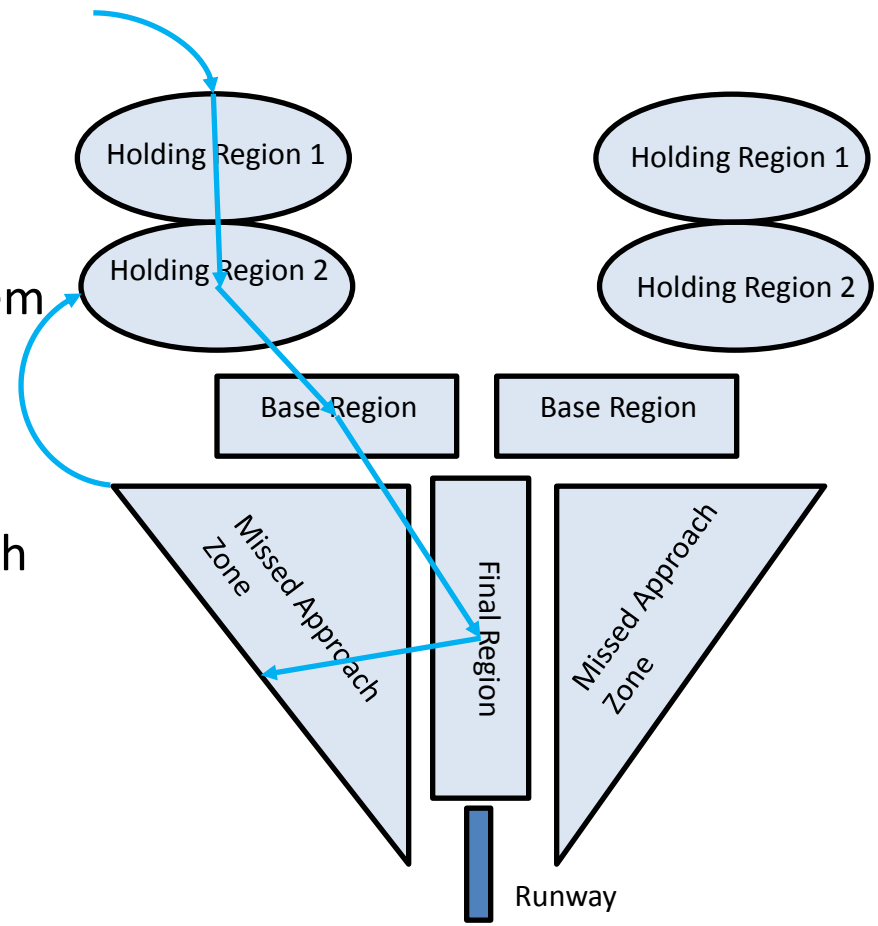
A Small Model Theorem for Rectangular Hybrid Automata Networks

Taylor T. Johnson and Sayan Mitra
University of Illinois at Urbana-Champaign

FMOODS/FORTE 2012

Example: Distributed Air Traffic Control

- Goal: verify properties for distributed systems when **arbitrarily many agents participate**
  - Parameterization on the number of participating agents
- Example: Small Aircraft Transportation System (SATS) [Abbott et al. NASA Report 2002]
- Distributed traffic control for increasing general aviation access to small airports with minimal centralized infrastructure
- Features of the system
  - (Cyber) Location, sequence of aircraft
  - (Physical) Motion of aircraft
  - (Distributed) List-like ordering data-structure is spread across multiple aircraft



Parameterized Systems and Verification

- For every instantiation of a system, verify some property P
  - $\forall N \in \mathbb{N}. \mathcal{A}(N) \triangleq \mathcal{A}_1 \parallel \mathcal{A}_2 \parallel \cdots \parallel \mathcal{A}_N \vDash P(\mathrm{N})$
  - Safety properties
    - $\mathrm{P(N)}$: $\forall i, j \in [N]. i \neq j \Rightarrow x_i - x_j \geq L_S$
      No two aircraft ever collide
    - $\mathrm{P(N)}$: $\forall i, j \in [N]. i \neq j \wedge q_i = cs \Rightarrow q_j \neq cs$
      No two processes are in a critical section simultaneously

- Example systems
  - Adaptive cruise control
  - Swarms of mobile robots (platooning, flocking)
  - Clock synchronization protocols
  - Mutual exclusion
  - Cache coherence protocols

Small Model Theorems

- Theorems state: if some property does **not** hold, then an instance of **some small bound size must be a counterexample**
- If **no instance up to the bound size violates the property**, then the property holds for instances of **any size**
- Introduced for safety properties of discrete systems
  - [Arons, Pnueli, Ruah, Xu, Zuck, CAV 2001], [Pnueli, Ruah, Zuck, TACAS 2001]
  - Extended for liveness properties
    - [Fang, Piterman, Pnueli, Zuck, 2004 TACAS and 2004 VMCAI]
    - [Fang, McMillan, Pnueli, Zuck, 2006 FORTE]
  - Completeness and relationship to Owicki-Gries proofs [Namjoshi 2007 VMCAI]
- Contributions of this paper
  - Extension to allow for safety verification with:
    - Continuous dynamics (hybrid systems): Rectangular differential inclusions: $\dot{x} \in [a, b]$
    - Distributed data structures (pointers between automata)
  - Implementation using Microsoft Z3 SMT-Solver

Grammar

- Grammar for LH-Assertions
  - $ITerm ::= \perp \mid 1 \mid N \mid i_j \mid p[i_j]$
  - $DTerm ::= L \mid q[ITerm]$
  - $RTerm ::= x \mid x[ITerm]$
  - $Atom ::= ITerm_1 = ITerm_2 \mid DTerm = L \mid$
    $a_1 RTerm_1 + a_2 RTerm_2 + a_3 < 0$
  - $Formula ::= Atom \mid \neg Formula \mid Formula_1 \wedge \mathrm{Formula}_2$
- Example
  - Initial states: $\forall i \in [N] \,.\, q[i] = Fly \wedge x[i] = 0.0 \wedge p[i] = \perp \wedge last = \perp$

Models

- Provide interpretation to elements in assertion
  - Example
    - Assertion: $\forall i, j \in [N].\, i \neq j \wedge p[j] = i \wedge x\big[p[j]\big] \geq L_S$
    - n-Model: $N = 2, \overline{p} = \langle 2,1 \rangle, \overline{x} = \langle 5.0,10.0 \rangle, L_S = 1.6$
  - Satisfiability: some model evaluates to true
  - Validity: all models satisfy assertion

Transitions and Trajectories

- Discrete actions
  - $T(N, \text{Final}, \text{Land}) \triangleq \exists i \in [N]\,.$

    $q_i = \text{Final} \wedge \mathrm{x_i} \geq L_B \wedge n_i = \bot \wedge$

    $q_i' = \text{Land} \wedge\ \mathrm{x}_i' = 0\ \wedge (\mathrm{l} = \mathrm{i} \Rightarrow g' = \bot) \wedge$

    $\forall j \in [N].\, j \neq i \wedge \left(n_\mathrm{j} = i \Rightarrow n_j' = \bot\right) \wedge$

    $q_j' = \mathrm{q_j} \wedge x_j' = x_j$
- Trajectories
  - $Trajs(N) \triangleq$

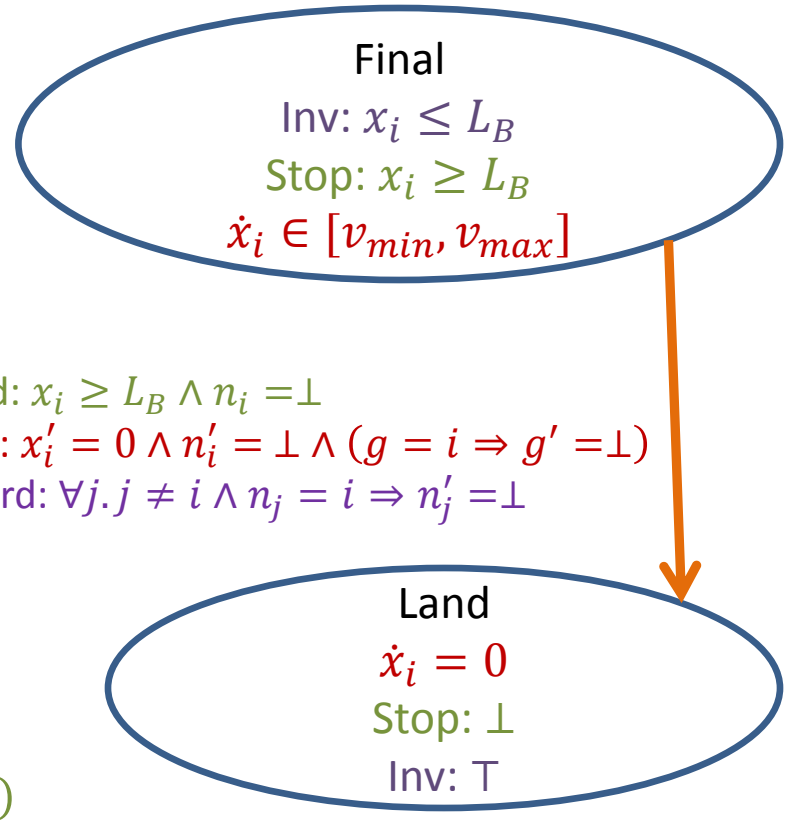
    $\exists t_1 \in \mathbb{R}_{\geq 0}, \forall i \in [N], \forall \mathrm{t}_2 \in \mathbb{R}_{\geq 0}\,.\, t_2 \leq t_1 \wedge$

    $q_i = \text{Final} \Rightarrow$

    $(\mathrm{x_i} + \mathrm{v_{min}}\ \mathrm{t}_2 \leq L_B \wedge x_i + v_{\mathrm{max}}\ \mathrm{t}_2 \leq L_B)\ \wedge$

    $(\mathrm{x_i} + \mathrm{v_{min}}\ \mathrm{t}_2 \geq L_B \vee x_i + v_{max} t_2 \geq L_B \Rightarrow t_2 = t_1)$

    $\wedge\, x_i' \in x_i + [v_{min} t_1, v_{max} t_1]$



Small Model Theorem

Consider an LH-assertion ψ of the form
$\forall t_1 \in \mathbb{R}\, \forall i_1, \ldots, \mathrm{i}_k \in [N] \exists t_2 \in \mathbb{R}\, \exists j_1, \ldots, j_m \in [N].\, \varphi$,
where φ is a quantifier-free formula involving the quantified variables and automaton variables

ψ is valid iff for all $n \leq N_0 = (e+1)(k+2)$, ψ is satisfied by all n-models

e: number of index array variables (pointers) in φ

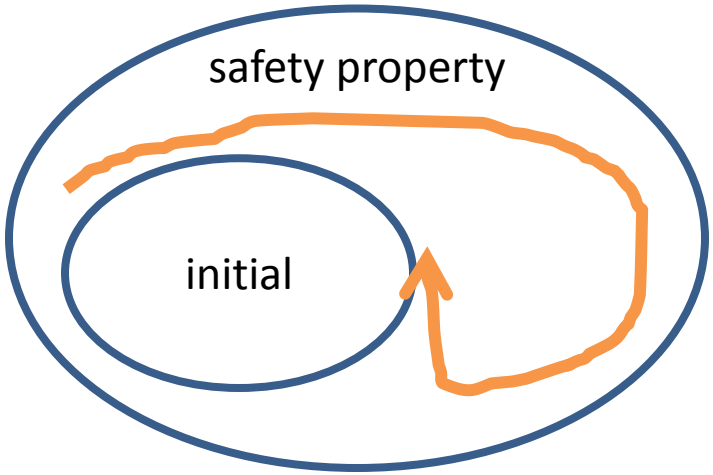
k: largest subscript of universally quantified variables in ψ

Proof Sketch

- Assume all n-models for $n \leq (e+1)(k+2)$ satisfy ψ, and we show ψ is valid
  - Suppose not, so ψ is not valid
    - Then, there exists a model of size $n > (e+1)(k+2)$ satisfying $\neg\psi = \exists t_1, i_1, \ldots, i_k \forall t_2, j_1, \ldots, j_m : \neg\varphi$
    - For any n-model of size $n > (e+1)(k+2)$, we will construct (n-1)-model contradicting the assumption
      - n-model assigns values to all variables
      - Universally quantified not important: any assignment satisfies $\neg\psi$
      - Set of **distinct** values assigned to index variables: $\leq k$
      - Distinct index array term values: $\leq k+2$
        - » At most e index arrays
        - » **Pigeonhole**: implies there is an unassigned value, u, in {1,…,n}
      - Remove u and reassign any variables that were u
        - » Index variables
          - < u: assign same value
          - > u: decrement by 1
        - » Reassign model values for array variables

Proving Inductive Invariants

- Using the small model theorem
  - If there is a counterexample, we will find it when attempting to verify $\mathcal{A}_1 \parallel \mathcal{A}_2 \parallel \cdots \parallel \mathcal{A}_N \vDash P(\mathrm{N})$ for some $1 \leq N \leq (e+1)(k+2)$
- Conditions for checking inductive invariance are LH-assertions
  - Initiation
    - $\forall i \in [N].\, Init_i \Rightarrow \forall i \in [N].\, P(i)$
  - Transition consecution
  - Trajectory consecution



Passel: Tool Implementation

- Passel: “a large group of people or things of indeterminate number”
- Microsoft Z3 SMT-solver backend
- Variables modeled using uninterpreted functions
  - Map from set of process ids to the variable type
- Transitions modeled using quantified formulas
  - Using model-based quantifier instantiation (MBQI) and quantifier elimination procedures
- List of properties given with the protocol
  - Check if a property is an inductive invariant
    - Checks each discrete transition and the trajectory relation
    - If so, assert it as a quantified axiom, continue

Passel Results: SATS

Property	Time (s)	Quantifier Instantiations	N_0	e	k
$\forall i \in [N]. q_i = \boldsymbol{F} \Rightarrow l \neq i$	0.47	166	9	2	1
$\forall i, j \in [N]. n_j = i \Rightarrow q_i \neq \boldsymbol{F}$	0.591	373	12	2	2
$\forall i, j \in [N]. q_i = \boldsymbol{H} \wedge n_j = i \Rightarrow q_j = \boldsymbol{H}$	0.586	703	12	2	2
$\forall i, j \in [N]. q_i = \boldsymbol{B} \wedge q_j = \boldsymbol{B} \wedge n_j = i \Rightarrow x_i \geq L_S + c(L_B - x_j)$	0.757	8298	12	2	2
$\forall i, j \in . i \neq j \wedge q_i = \boldsymbol{B} \wedge q_j = \boldsymbol{B} \wedge n_j = i \Rightarrow x_i - x_j \geq L_S$	0.467	3032	12	2	2

Passel Results: Fischer's

Property	Time (s)	Quantifier Instantiations	N_0	e	k
$\forall i, j \in [N]. x_i = x_j$	0.498	305	8	1	2
$\forall i \in [N]. q_i = \boldsymbol{set} \Rightarrow last_i \leq x_i + A$	0.330	204	6	1	1
$\forall i \in [N]. q_i = \boldsymbol{set} \Rightarrow x_i \leq last_i$	0.376	544	6	1	1
$\forall i, j \in [N]. \left(q_i = \boldsymbol{check} \wedge g = i \wedge q_j = \boldsymbol{set}\right) \Rightarrow first_i > last_j$	0.396	618	8	1	2
$\forall i, j \in [N]. q_i = \boldsymbol{crit} \Rightarrow \left(g = i \wedge q_j \neq \boldsymbol{set}\right)$	0.435	1306	8	1	2
$\forall i, j \in [N]. (i \neq j) \Rightarrow (q_i \neq crit \vee q_j \neq crit)$	0.414	1036	8	1	2

Conclusion

- Extended small model results for networks of rectangular hybrid automata
- Passel: tool for automatically checking inductive invariants using Microsoft Z3

- Future work
  - Invariant synthesis
  - Additional case studies

Thanks and Questions

- Thank you for your attention

- Acknowledgement
  - This work was supported by the National Science Foundation under CAREER Grant 1054247

Extra Slides

Inductive Invariance and Semantics

transition consecution: for each $(\mathbf{v}, \mathbf{v}') \in D_N$, $\mathbf{v} \models \psi \Rightarrow \mathbf{v}' \models \psi$
trajectory consecution: for each $(\mathbf{v}, \mathbf{v}') \in T_N$, $\mathbf{v} \models \psi \Rightarrow \mathbf{v}' \models \psi$

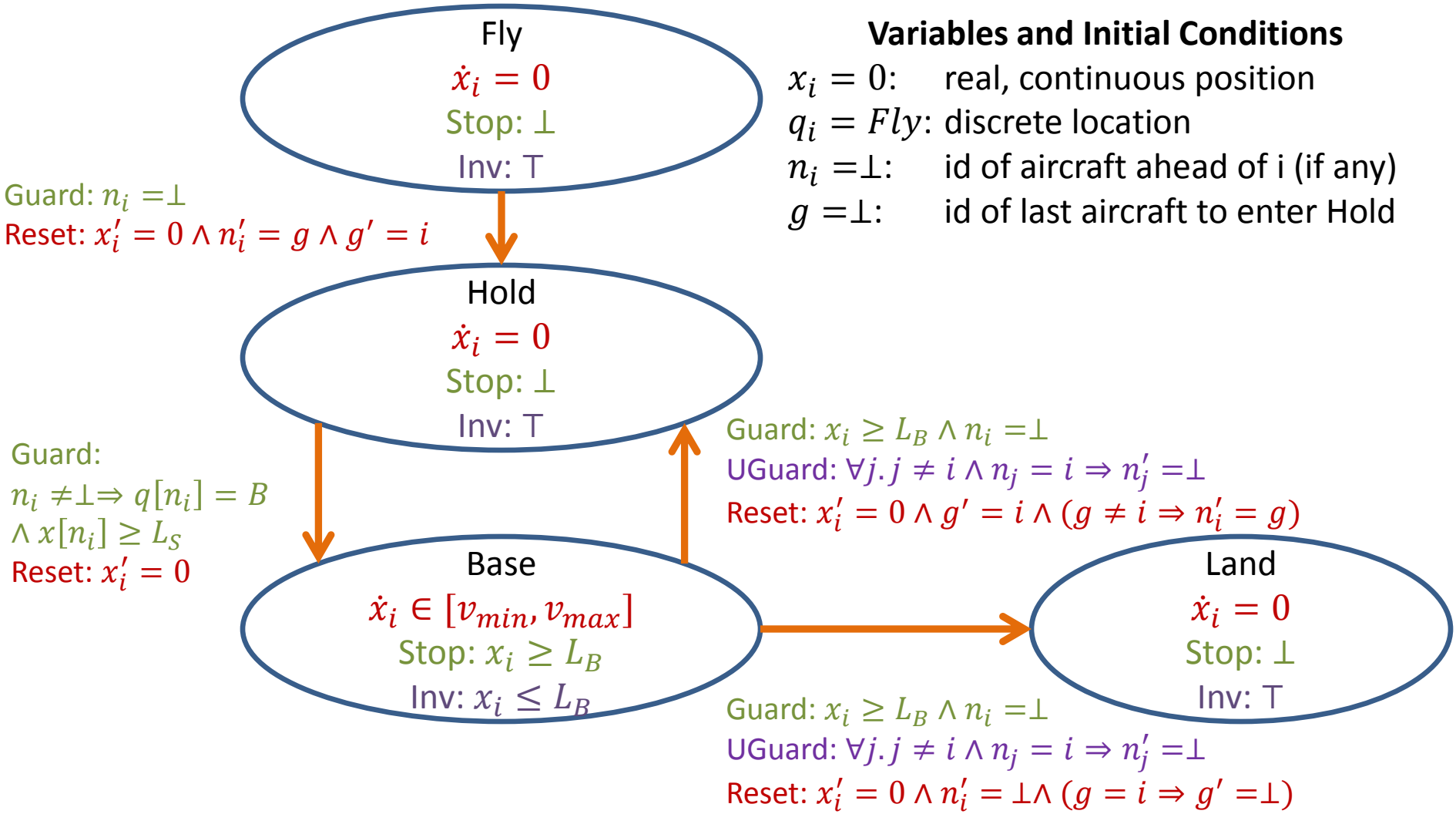
$$\exists t_1 \in \mathbb{R}_{\geq 0} : \ \forall i \in [N] : \exists \mathsf{m} \in \mathsf{Mode}_i : \forall t_2 \in \mathbb{R}_{\geq 0} : t_2 \leq t_1 \wedge$$
$$(\mathbf{flow}(\mathsf{m}, \mathbf{v}[i], t_2) \models \mathbf{inv}(\mathsf{m}, i) \wedge \mathbf{flow}(\mathsf{m}, \mathbf{v}[i], t_2) \models \mathbf{stop}(\mathsf{m}, i) \Rightarrow t_2 = t_1)$$
$$\wedge\ \mathbf{v}'[i] \in \mathbf{flow}(\mathsf{m}, \mathbf{v}[i], t_1).$$

$$\exists\ i \in [N] : \ \exists\ \mathsf{a} \in \mathsf{Act}_i : \mathbf{v} \ \models \mathbf{pre}(\mathsf{a}, i) \ \wedge \ (\mathbf{v}, \mathbf{v}') \models \mathbf{eff}(\mathsf{a}, i) \ \wedge$$
$$\forall\ j \in [N] : \ j \neq i \wedge j \notin M \implies \mathbf{v}'[j] = \mathbf{v}[j]$$

Parameterized Verification Approaches

- "Semi-formal": Fix N, verify for small number using traditional model checker (Uppaal, HyTech, PHAVer, etc.)
- Formal
  - Finite-state automata: Undecidable in general [Apt and Kozen, 1986]
  - Timed automata
    - Decidable with a single real-valued clock, finite number of integer clocks [Abdulla et al. 2001-04]
    - Undecidable with two or more real-valued clocks, urgency, universal guards [Abdulla et al. 2001-04]
  - Counter abstraction [Delzanno 2000], Environment abstraction [Clarke, Talupur, and Veith 2006], Network invariants [Wolper, Lovinfosse, 1990], Small model theorems [Pnueli et al. 2001]
  - Theorem proving
    - SATS: Discrete abstractions [Munoz, Dowek, and Carreno 2004], Hybrid versions [Munoz and Dowek 2005-06], [Umeno and Lynch 2007]
    - Adaptive cruise control [Loos, Platzer, et al. 2011]
    - Fischer's mutual exclusion [Dutertre and Sorea 2004]
  - MCMT: tool for backward reachability [Ghilardi et al. IJCAR 2008], Timed automata [Carioni et al. 2010]

Simplified SATS Hybrid Automaton



Variables and Initial Conditions

$x_i = 0$: real, continuous position

$q_i = Fly$: discrete location

$n_i = \perp$: id of aircraft ahead of i (if any)

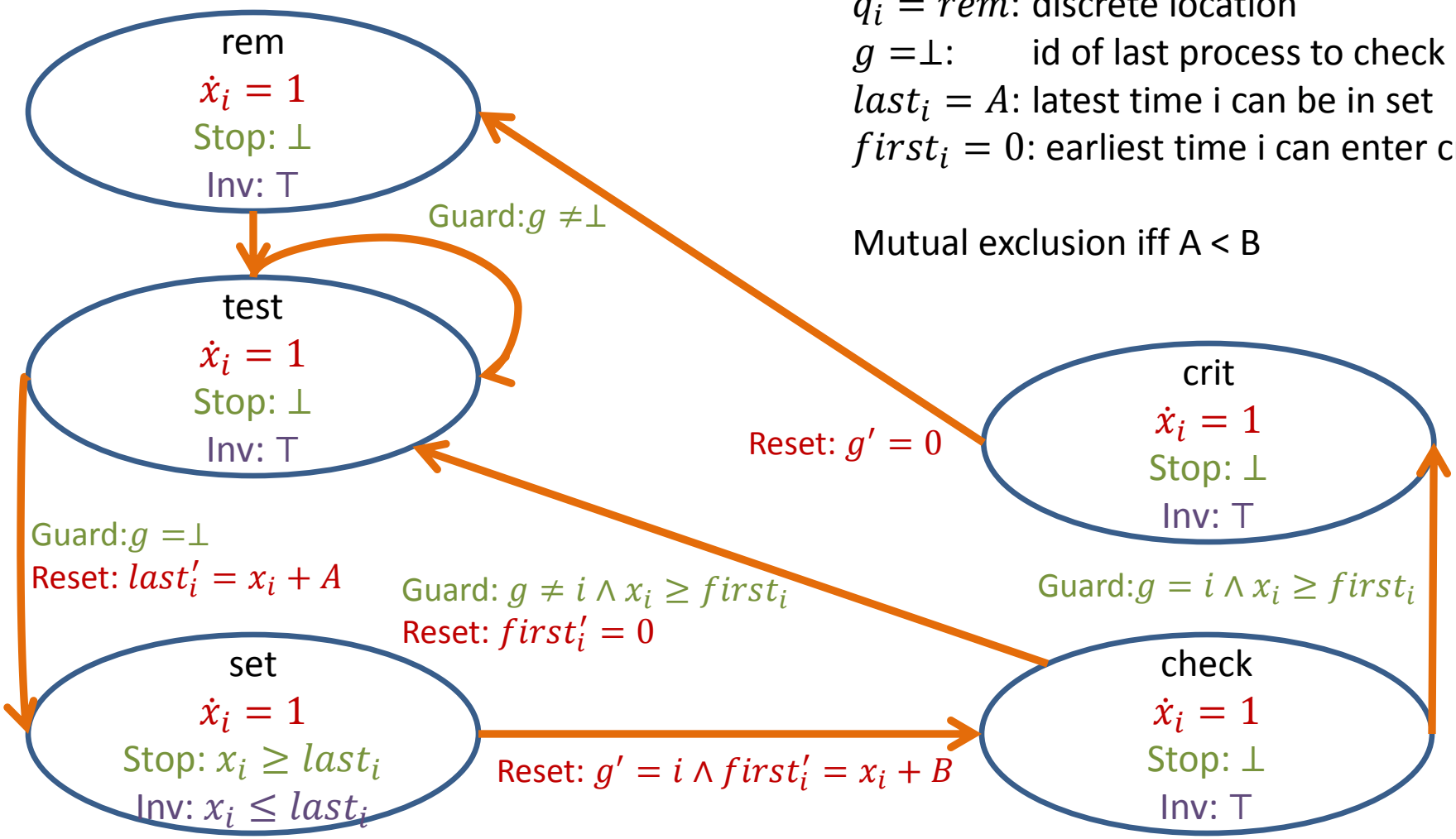
$g = \perp$: id of last aircraft to enter Hold

Fischer's Mutual Exclusion Automaton

Variables and Initial Conditions

$x_i = 0$: real, continuous clock
$q_i = rem$: discrete location
$g = \perp$: id of last process to check
$last_i = A$: latest time i can be in set
$first_i = 0$: earliest time i can enter crit

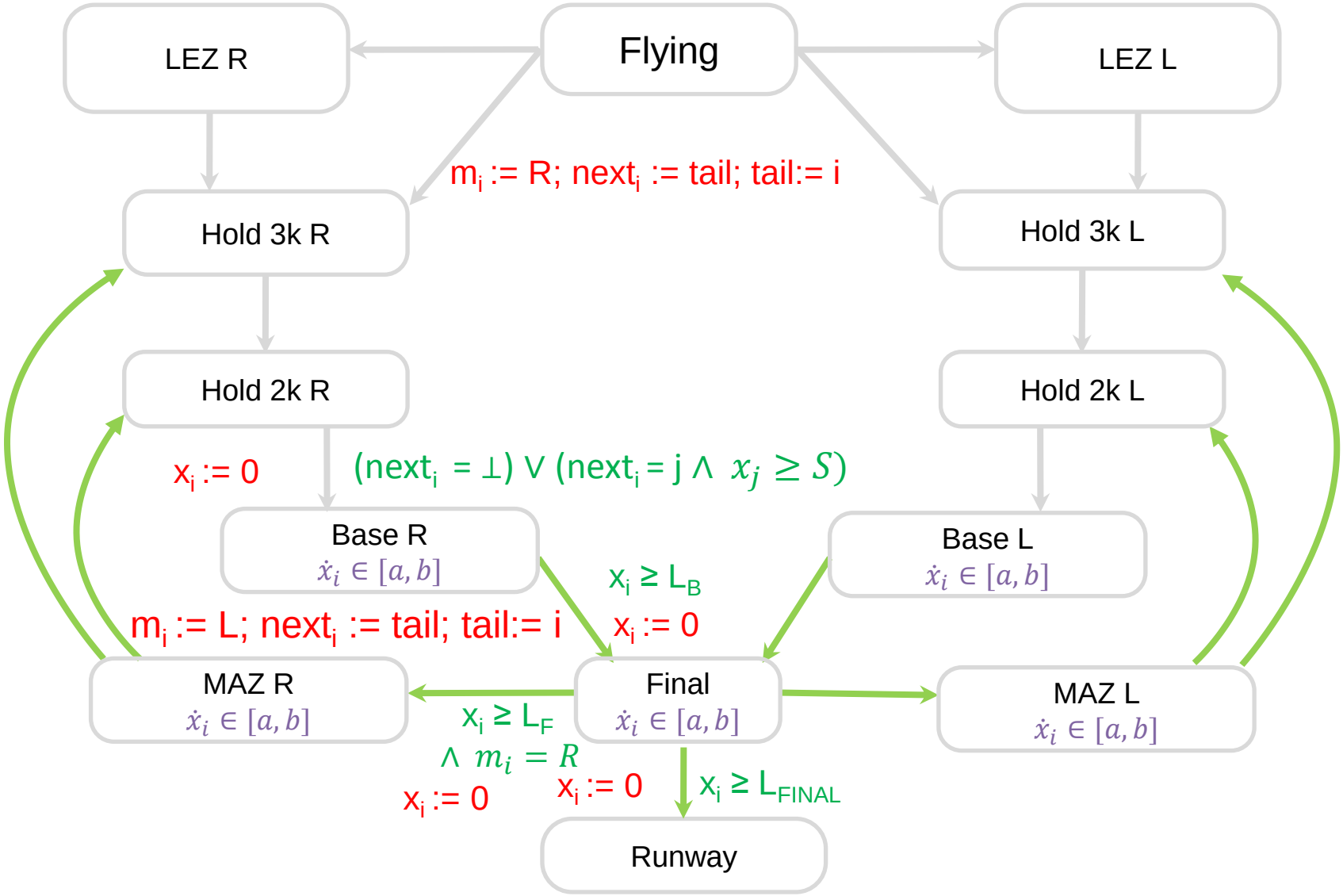
Mutual exclusion iff A < B



Future Work: Invariant Synthesis

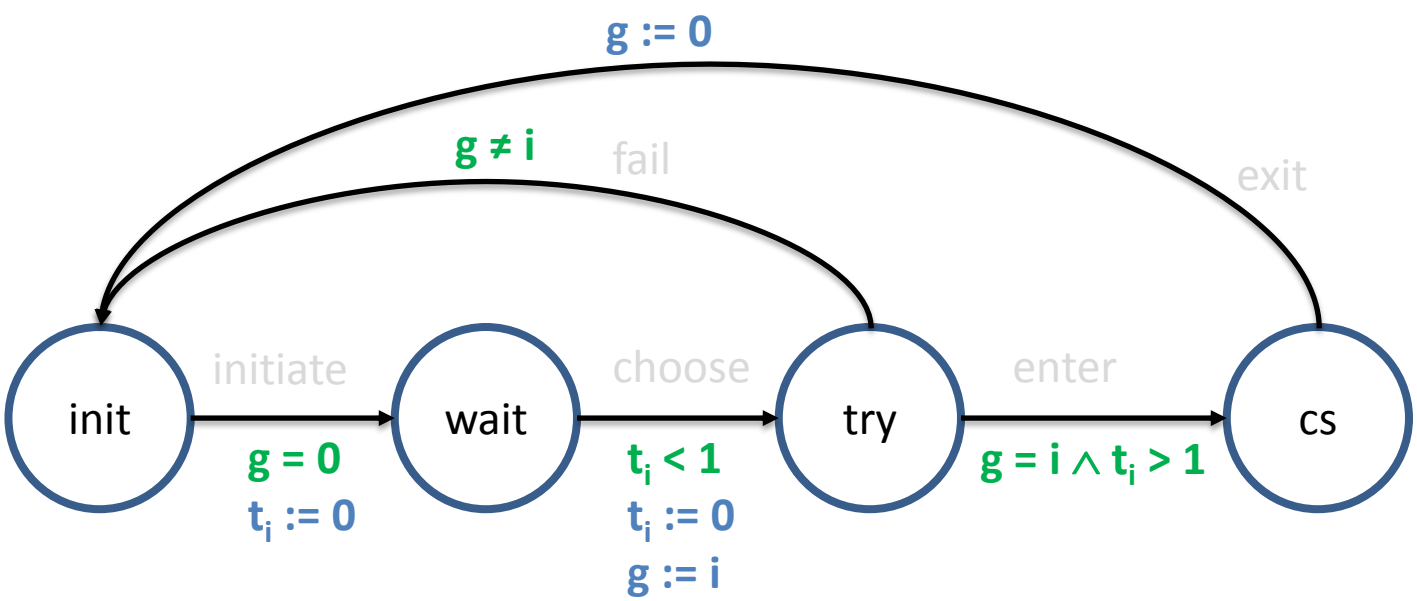
- Invisible invariant method
  - Compute reach set for a small instantiation
  - Reach sets will be symmetric (every process is identical)
  - Generalize reach set with replacement (q[1/i])
  - Complete for split invariants, if reach set can be computed
    - Split invariants: Conjunction of local variables
    - Abstractions for reach set approximation?

Example: Full SATS Model



Example: Fischer's Protocol

- Timed mutual exclusion protocol
  - 4 states: initial, waiting, trying, critical section
  - 1 real-valued clock per process
  - 1 globally shared (atomic) variable, g, ranging over process ids
  - Process ids: {1, …, n}
  - Safety property: at most one process is in critical section



Small Model Theorem

- For an LH-assertion ψ of the form $\forall t_1 \in \mathbb{R}\, \forall i_1, \ldots, \mathrm{i}_k \in [N] \exists t_2 \in \mathbb{R}\, \exists j_1, \ldots, j_m \in [N].\, \varphi$, where φ is a quantifier-free formula involving the quantified variables and agent variables. Then, ψ is valid iff for all $n \leq N_0 = (e+1)(k+2)$, ψ is satisfied by all n-models.
- Proof (intuition):
  - Assume all n-models for $n \leq (e+1)(k+2)$ satisfy ψ, and show ψ is valid
    - Suppose not, then there is a model of size $n > (e+1)(k+2)$ satisfying $\neg\psi = \exists t_1, i_1, \ldots, i_k \forall t_2, j_1, \ldots, j_m. \neg\, \varphi$
    - For any model of size $n > (e+1)(k+2)$, there is a contradicting model of size $n-1$
      - Show there is some unassigned index array term, and remap array indices
- Theorem and proof are minor extensions
  - Pnueli et al's invisible invariant papers, early 2000s

Results

Example	Property	Correct	Time (s)	QI	Buggy	Time (s)	QI
SATS	A	✓	0.47	166	✓	24.429	63
	B	✓	0.591	373	✓	0.595	197
	C	✓	0.586	703	✗	1.041	113485
	D	✓	0.757	8298	✗	1.256	1659
Timed SATS	A	✓	0.349	66	✓	0.34	61
	B	✓	0.304	673	✓	0.317	460
	C	✓	0.244	373	✗	1.763	140512
	E	✓	0.467	3032	✗	2.204	26958
Fischer	a	✓	0.498	305	✓	0.491	305
	b	✓	0.33	204	✓	0.325	204
	c	✓	0.376	544	✓	0.33	544
	d	✓	0.396	618	✗	0.533	2548
	e	✓	0.435	1306	✗	0.532	1202
	f	✓	0.414	1036	✗	0.437	3162

(A) $\forall i \in [N] : l[i] = F \Rightarrow last \neq i$,
(B) $\forall i, j \in [N] : next[j] = i \Rightarrow l[i] \neq F$,
(C) $\forall i, j \in [N] : l[i] = H \wedge next[j] = i \Rightarrow l[j] = H$,

(D) $\forall i, j \in [N] : l[i] = B \wedge l[j] = B \wedge next[j] = i \Rightarrow x[i] \geq L_S + (v_{max} - v_{min})\frac{L_B - x[j]}{v_{min}}$, and
(E) $\forall i, j \in [N] : i \neq j \wedge l[i] = B \wedge l[j] = B \wedge next[j] = i \Rightarrow x[i] - x[j] \geq L_S$.

(a) $\forall i, j \in [N] : x[i] = x[j]$,
(b) $\forall i \in [N] : q[i] = set \Rightarrow last[i] \leq x[i] + A$,
(c) $\forall i \in [N] : q[i] = set \Rightarrow x[i] \leq last[i]$,
(d) $\forall i, j \in [N] : (q[i] = check \wedge g = i \wedge q[j] = set) \Rightarrow first[i] > last[j]$,
(e) $\forall i, j \in [N] : q[i] = crit \Rightarrow (g = i \wedge q[j] = set)$, and
(f) $\forall i, j \in [N] : (i = j) \Rightarrow (q[i] = crit \vee q[j] = crit)$,

Passel Results: Fischer's

Property	Time (s)	Quantifier Instantiations	N_0	e	k
$\forall i, j \in [N]. x_i = x_j$	0.498	305	8	1	2
$\forall i \in [N]. q_i = \boldsymbol{set} \Rightarrow last_i \leq x_i + A$	0.330	204	6	1	1
$\forall i \in [N]. q_i = \boldsymbol{set} \Rightarrow x_i \leq last_i$	0.376	544	6	1	1
$\forall i, j \in [N]. \left(q_i = \boldsymbol{check} \wedge g = i \wedge q_j = \boldsymbol{set}\right) \Rightarrow first_i > last_j$	0.396	618	8	1	2
$\forall i, j \in [N]. q_i = \boldsymbol{crit} \Rightarrow \left(g = i \wedge q_j \neq \boldsymbol{set}\right)$	0.435	1306	8	1	2
$\forall i, j \in [N]. (i \neq j) \Rightarrow (q_i \neq crit \vee q_j \neq crit)$	0.414	1036	8	1	2