bar{a}]) \;\leftarrow\; \left((\mathsf{g}, \bar{\mathsf{l}}[p]) \xrightarrow{p} (\mathsf{g}', \mathsf{l}')\right) \wedge R_{\bar{a}}(\mathsf{g}, \bar{\mathsf{l}}[\bar{a}]) \wedge Ctxt(\{p\}, \mathsf{g}, \bar{\mathsf{l}}) \right\}_{\substack{p=1,\ldots,n \\ \bar{a} \in A}}$$

- Absence of errors $(\langle p_1, E_1 \rangle, \ldots, \langle p_m, E_m \rangle)$

$$false \;\leftarrow\; \left(\bigwedge_{j=1,\ldots,m} (\mathsf{g}, \bar{\mathsf{l}}[p_j]) \in E_j \right) \wedge Ctxt(\{p_1, \ldots, p_m\}, \mathsf{g}, \bar{\mathsf{l}})$$

- 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 $A = \{(k)\}$

$$\left\{ R(\mathsf{g}, \mathsf{p}_{\sigma(1)}, \mathsf{l}_{\sigma(1)}, \ldots, \mathsf{p}_{\sigma(k)}, \mathsf{l}_{\sigma(k)}) \leftarrow \mathit{dist}(\mathsf{p}_1, \ldots, \mathsf{p}_k) \wedge R(\mathsf{g}, \mathsf{p}_1, \mathsf{l}_1, \ldots, \mathsf{p}_k, \mathsf{l}_k) \right\}_{\sigma \in S_k} \quad (6)$$

$$R(\mathsf{g}, \mathsf{p}_1, \mathsf{l}_1, \ldots, \mathsf{p}_k, \mathsf{l}_k) \leftarrow \mathit{dist}(\mathsf{p}_1, \ldots, \mathsf{p}_k) \wedge \mathit{Init}(\mathsf{g}, \mathsf{l}_1) \wedge \cdots \wedge \mathit{Init}(\mathsf{g}, \mathsf{l}_k) \quad (7)$$

$$R(\mathsf{g}', \mathsf{p}_1, \mathsf{l}'_1, \ldots, \mathsf{p}_k, \mathsf{l}_k) \leftarrow \mathit{dist}(\mathsf{p}_1, \ldots, \mathsf{p}_k) \wedge \left((\mathsf{g}, \mathsf{l}_1) \xrightarrow{\mathsf{p}_1} (\mathsf{g}', \mathsf{l}'_1)\right) \wedge R(\mathsf{g}, \mathsf{p}_1, \mathsf{l}_1, \ldots, \mathsf{p}_k, \mathsf{l}_k) \quad (8)$$

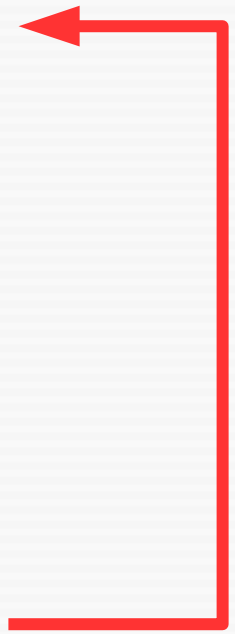
$$R(\mathsf{g}', \mathsf{p}_1, \mathsf{l}_1, \ldots, \mathsf{p}_k, \mathsf{l}_k) \leftarrow \mathit{dist}(\mathsf{p}_0, \mathsf{p}_1, \ldots, \mathsf{p}_k) \wedge \left((\mathsf{g}, \mathsf{l}_0) \xrightarrow{\mathsf{p}_0} (\mathsf{g}', \mathsf{l}'_0)\right) \wedge \mathit{RConj}(0, \ldots, k) \quad (9)$$

$$\mathit{false} \leftarrow \mathit{dist}(\mathsf{p}_1, \ldots, \mathsf{p}_r) \wedge \left(\bigwedge_{j=1,\ldots,m} (\mathsf{p}_j = p_j \wedge (\mathsf{g}, \mathsf{l}_j) \in E_j) \right) \wedge \mathit{RConj}(1, \ldots, r) \quad (10)$$

$$\mathit{RConj}(a, \ldots, b) = \bigwedge_{\substack{i_1, \ldots, i_k \in \{a, \ldots, b\} \\ i_1 < i_2 < \cdots < i_k}} R(\mathsf{g}, \mathsf{p}_{i_1}, \mathsf{l}_{i_1}, \ldots, \mathsf{p}_{i_k}, \mathsf{l}_{i_k})$$

CEX-Guided Refinement of Schemata

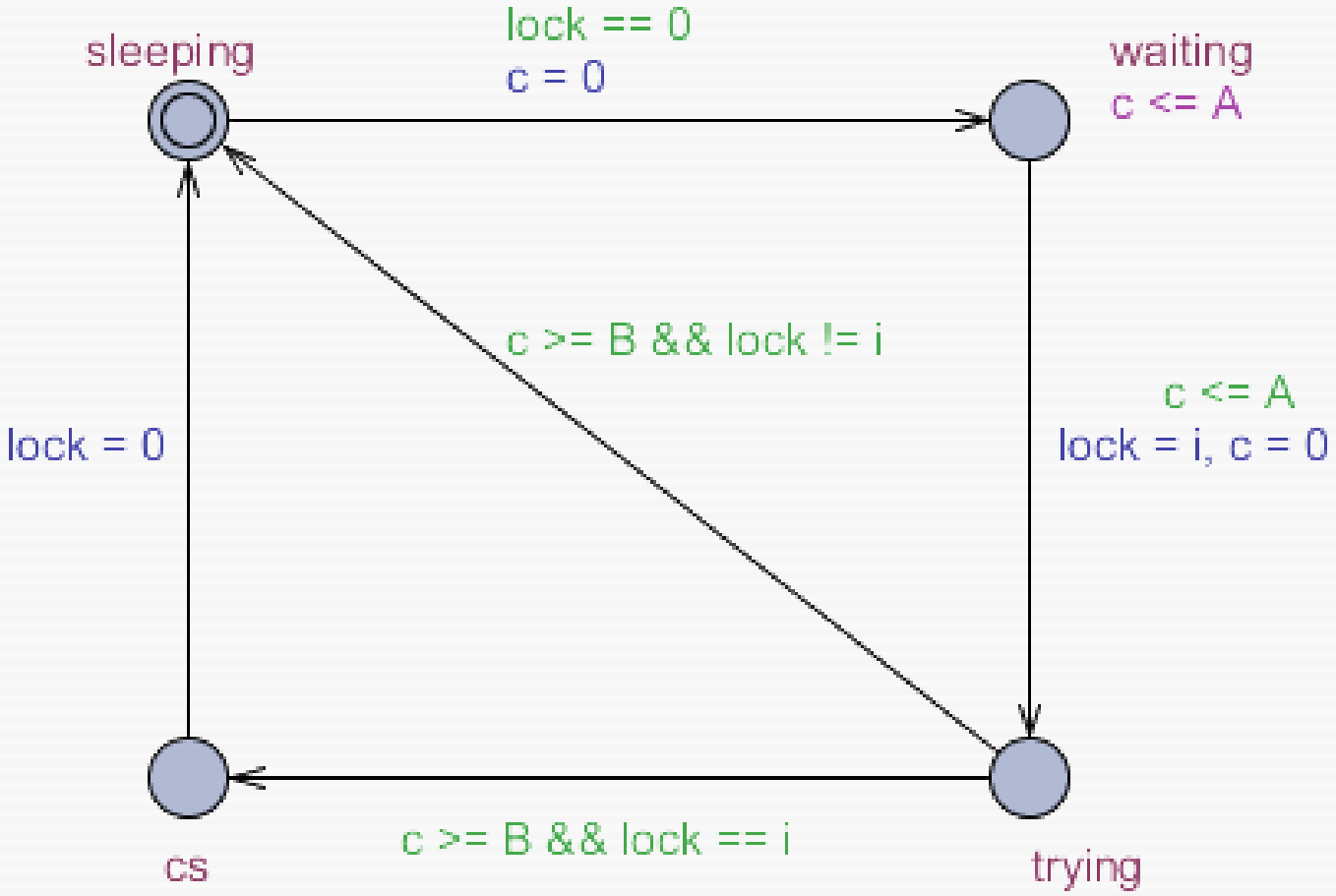
- Start with weakest (fully modular) invariant schema A_0
- If Horn constraints are not solvable, check whether CEX is genuine
  - ***Yes*** $\rightarrow$ System CEX
  - ***No*** $\rightarrow$ Choose stronger schema A_{i+1}
- Example: trains model



Extensions of the Basic Model

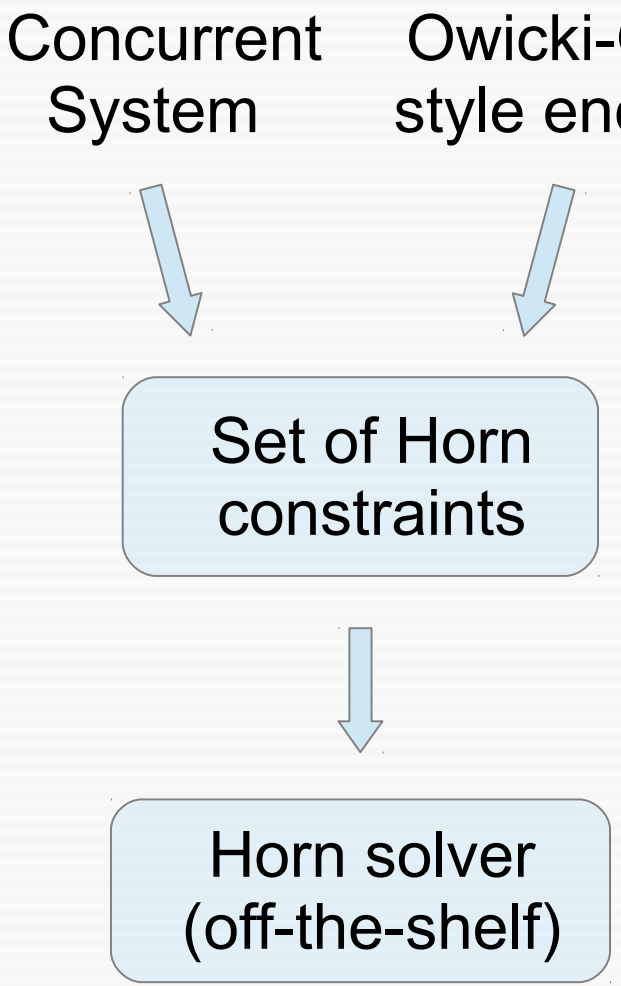
- **Physical time** (discrete, dense)
  - Add a global clock (global variable)
  - Delay transitions
- **Communication channels** (UPPAAL-style)
  - Clauses encoding simultaneous transition of two processes
- **Barriers**
- **BIP**-style interactions

Protocol X (link)



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

On Completeness



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

Usually **incomplete**
→ “Best-effort”

Experiments

Benchmark	$\sharp Cl_i$	$\sharp Cl_f$	N^{th} *(sec)*	*Inv Schema*	*Total (sec)*
Temperature Control System (unsafe)	48	110	0.37	(1,1,1)	3.86
Temperature Control System	48	110	1.12	(1,1,1)	4.31
Fischer	47	221	5.62	(2,1)	12.21
Fischer (unsafe)	47	221	2.84	(2,1)	8.91
CSMA/CD	50	162	3.60	(2,1)	8.22
CSMA/CD (unsafe)	50	793	2.91	(4,1)	13.81
Lynch-Shavit	50	299	66.11	(2)	70.10
Lynch-Shavit (unsafe)	50	299	2.58	(2)	5.35
Train Crossing	28	686	2.51	(1,3)	8.84
Train Crossing (unsafe)	28	240	1.53	(1,2)	3.91

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 $C \wedge B_1, \ldots, B_n \rightarrow H$ 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 $false$, 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.