Unbounded, Fully Symbolic Model Checking of Timed Automata using Boolean Methods

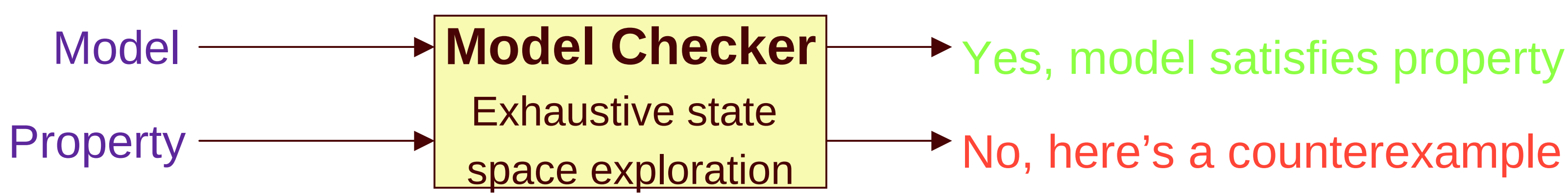
Sanjit A. Seshia and Randal E. Bryant

Computer Science Department

Carnegie Mellon University

Verifying Timed Embedded Systems

- **Many embedded systems are real-time**
  - **E.g., drive-by-wire systems in automobiles**
- **Confidence in system reliability is increased by verification of system models**
- ***Model Checking* has been successfully used for verifying finite-state models**

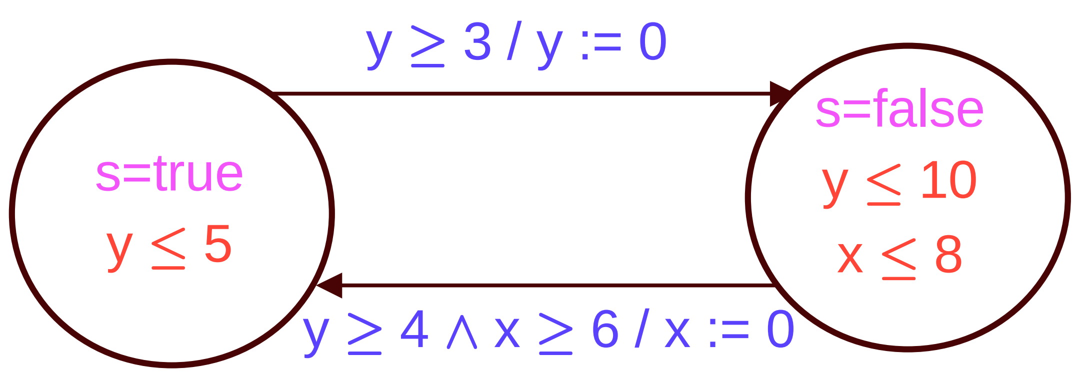


- **However, the same level of success has eluded model checking of real-time models**
  - **State space contains both continuous and discrete parts**
  - **Hard to find a compact representation that combines both parts**

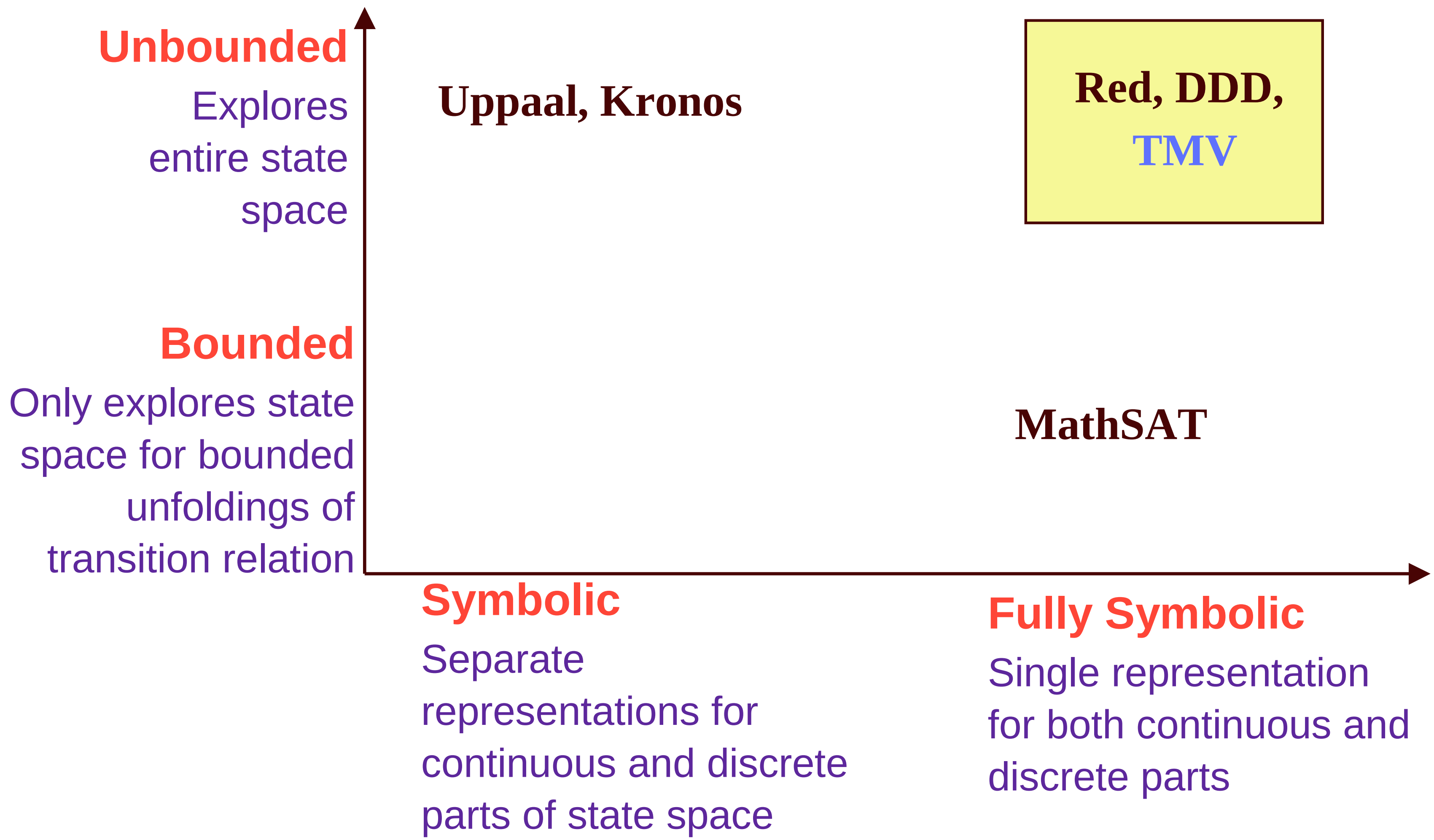
Timed Automata

Alur, Courcobetis, & Dill, ‘90

- **A modeling formalism for timed systems**
- **Generalization of finite automaton with:**
  - **Non-negative real-valued clock variables**
  - **Constraints on clocks as guards on states and transitions**



Timed Model Checking Taxonomy



Unbounded, Fully Symbolic Model Checking

Henzinger, Nicollin, Sifakis, Yovine ‘94

- **Set of states represented as a formula ϕ in *separation logic* (SL)**
  - **Boolean Combinations** ($\wedge$, $\vee$, $\neg$) **of**
    - **Boolean variables: e_i**
    - **Separation Predicates: $x_i \geq x_j + c$, $x_i > x_j + c$**
      - » **Also called “difference-bound” or “gap-order” constraints**
    - ➢ 0 represented as special variable x_0
- **Properties are in Timed CTL***
  - **Two kinds of TCTL* formulas:**
    - **Reachability properties: Safety and bounded liveness**
      - » E.g. AG (file requested ➔ $AF_{\leq 5}$ (file received))
    - **Non-reachability properties: Unbounded liveness**
      - » E.g. EG . z:= 0 . F (z = 1) [non-zenoness]

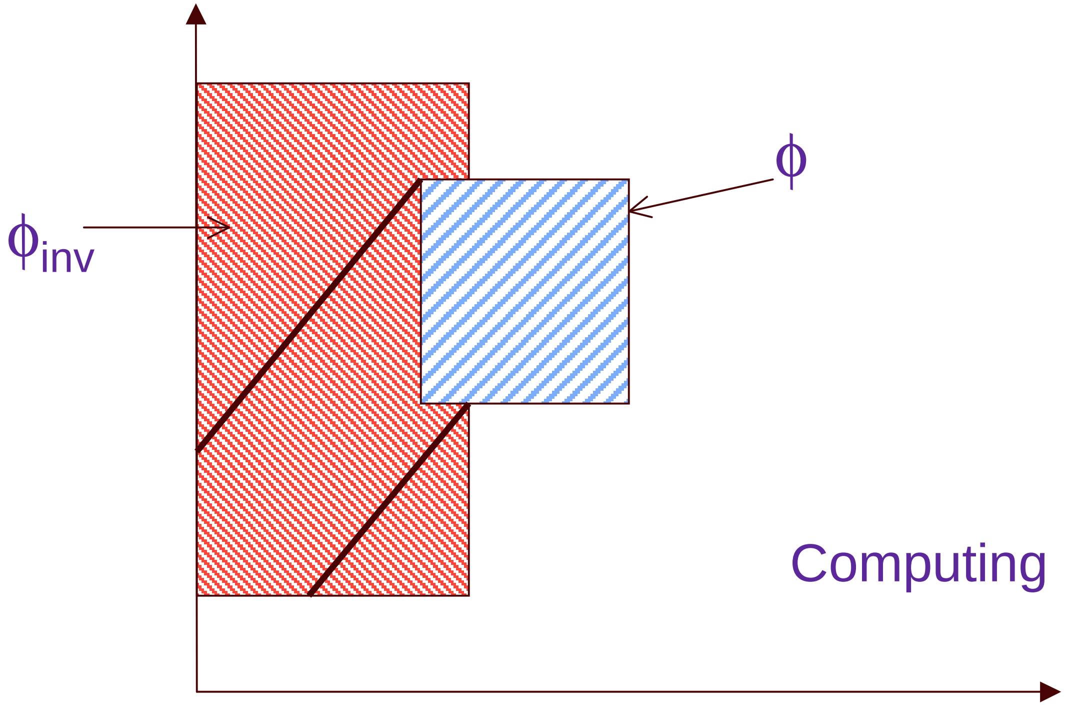
Pre Operator for Model Checking

- **Two ways to reach a set of states ϕ:**
  - **Let time elapse**
    - Only clock variables change, discrete variables remain unchanged
  - **Make a discrete transition**
    - Some clock variables reset, all others unchanged
    - Discrete state changes as per transition relation
- **Pre Operator can be written as**

  $pre(\phi) \triangleq pre_d(\phi) \vee pre_t(\phi)$

  - $pre_d(\phi)$ is the same as in Boolean model checking
  - $pre_t(\phi)$ is expressed in Quantified Separation Logic (QSL)

Timed Pre Operator in QSL



Computing $pre_t(\phi)$

- $pre_t(\phi) \triangleq \exists \delta \{ \delta \leq x_0 \wedge \phi [\delta / x_0] \wedge \forall \varepsilon (\delta \leq \varepsilon \leq x_0 \wedge \phi_{inv}[\varepsilon / x_0]) \}$
  - ϕ_{inv} is the conjunction of all state guards
- **Need quantifier elimination procedure for QSL**

QSL Quantifier Elimination

- **Start with QSL formula ω, where $\omega \triangleq \exists x_a . \phi$**
  - To handle $\forall x_a . \phi$, start with $\exists x_a . \neg \phi$, and negate the result
- **Quantifier elimination done in 3 phases:**
  1. Translate ω to another QSL formula ω' where:
     - ω' has quantifiers only over Boolean variables
     - ω is equivalent to ω'
  2. Encode ω' as a QBL formula and eliminate Boolean quantifiers
  3. Translate the result back to SL
- **Benefit of this method**
  - Unlike other methods, avoids translation to DNF

Quantifier Elimination Phase 1

Input $\omega \triangleq \exists x_3 . (x_1 \geq x_3 \vee x_3 \geq x_1+2) \wedge x_0 \geq x_3-5 \wedge x_3 \geq x_2$

Boolean encoding ϕ_{bool}

$(e_{1,3}^{\geq,0} \vee e_{3,1}^{\geq,2}) \wedge e_{0,3}^{\geq,-5} \wedge e_{3,2}^{\geq,0}$

Transitivity constraints ϕ_{cons}

$(e_{1,3}^{\geq,0} \wedge e_{3,2}^{\geq,0}) \Rightarrow (x_1 \geq x_2)$

$\wedge (e_{3,1}^{\geq,2} \wedge e_{0,3}^{\geq,-5}) \Rightarrow (x_0 \geq x_1-3)$

$\wedge (e_{0,3}^{\geq,-5} \wedge e_{3,2}^{\geq,0}) \Rightarrow (x_0 \geq x_2-5)$

Generate QSL formula ω'

$\exists e_{1,3}^{\geq,0}, e_{3,1}^{\geq,2}, e_{0,3}^{\geq,-5}, e_{3,2}^{\geq,0}. \; [\phi_{bool} \wedge \phi_{cons}]$

Quantifier Elimination Phase 2 & 3

Generate QBL formula ρ from ω'

$\exists\, e_{1,3}^{\geq,0}, e_{3,1}^{\geq,2}, e_{0,3}^{\geq,-5}, e_{3,2}^{\geq,0}.$

$[\phi_{bool} \wedge (e_{1,3}^{\geq,0} \wedge e_{3,2}^{\geq,0} \Rightarrow \underline{e_{1,2}^{\geq,0}}) \wedge$

$(e_{3,1}^{\geq,2} \wedge e_{0,3}^{\geq,-5} \Rightarrow \underline{e_{0,1}^{\geq,-3}}) \wedge (e_{0,3}^{\geq,-5} \wedge e_{3,2}^{\geq,0}) \Rightarrow \underline{e_{0,2}^{\geq,-5}}]$

Eliminating quantifiers from ρ yields

$(e_{1,2}^{\geq,0} \wedge e_{0,2}^{\geq,-5}) \vee (e_{0,1}^{\geq,-3} \wedge e_{0,2}^{\geq,-5})$

Translating back to separation logic

$(x_1 \geq x_2 \wedge x_0 \geq x_2-5) \vee (x_0 \geq x_1-3 \wedge x_0 \geq x_2-5)$

Special Class of QSL formulas

- **Consider QSL formulas of the form:**

  $$\exists \varepsilon . \{ \varepsilon \leq x_0 \land \phi [\varepsilon / x_0] \}$$

  - **Recall that x_0 stands for 0**
- **We can do quantifier elimination more efficiently, generating fewer quantified Boolean variables**
- **Can similarly handle $\exists \varepsilon . \{ \varepsilon \geq x_0 \land \phi [\varepsilon / x_0] \}$**
- **Half of all quantifier elimination operations**
  - **Experimentally, leads to 10X-20X speedup**

Preliminary Results

- Fischer's timed mutual exclusion protocol, for increasing numbers of processes
- Results for non-reachability formula (non-zenoness)
  - Timed Model Verifier (TMV): Our model checker
    - Uses a BDD package (CUDD) as a QBL solver
  - Kronos & Red are the only other model checkers that can handle non-reachability properties

Number of Processes	Kronos Time (sec.)	Red Time (sec.)	TMV Time (sec.)	TMV (peak nodes)
3	0.03	0.28	0.24	28
4	0.23	1.30	0.44	39
5	1.98	5.05	0.80	54
6	*	17.80	2.15	69
7	*	57.95	6.61	88

Publications & Future Work

- Work will appear at CAV 2003
  - Details in technical report CMU-CS-03-117
- Ongoing & Future Work:
  - Using a SAT-based QBL solver
  - Improving current BDD-based implementation
  - Applications to real-world benchmarks
  - Investigating other applications
    - Convergence checking for bounded model checking of timed automata
    - Theorem proving
    - Hybrid systems