Towards Bounded Model Checking for Timed and Hybrid Automata with a Quantified Encoding

Luan Viet Nguyen

Adviser: Taylor T. Johnson

VeriVITAL - The ***Veri***fication and ***V***alidation for ***I***ntelligent and ***T***rustworthy ***A***utonomy ***L***aboratory (http://verivital.uta.edu/)

Department of Computer Science and Engineering

FMCAD Student Forum, September 28, 2015

UNIVERSITY OF TEXAS ARLINGTON

Hybrid Automata

An execution of Hybrid Automata H is a sequence:

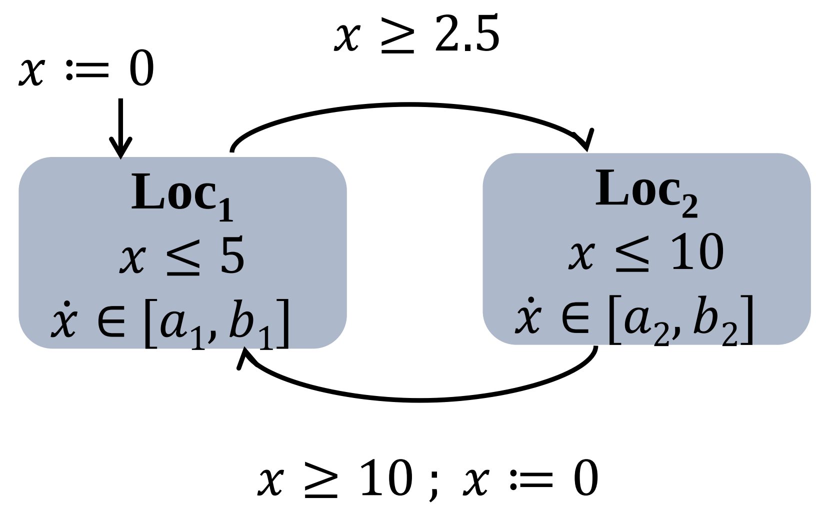
$$\pi = s_0 \to s_1 \to s_2 \to \ldots$$

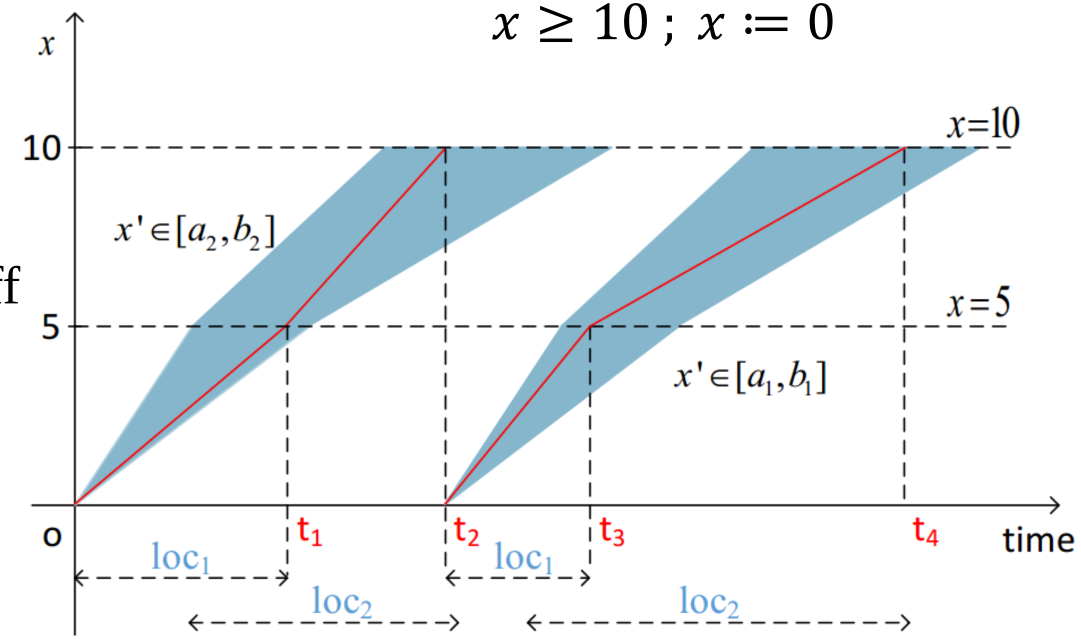
$s_0 \in$ a set of initial states

$s_i \to s_{i+1}$: a discrete transition or a continuous trajectory

s_k is reachable from initial sate s_0 iff there exists:

$$\pi = s_0 \to s_1 \to \ldots \to s_{k-1} \to s_k$$





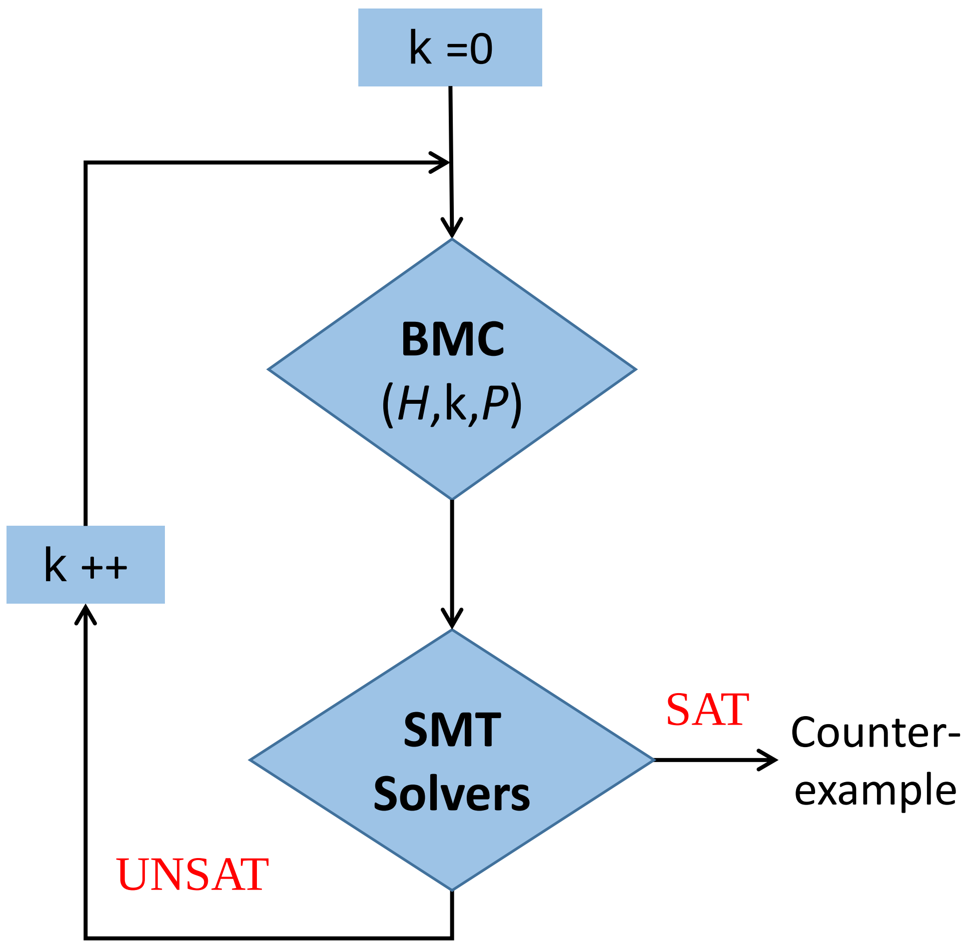
BMC for Hybrid Automata

Quantifier-Free BMC for Hybrid Automata

- dRreach uses the dReal SMT solver
- HyComp built on top of nuXmv that uses the MathSAT SMT solver
- Other reachability tools: Uppaal, HyTech, SpaceEx ,Flow*, etc.

Quantified BMC for Hybrid Automata

- New encoding in our work
- Builds on BMC for discrete systems using QBF solvers instead of SAT solvers



Quantified BMC (QBMC) for Hybrid Automata

Quantifier-Free BMC formula:

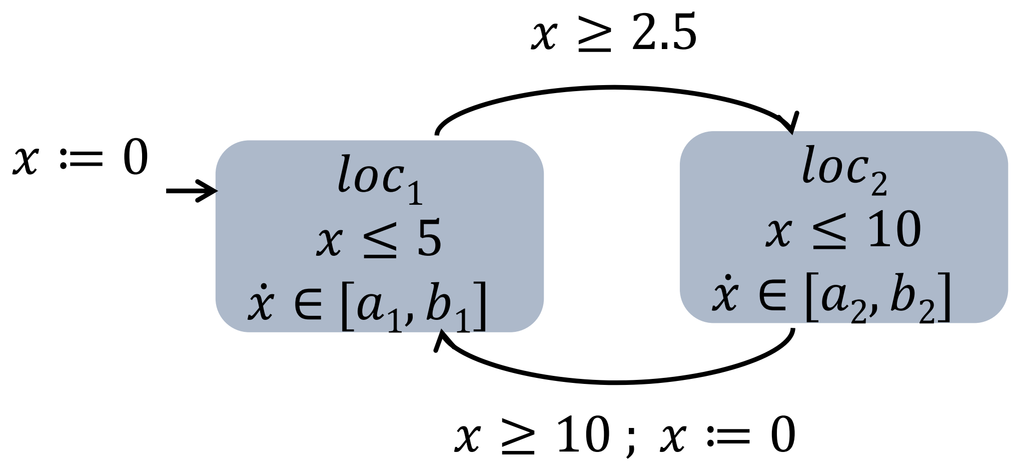
$$\Phi(\mathrm{k}) \triangleq I(\mathrm{V}_0) \wedge \bigwedge_{i=0}^{k-1} T_i(\mathrm{V}, \mathrm{V}') \wedge (\bigvee_{i=0}^{k} P(\mathrm{V}_i))$$

Quantified BMC formula:

$$\Omega(\mathrm{k}) \triangleq \exists \mathrm{V}_0, \mathrm{V}_1, \ldots, \mathrm{V}_\mathrm{k}, \delta \forall t \exists \mathrm{V}, \mathrm{V}' \mid I(\mathrm{V}_0) \wedge T(\mathrm{V}, \mathrm{V}') \wedge \bigwedge_{i=0}^{k-1} t_{i+1} \longrightarrow [(\mathrm{V} = \mathrm{V}_i) \wedge (\mathrm{V}' = \mathrm{V}_{i+1})] \wedge (\bigvee_{i=0}^{k} P(\mathrm{V}_i))$$

- $I(\mathrm{V}_0)$: an initial set of states
- $T_i(\mathrm{V}, \mathrm{V}') \triangleq D_i(\mathrm{V}, \mathrm{V}') \vee \mathcal{T}_i(\mathrm{V}, \mathrm{V}')$: a transition (discrete or continuous trajectory) between consecutive pairs of sets of states
- $P(\mathrm{V}_i)$: a safety specification at iteration i
- δ: a real time elapse in the trajectories
- $t = \langle t_1, t_2, \ldots, t_{\lceil log_2 k \rceil} \rangle$: a boolean vector to index each iteration of the BMC of hybrid automata

Example



Quantifier-free BMC formula up to k = 2:

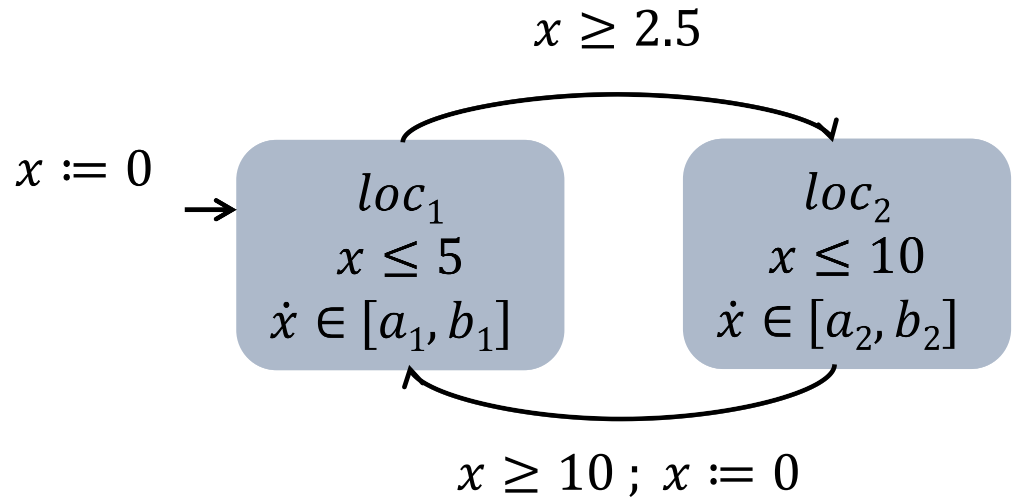
$$\Phi(2) \triangleq I_0 \wedge (D_0 \wedge \mathcal{T}_0) \wedge (D_1 \wedge \mathcal{T}_1) \wedge (P(\mathrm{V}_0) \vee P(\mathrm{V}_1) \vee P(\mathrm{V}_2))$$

- $k = 0$: $I_0 := (l_0 = loc_1 \wedge x_0 = 0)$

- $k = 1$ (D_0): $(l_0 = loc_1 \wedge l_1 = loc_2 \wedge x_0 \leq 5 \wedge x_0 \geq 2.5 \wedge x_1 = x_0)$,
- $k = 1$ $(\mathcal{T}_0)$: $((l_0 = loc_1 \rightarrow (l_1 = l_0 \wedge x_0 + a_1\delta \leq x_1 \wedge x_1 \leq x_0 + b_1\delta \wedge x_1 \leq 5))$

- $k = 2$ (D_1): $(l_1 = loc_1 \wedge l_2 = loc_2 \wedge x_1 \leq 5 \wedge x_1 \geq 2.5 \wedge x_2 = x_1)$,
- $k = 2$ $(\mathcal{T}_1)$: $(l_1 = loc_1 \rightarrow (l_2 = l_1 \wedge x_1 + a_1\delta \leq x_2 \wedge x_2 \leq x_1 + b_1\delta \wedge x_2 \leq 5)$

Example

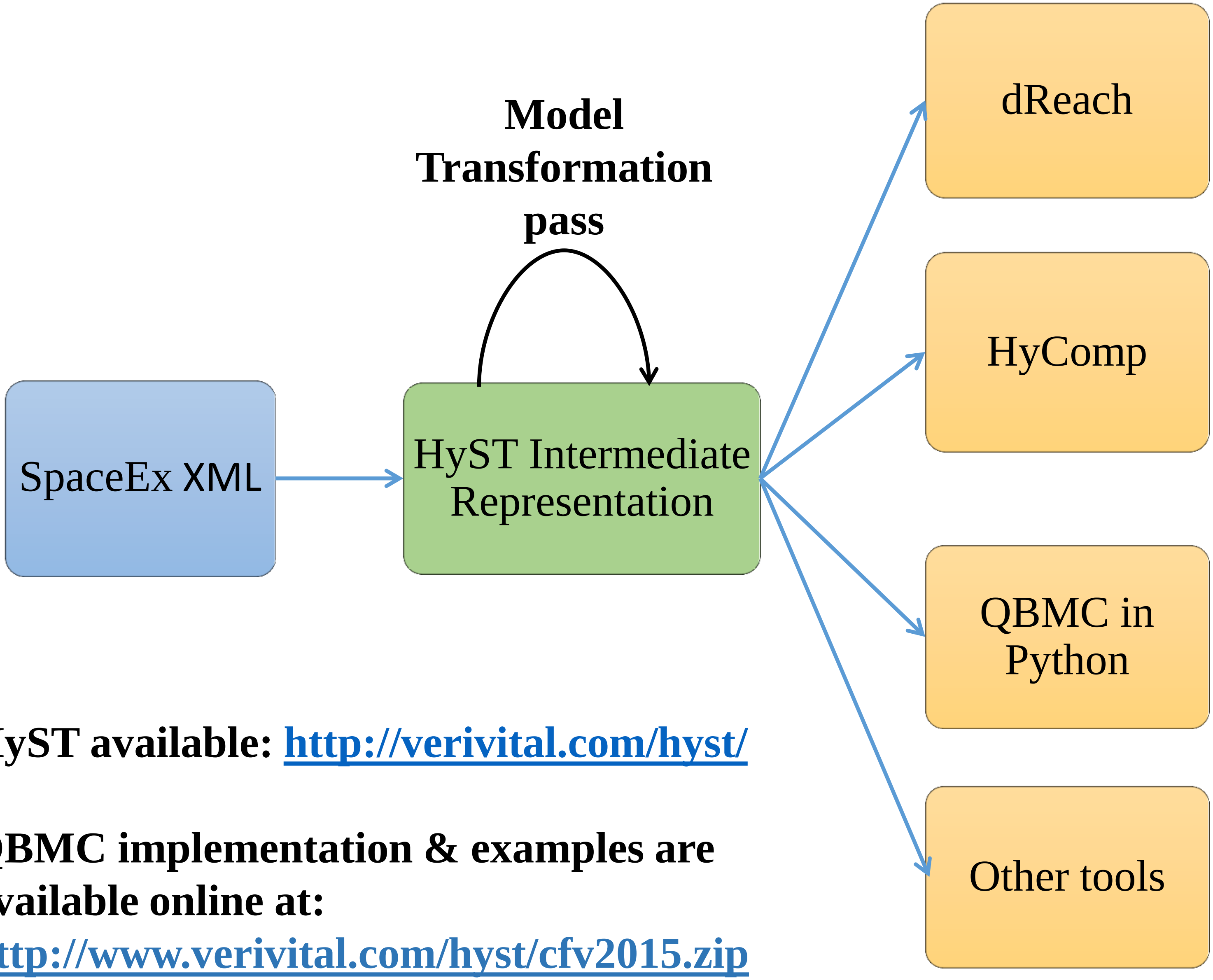


Quantified BMC
formula up to k = 2:

$$\begin{aligned}\Omega(2) \triangleq \exists \mathrm{V}_0, \mathrm{V}_1, \mathrm{V}_2, \delta \forall t_1 \exists \mathrm{V}, \mathrm{V}' \mid\ & I(\mathrm{V}_0) \wedge T(\mathrm{V}, \mathrm{V}') \\ & \wedge \{\neg t_1 \rightarrow [(\mathrm{V} = \mathrm{V}_0) \wedge (\mathrm{V}' = \mathrm{V}_1)]\} \\ & \wedge \{t_1 \rightarrow [(\mathrm{V} = \mathrm{V}_1) \wedge (\mathrm{V}' = \mathrm{V}_2)]\} \\ & \wedge \{P(\mathrm{V}_0) \vee P(\mathrm{V}_1) \vee P(\mathrm{V}_2)\}\end{aligned}$$

- $k = 0$: $I_0 := (l_0 = loc_1 \wedge x_0 = 0)$
- $k = 1$: $\neg t_1 \rightarrow (l_0 = loc_1 \rightarrow (l_1 = l_0 \wedge x_0 + a_1\delta \leq x_1 \wedge x_1 \leq x_0 + b_1\delta \wedge x_1 \leq 5))$
- $k = 2$: $t_1 \rightarrow (l_1 = loc_1 \wedge l_2 = loc_2 \wedge x_1 \leq 5 \wedge x_1 \geq 2.5 \wedge x_2 = x_1)$

Implementation in HyST



HyST available: http://verivital.com/hyst/

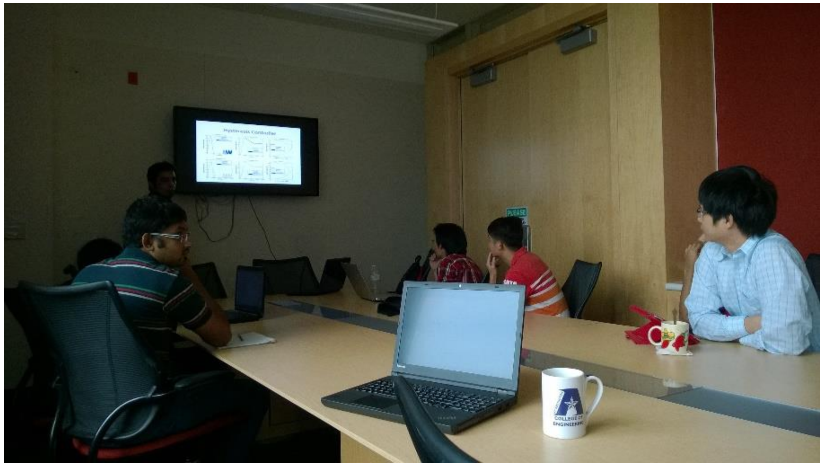
QBMC implementation & examples are available online at: http://www.verivital.com/hyst/cfv2015.zip

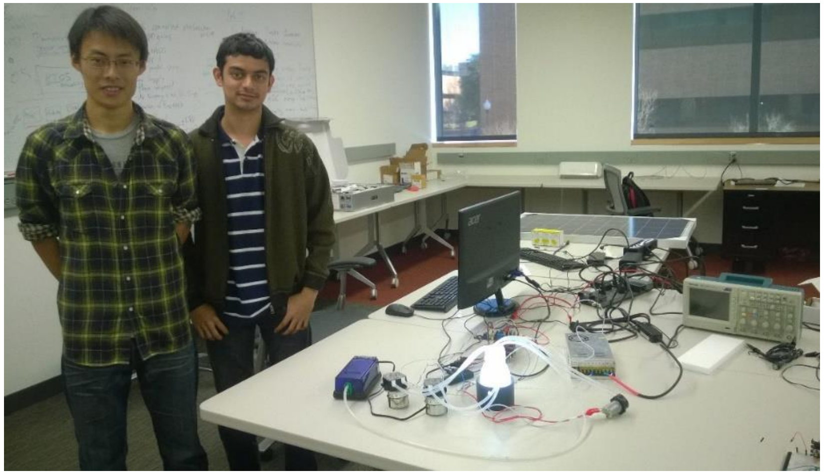
Conclusion

- QBMC, a new SMT-based technique that encodes, in a quantified form, the BMC problem for rectangular hybrid automata
  - Encompasses problem for timed automata
- QBMC can solve the BMC problem for hybrid systems such as Fischer and Lynch-Shavit mutual exclusion protocols including more than a thousand locations
  - Requires less memory usage compared to dReach and HyComp
- Follow-up paper with more details:

  Luan Viet Nguyen, Djordje Maksimovic, Taylor T. Johnson, Andreas Veneris, "*Quantified Bounded Model Checking for Rectangular Hybrid Automata*", In 9th International Workshop on Constraints in Formal Verification (CFV 2015), Austin, Texas, November 2015 (To appear)
- Future work:
  - conduct additional experiments and compare the results to other tools and techniques, such as UPPAAL
  - investigate more general classes of hybrid automata, such as those with linear or polynomial differential equations

Thank You! Questions?





- **Acknowledgements**
  - Follow-up paper “Quantified Bounded Model Checking for Rectangular Hybrid Automata” with Andreas Veneris and Djordje Maksimovic of U. Toronto to appear at ICCAD 2015 workshop CFV
  - **UTA CSE:** Luan Viet Nguyen (PhD), Hoang-Dung Tran (PhD), Mousa Almotairi (PhD), Nathan Hervey (MSc)
  - **UTA EE:** Omar Beg (PhD)

Luan Viet Nguyen

http://verivital.uta.edu

UNIVERSITY OF TEXAS ARLINGTON

Extra Slides

Boolean Satisfiability (SAT)

Given a Boolean Formula in Conjunctive Normal Form (CNF)

Is there an assignment to Boolean variables that makes the formula True?

Example: $\Omega(x_1, x_2, x_3) \triangleq (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3)$

$\downarrow$

$A \triangleq \{x_1 = 1, x_2 = 0, x_3 = 1\}$ is SAT assignment

SAT solver: tool to find a SAT assignment

Satisfiability modulo theories (SMT) : generalization of SAT with respect to combinations of background theories

SAT-Based Bounded Model Checking(BMC)

The reachable states in *k* steps are captured by:

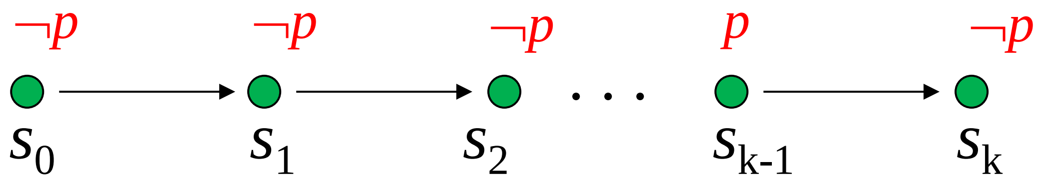
$$\Omega(\mathrm{k}) \triangleq I(\mathrm{s}_0) \wedge \bigwedge_{i=0}^{k-1} T(\mathrm{si}, \mathrm{si}_{+1)} \wedge (\bigvee_{i=0}^{k} P(\mathrm{si}))$$

initial states

transition relation

safety specification

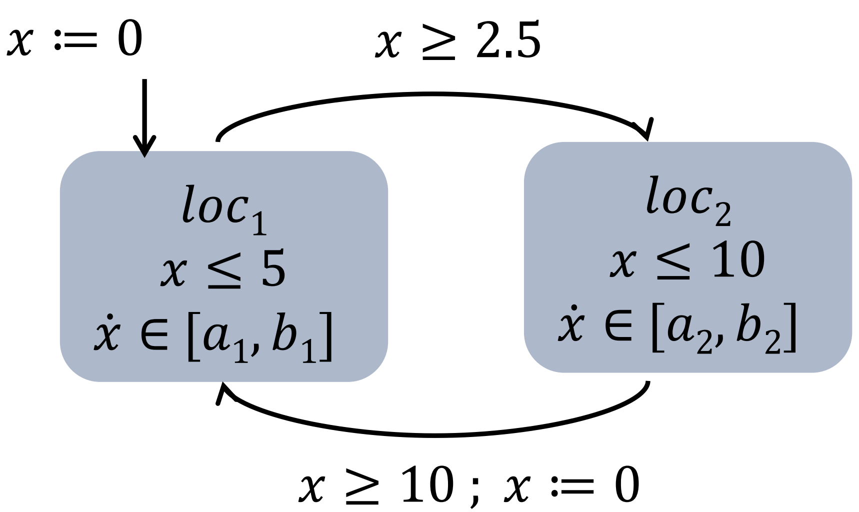
The safety property *p* is satisfied up to step *k* iff Ω (k) is unsatisfiable:



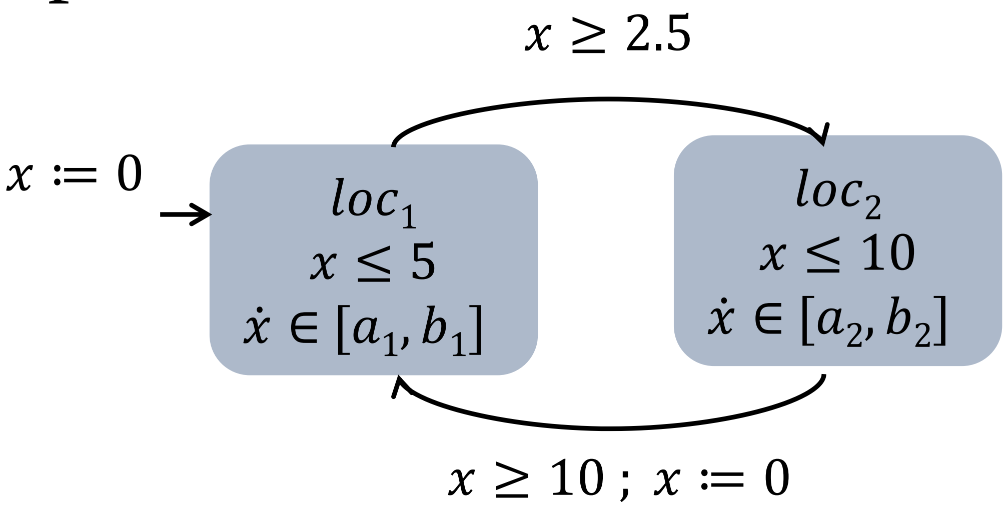
Hybrid Automata

A Hybrid Automaton *H* = <*Loc*, *Var*, *Inv*, *Flow*, *Trans*, *Init*>

- *Loc:* a finite set of discrete locations
- *Var*: a finite set of n continuous, real-valued variables
- *Inv*: a finite set of invariants
- *Flow*: a finite set of ordinary differential inclusions
- *Trans*: a finite set of transitions between locations
  - Guard : the condition enables the transition from a source location to a target location
  - Update: the update map of variables for each transition
- *Init:* a finite set of initial states



Example

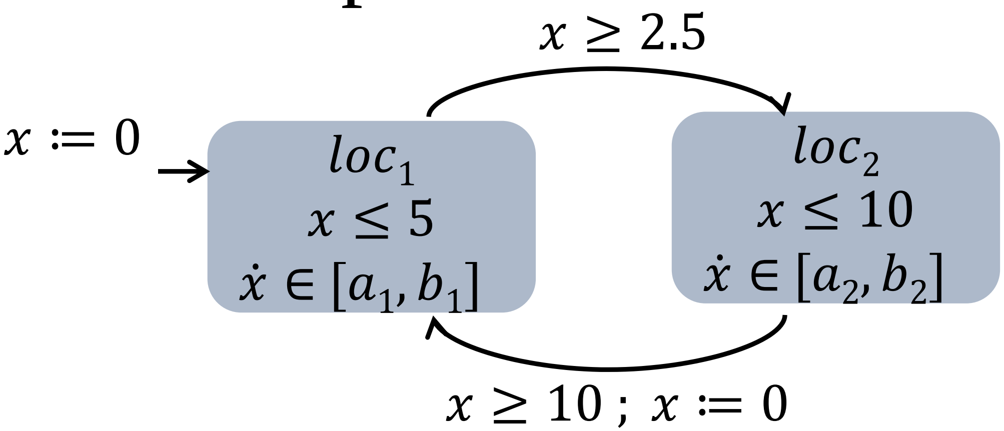


- Timed automata: $a_1 = b_1 = a_2 = b_2$
- Multirate-timed automata: $a_1 = b_1$ and $a_2 = b_2$, but possibly $a_1 \neq a_2$
- Rectangular hybrid automata: Otherwise

Bad States:

$$P \triangleq \bigvee_{i=0}^{k} \neg(loc_i = loc_1 \rightarrow x \geq 2.5)$$

Example

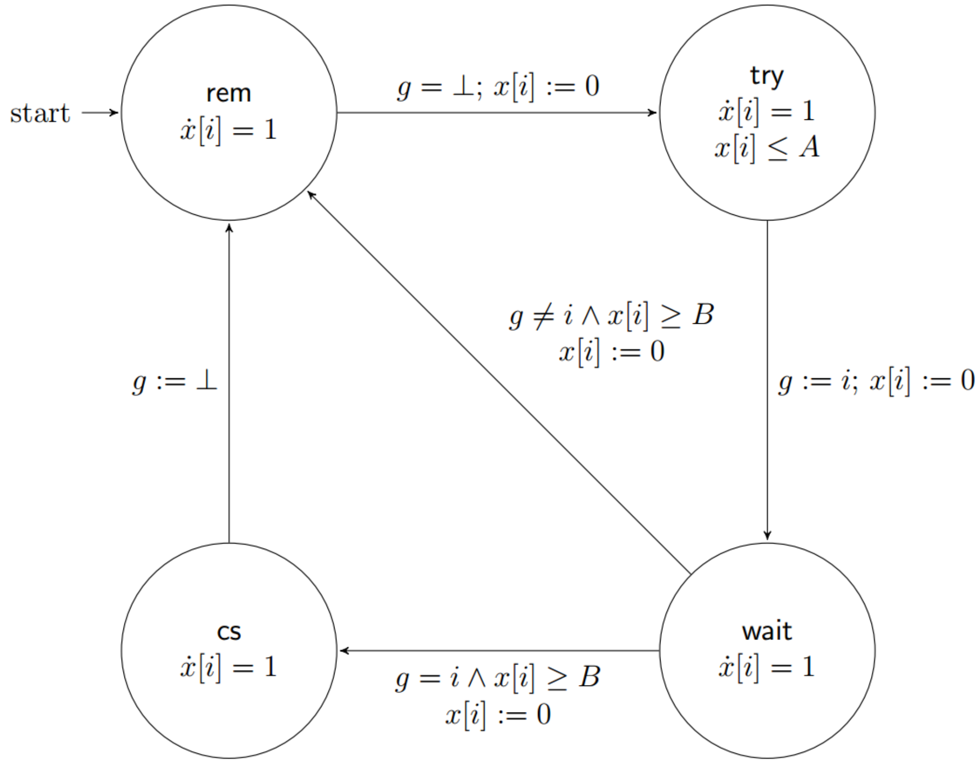


$$P \triangleq \bigvee_{i=0}^{k} \neg(loc_i = loc_1 \rightarrow x \geq 2.5)$$

Rectangular hybrid automata: $a_1 = 0$, $b_1 = 1$, $a_2 = 0$, $b_2 = 2$

Tools	L	$k \leq 32$		$k \leq 64$		$k \leq 128$	
		Time (sec)	Mem (MB)	Time (sec)	Mem (MB)	Time (sec)	Mem (MB)
QBMC	2	1.11	27.2	3.68	39.4	19.9	91.2
dReach	2	86.7	102.4	1176.4	284.7	20034	829.2
HyComp	2	0.4	97.3	0.6	101.8	1.44	109.3

Fischer Mutual Exclusion Protocol



- Safe version $A \leq B$
- Unsafe version: $A > B$
- Discrete locations: 4^N
- Discrete state-spaces: $(N + 1)(4N)^N$

Number of processes

Bad States:

$$\phi \triangleq \neg \forall i, j \in \{1, \ldots, N\} \mid (i \neq j \wedge q_i = cs) \rightarrow q_j \neq cs$$

Fischer Mutual Exclusion Protocol

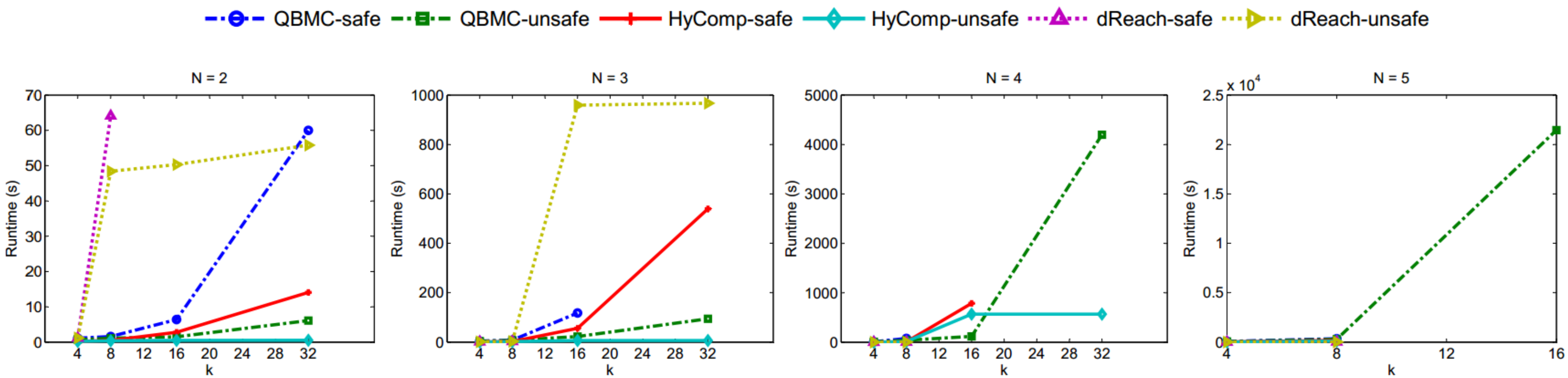


Fig. 4. Runtime comparison of HyComp, dReach and QBMC in solving the BMC of Fischer protocol.

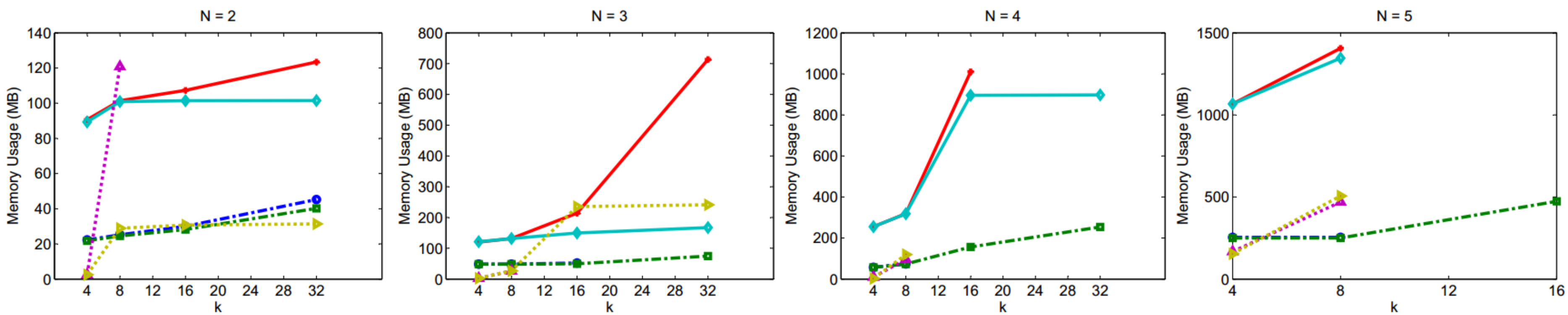
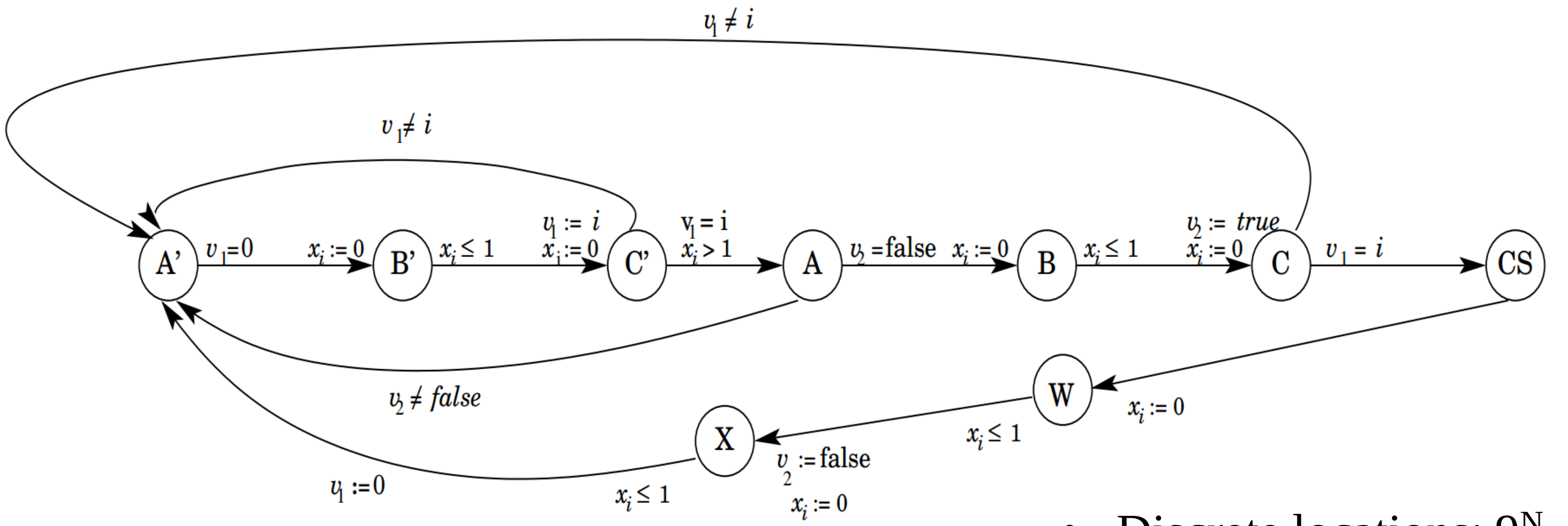


Fig. 5. Memory usage comparison of HyComp, dReach and QBMC in solving the BMC of Fischer protocol.

Lynch-Shavit Mutual Exclusion Protocol



- Discrete locations: 9^N
- Discrete state-spaces: $(N + 1)(9N)^N$

Bad States:

$$\phi \triangleq \neg\forall i, j \in \{1, \ldots, N\} \mid (i \neq j \wedge q_i = cs) \rightarrow q_j \neq cs$$

Mahata, Pritha. "Model checking parameterized timed systems." (2005)

Lynch-Shavit Mutual Exclusion Protocol

Tools	L	$k \leq 4$		$k \leq 8$		$k \leq 16$	
		Time (sec)	Mem (MB)	Time (sec)	Mem (MB)	Time (sec)	Mem (MB)
QBMC	9^2	3.7	52.2	5.1	52.3	25.9	52.7
	9^3	15.5	65.6	31.3	87.5	1091.5	144.5
	9^4	256.1	702.8	1062.1	708.9	43578	1196.2
HyComp	9^2	0.8	121.9	1.33	132.8	9.5	170.5
	9^3	2.7	307.9	12.81	380.8	192.8	771.4
	9^4	63.9	2655.4	N/A	M/O	N/A	M/O