Discrete Controller Synthesis for Infinite State Systems with ReaX

Nicolas BERTHIER Hervé MARCHAND

Inria Rennes — Bretagne Atlantique

WODES' 14

May 14, 2014

Outline

- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
- ReaX: Technical Choices, Implementation & Evaluations
- Conclusions

Outline

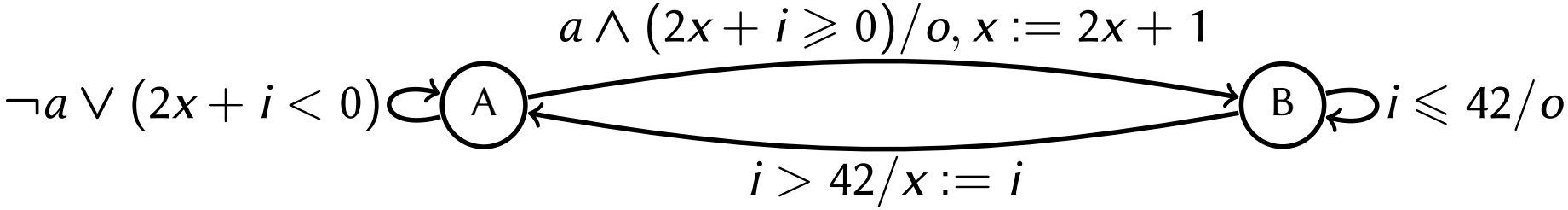
- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
- ReaX: Technical Choices, Implementation & Evaluations
- Conclusions

ASTS Model for Infinite State Systems

Definition (Arithmetic Symbolic Transition System — ASTS)

$S = \langle X, I, T, A, \Theta_0 \rangle$ where:

- $X = \langle x_1, \ldots, x_n \rangle$; $\mathcal{D}_X = \mathbb{B}^r \times \mathbb{Z}^s \times \mathbb{R}^t$ ← Vector of State Variables
- $I = \langle i_1, \ldots, i_m \rangle$; $\mathcal{D}_I = \mathbb{B}^u \times \mathbb{Z}^v \times \mathbb{R}^w$ ← Vector of Input Variables
- $T = \langle t_1, \ldots, t_n \rangle$; t_i: Expression on $X \cup I$ ← Transition Function
- A: Predicate on $X \cup I$ ← Assertion
- Θ_0: Predicate on X ← Initial State(s)



- $X = \langle \xi, x, o \rangle$, $I = \langle a, i \rangle$ $\mathcal{D}_X = \{\mathrm{A}, \mathrm{B}\} \times \mathbb{Z} \times \mathbb{B}$, $\mathcal{D}_I = \mathbb{B} \times \mathbb{Z}$
- $A(\langle \xi, x, o, a, i \rangle) = (\xi = \mathrm{B} \wedge 3x + 2i \leqslant 41 \wedge a)$
- $\Theta_0(\langle \xi, x, o \rangle) = (\xi = \mathrm{A} \wedge x = 0 \wedge \neg o)$

Outline

- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
- ReaX: Technical Choices, Implementation & Evaluations
- Conclusions

Safety Control Problem

- Initiated by Ramadge and Wonham 1989[1]

Definition (Invariant for an ASTS)

Given an ASTS $S = \langle X, I, T, A, \Theta_0 \rangle$, a Predicate Φ over X is an *Invariant* of S (Noted $S \models \Phi$) iff All Reachable States of S Satisfy Φ *i.e.*, $\forall p \in \mathbb{N}$

$\forall x_0 \in \mathcal{D}_X$ ← Initial State

$\forall (\iota_0, \ldots, \iota_p) \in \mathcal{D}_I^p$ ← Sequence of p Vectors of Inputs

$$\Theta_0(x_0) \quad \wedge \quad \forall i \in [0, p], A(T(\ldots T(x_0, \iota_0) \ldots, \iota_i)) \\ \Rightarrow \quad \forall i \in [0, p], \Phi(T(\ldots T(x_0, \iota_0) \ldots, \iota_i))$$

[1] Peter J. G. Ramadge and W. Murray Wonham. "The control of discrete event systems". In: *Proceedings of the IEEE* 77.1 (Jan. 1989), pp. 81–98.

Safety Control Problem

- Initiated by Ramadge and Wonham 1989

Definition (Invariant for an ASTS)

Given an ASTS $S = \langle X, I, T, A, \Theta_0 \rangle$, a Predicate Φ over X is an *Invariant* of S (Noted $S \models \Phi$) iff All Reachable States of S Satisfy Φ

Controller Synthesis Problem for Invariant Enforcement in ASTSs

Given and ASTS $S = \langle X, I_{uc} \uplus I_c, T, A, \Theta_0 \rangle$ where:

- I_{uc} ← *Non-controllable* Input Variables
- I_c ← *Controllable* Input Variables

and an Invariant Φ over X, ← Not Satisfied *a priori*
Compute a Predicate A_Φ such that:

$$S' = \langle X, I_{uc} \uplus I_c, T, A_\Phi, \Theta_0 \rangle \models \Phi \qquad \text{and} \qquad \forall v, A_\Phi(v) \Rightarrow A(v)$$

Outline

- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
  - Notations for Reasoning about State Spaces
  - Finite Case
  - Infinite Case (Contribution)
- ReaX: Technical Choices, Implementation & Evaluations
- Conclusions

Notations for Reasoning about State Spaces

Definition (Controllable Infinite Transition System of an ASTS)

One Associates to an ASTS $S = \langle X, I_{uc} \uplus I_c, T, A, \Theta_0 \rangle$ a *Controllable Infinite Transition System* $[S] = \langle \mathcal{X}, \mathcal{I}, \mathcal{T}_S, \mathcal{A}_S, \mathcal{X}_0 \rangle$ where:

- $\mathcal{X} = \mathcal{D}_X$ ← State Space
- $\mathcal{I} = \mathcal{U} \times \mathcal{C}$
  - $\mathcal{U} = \mathcal{D}_{I_{uc}}$ ← Non-controllable Input Space
  - $\mathcal{C} = \mathcal{D}_{I_c}$ ← Controllable Input Space
- $\mathcal{T}_S \subseteq \mathcal{X} \times \mathcal{I} \to \mathcal{X} \quad = \lambda(x, \iota).\,(t_i(x, \iota))_{i \in [1,n]}$ ← Transition Function
- $\mathcal{A}_S \subseteq \mathcal{X} \times \mathcal{I} \quad = \{(x, \iota) \mid A(x, \iota)\}$ ← Assertion on Environment
- $\mathcal{X}_0 \subseteq \mathcal{X} \quad = \{x \mid \Theta_0(x)\}$ ← Initial States

- $\mathcal{T}_S^{-1} \colon \wp(\mathcal{X}) \to \wp(\mathcal{X} \times \mathcal{U} \times \mathcal{C})$ ← Pre-image Function

Finite Case: Algorithmic Principle

Finite Case: State Variables on *Finite Domains* (*e.g.*, $\mathcal{X} = \mathbb{B}^n$)

- Proposed by Marchand et al. 2000[1]

$\rightsquigarrow$ Maximally Permissive Controller

[1] Hervé Marchand et al. "Synthesis of Discrete-Event Controllers based on the Signal Environment". In: *Discrete Event Dynamic System: Theory and Applications* 10.4 (Oct. 2000), pp. 325–346.

Finite Case: Algorithmic Principle

Finite Case: State Variables on *Finite Domains* (*e.g.*, $\mathcal{X} = \mathbb{B}^n$)

- Proposed by Marchand et al. 2000[1]

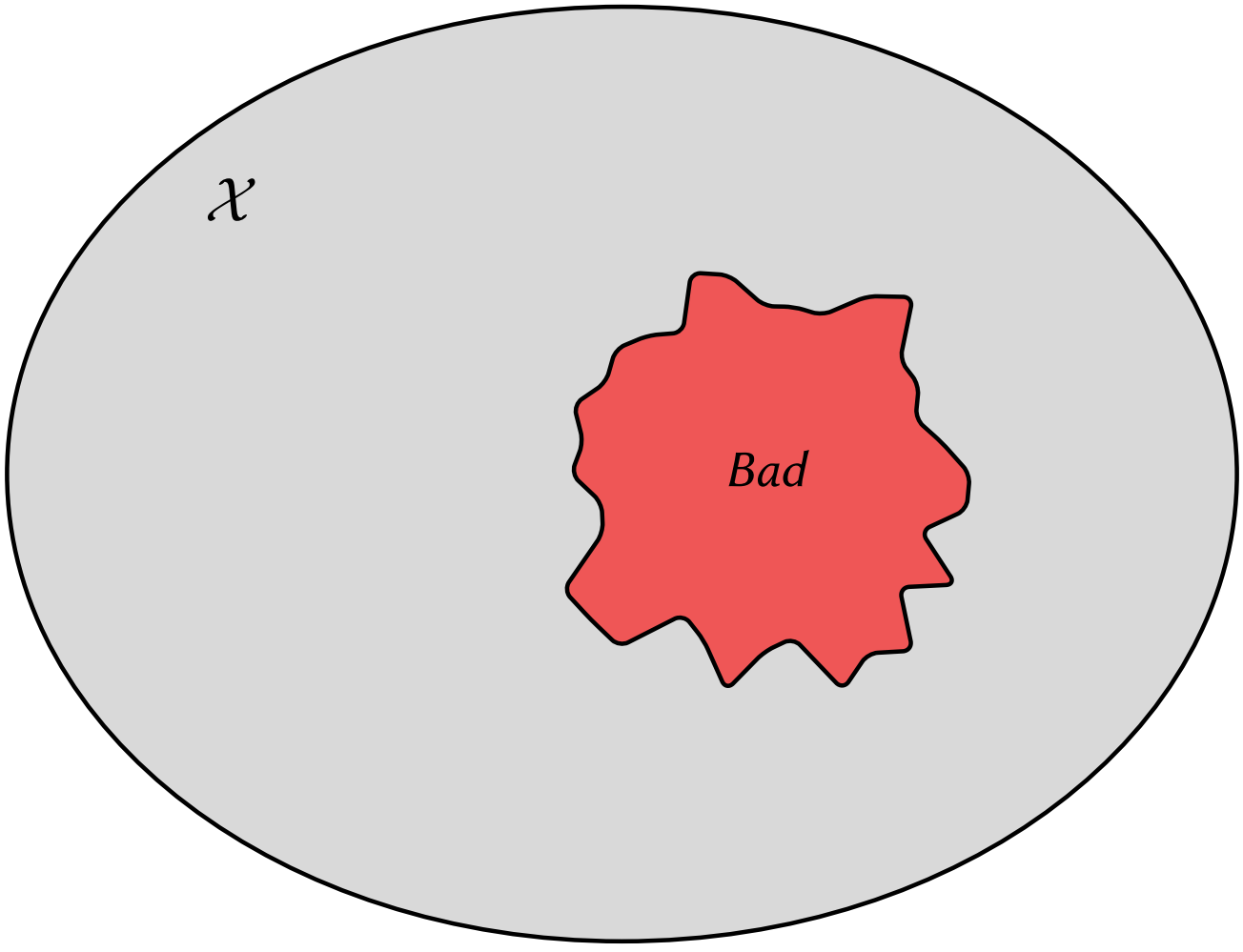
$\rightsquigarrow$ Maximally Permissive Controller

Informal Algorithm

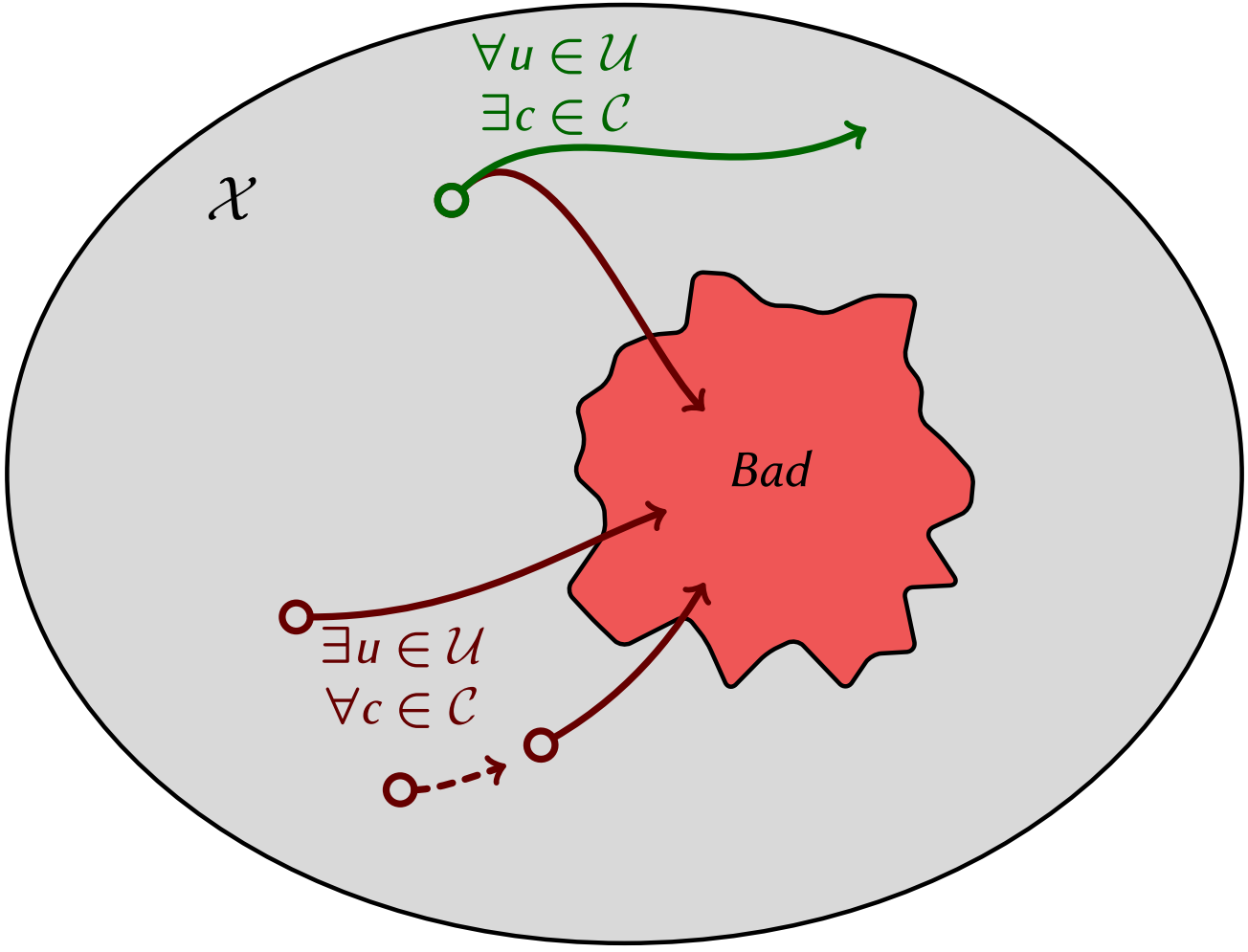
- Let $Bad = \{x \in \mathcal{X} \mid \neg\Phi(x)\}$ $\leftarrow$ States to Avoid
- I_{Bad} = States *Uncontrollably* Reaching *Bad* $\leftarrow$ Co-reachability
- Success iff $\mathcal{X}_0 \cap I_{Bad} = \varnothing$
- $\mathcal{A}_\Phi = \mathcal{T}_S^{-1}(I_{Bad}^c) \cap \mathcal{A}_S$ $\leftarrow$ Relating States with *Allowed Inputs*

[1] Hervé Marchand et al. "Synthesis of Discrete-Event Controllers based on the Signal Environment". In: *Discrete Event Dynamic System: Theory and Applications* 10.4 (Oct. 2000), pp. 325–346.

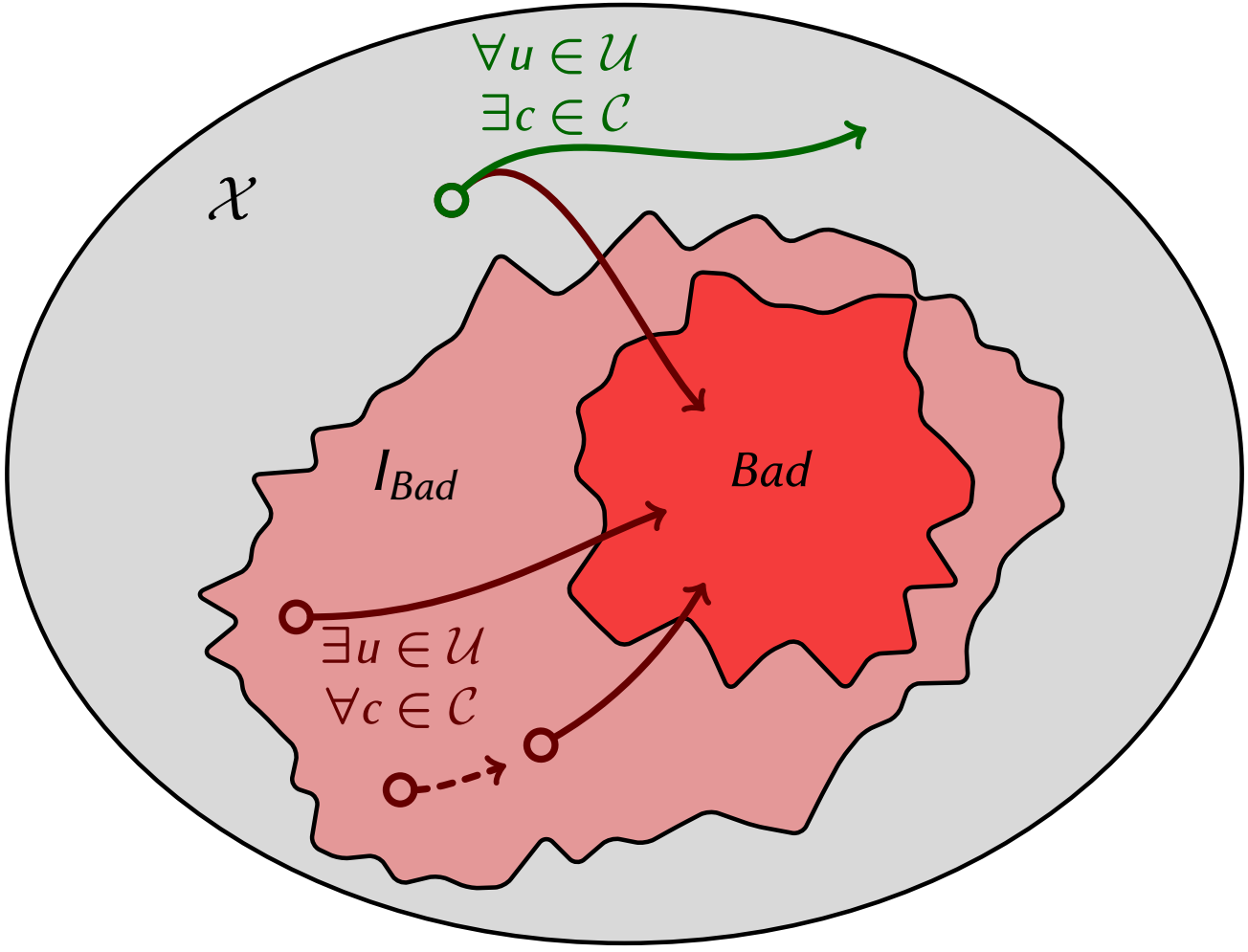
Finite Case: Computing I_{Bad}



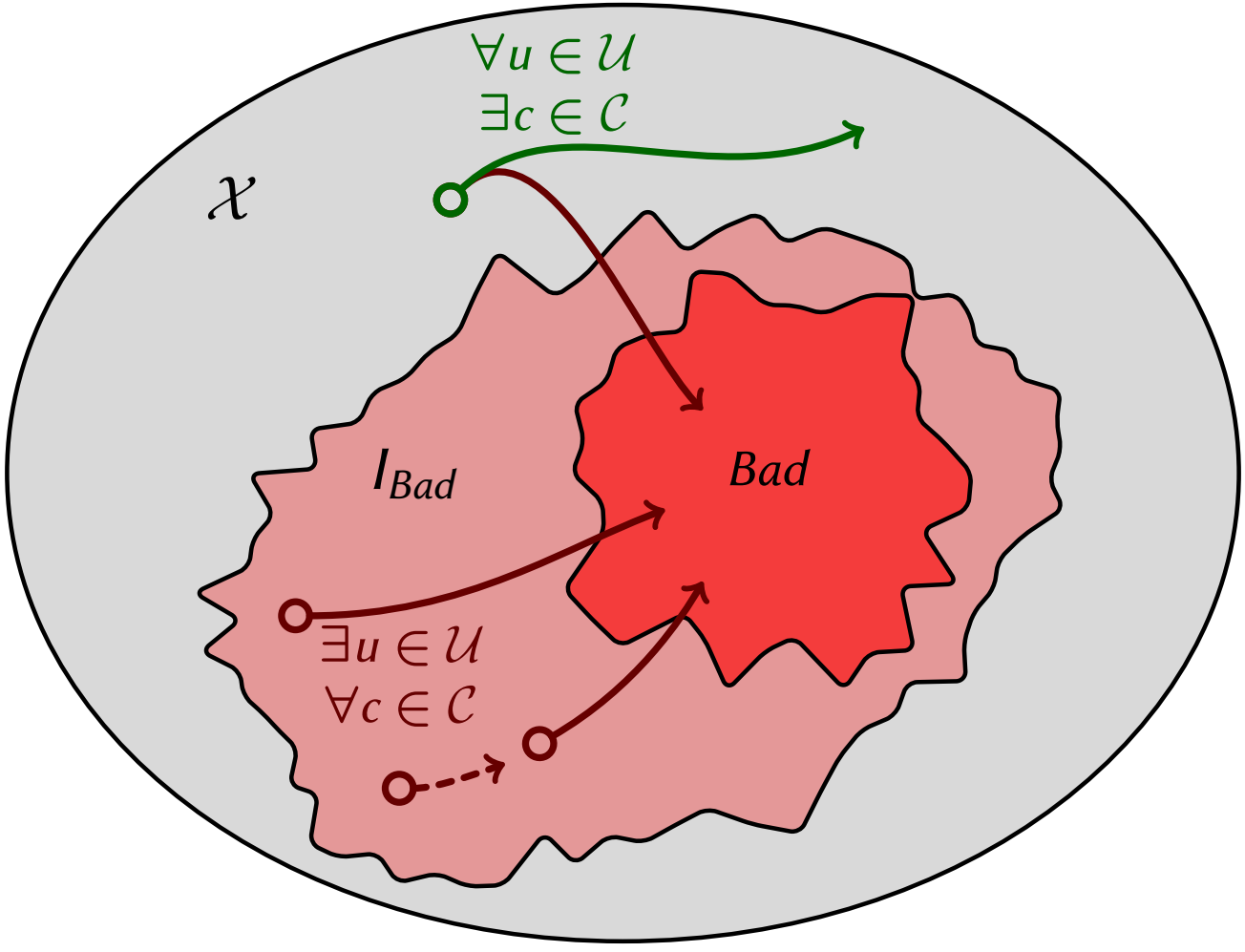
Finite Case: Computing I_{Bad}



Finite Case: Computing I_{Bad}



Finite Case: Computing I_{Bad}



$$I_{Bad} = \text{coreach}_u(Bad) \qquad \text{coreach}_u(B) \stackrel{\text{def}}{=} \text{lfp}(\lambda\beta.B \cup \text{pre}_u(\beta))$$

$$\text{pre}_u(B) \stackrel{\text{def}}{=} \left\{ x \in \mathcal{X} \,\middle|\, \exists u \in \mathcal{U}, \forall c \in \mathcal{C}, (x,u,c) \in \mathcal{T}_S^{-1}(B) \cap \mathcal{A}_S \right\}$$

Infinite Case: Algorithmic Principle (Contribution)

Infinite Case: Allowing *Numerical* Variables ($e.g., \mathcal{X} = \mathbb{B}^n \times \mathbb{Z}^m$)

- Undecidability Problem $\rightsquigarrow$ Over-approximating Solution
  - Using *Abstract Interpretation Techniques*
  - Computing $I'_{Bad}(\supseteq I_{Bad})$ $\not\rightsquigarrow$ Maximally Permissive Controller

Infinite Case: Algorithmic Principle (Contribution)

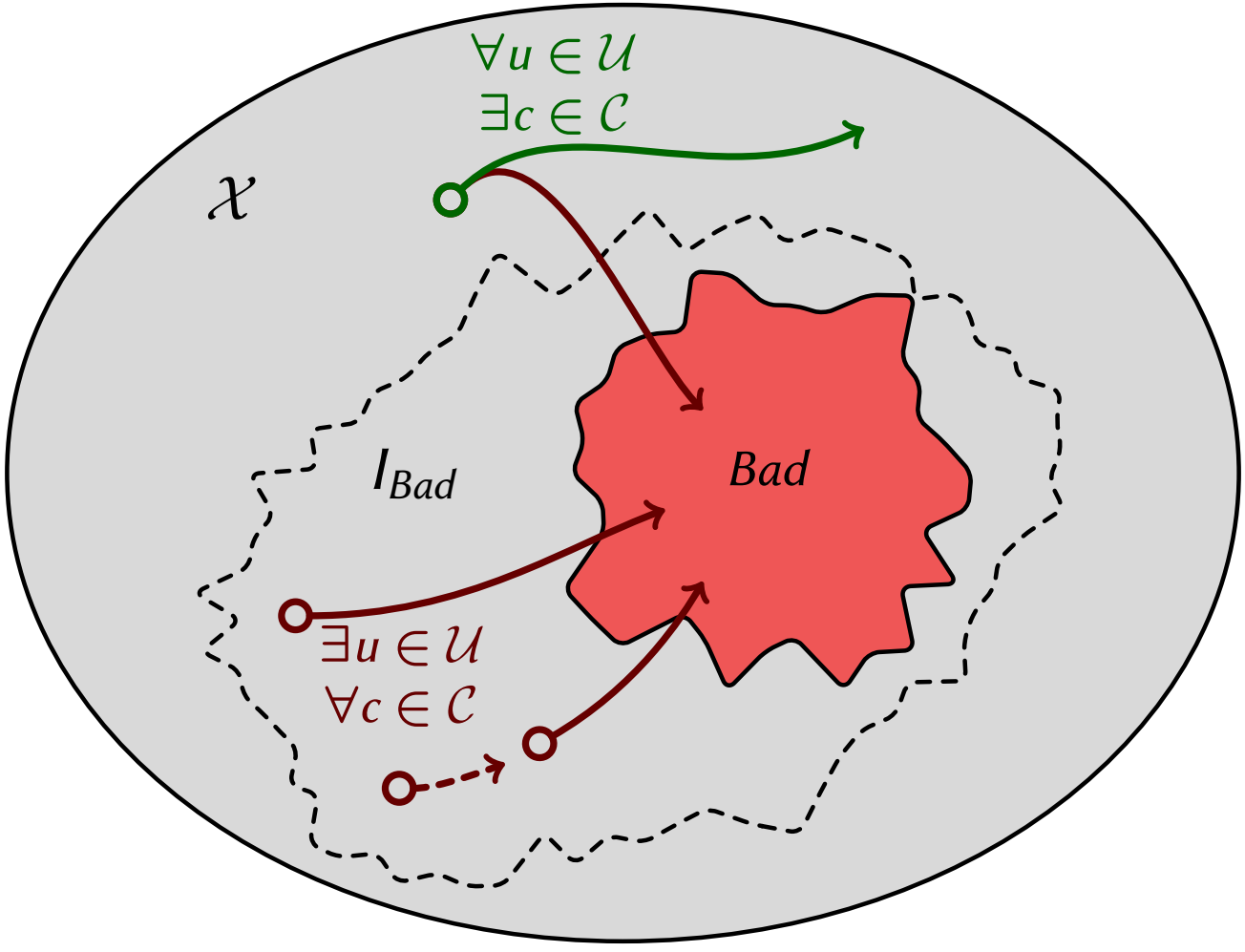
Infinite Case: Allowing *Numerical* Variables $(e.g., \mathcal{X} = \mathbb{B}^n \times \mathbb{Z}^m)$

- Undecidability Problem $\rightsquigarrow$ Over-approximating Solution
  - Using *Abstract Interpretation Techniques*
  - Computing $I'_{Bad}(\supseteq I_{Bad})$ $\not\rightsquigarrow$ Maximally Permissive Controller

Abstract Interpretation Requirements

- $\langle \Lambda, \sqsubseteq, \sqcup, \sqcap, \top, \bot \rangle$, α and γ such that: $\wp(\mathcal{X}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \Lambda$
  - $\wp(\mathcal{X})$ $\leftarrow$ Concrete Domain (Sets of States)
  - Λ $\leftarrow$ Abstract Domain (Finite Representation of Sets of States)
  - $\alpha : \wp(\mathcal{X}) \rightarrow \Lambda$ $\leftarrow$ Abstraction Function
  - $\gamma : \Lambda \rightarrow \wp(\mathcal{X})$ $\leftarrow$ Concretization Function
- $\mathcal{T}_S^{\sharp -1} : \Lambda \rightarrow \Lambda$ $\leftarrow$ Abstract Pre-image
- $\exists_Y^\sharp, \forall_Y^\sharp$ $\leftarrow$ Quantifier Elimination
- $\boldsymbol{\nabla} : \Lambda \times \Lambda \rightarrow \Lambda$ $\leftarrow$ Widening Operator, Forcing Convergence

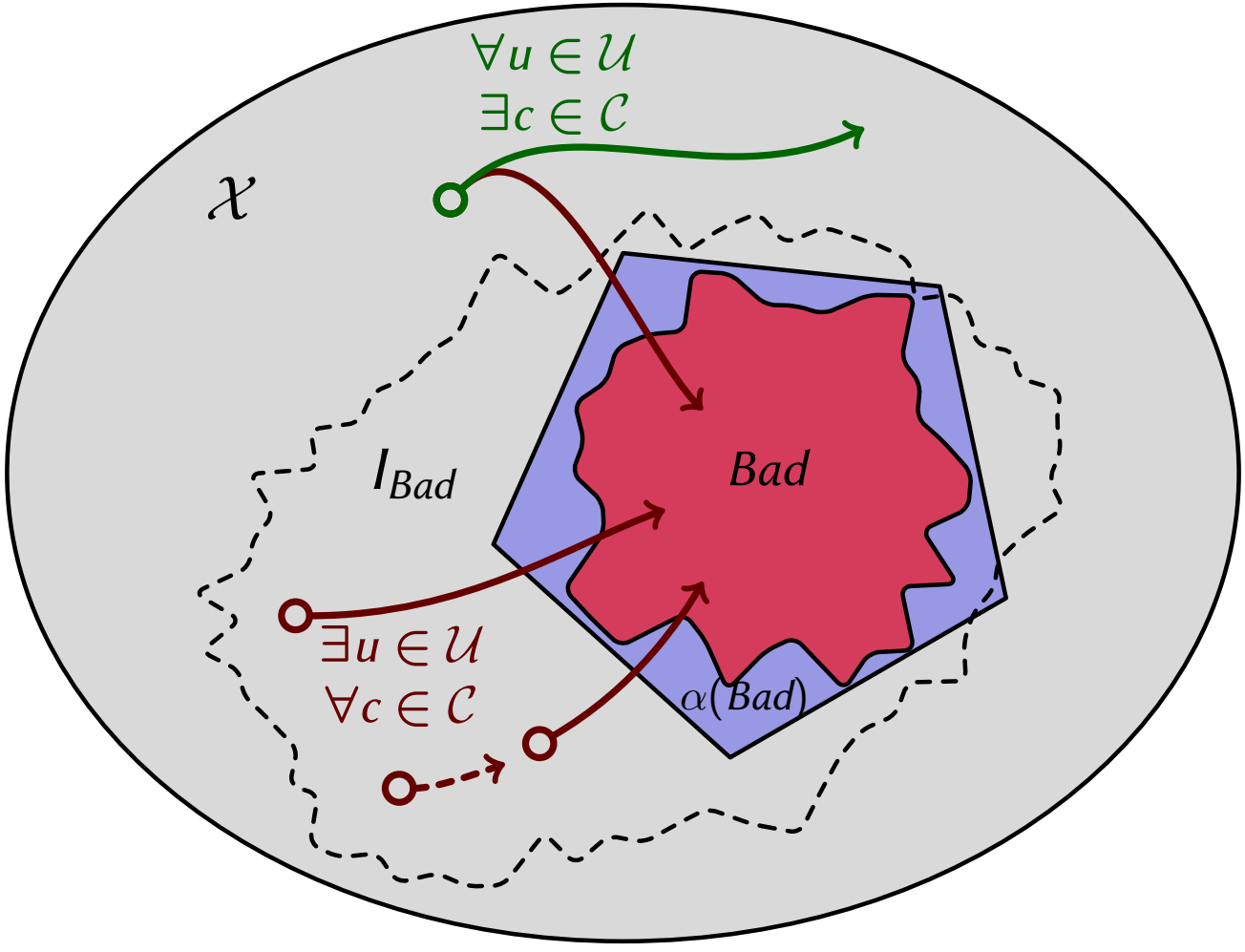
Infinite Case: Computing $I'_{Bad}(\supseteq I_{Bad})$



$$I_{Bad} = \text{coreach}_u(Bad) \qquad \text{coreach}_u(B) \stackrel{\text{def}}{=} \text{lfp}(\lambda\beta.B \cup \text{pre}_u(\beta))$$

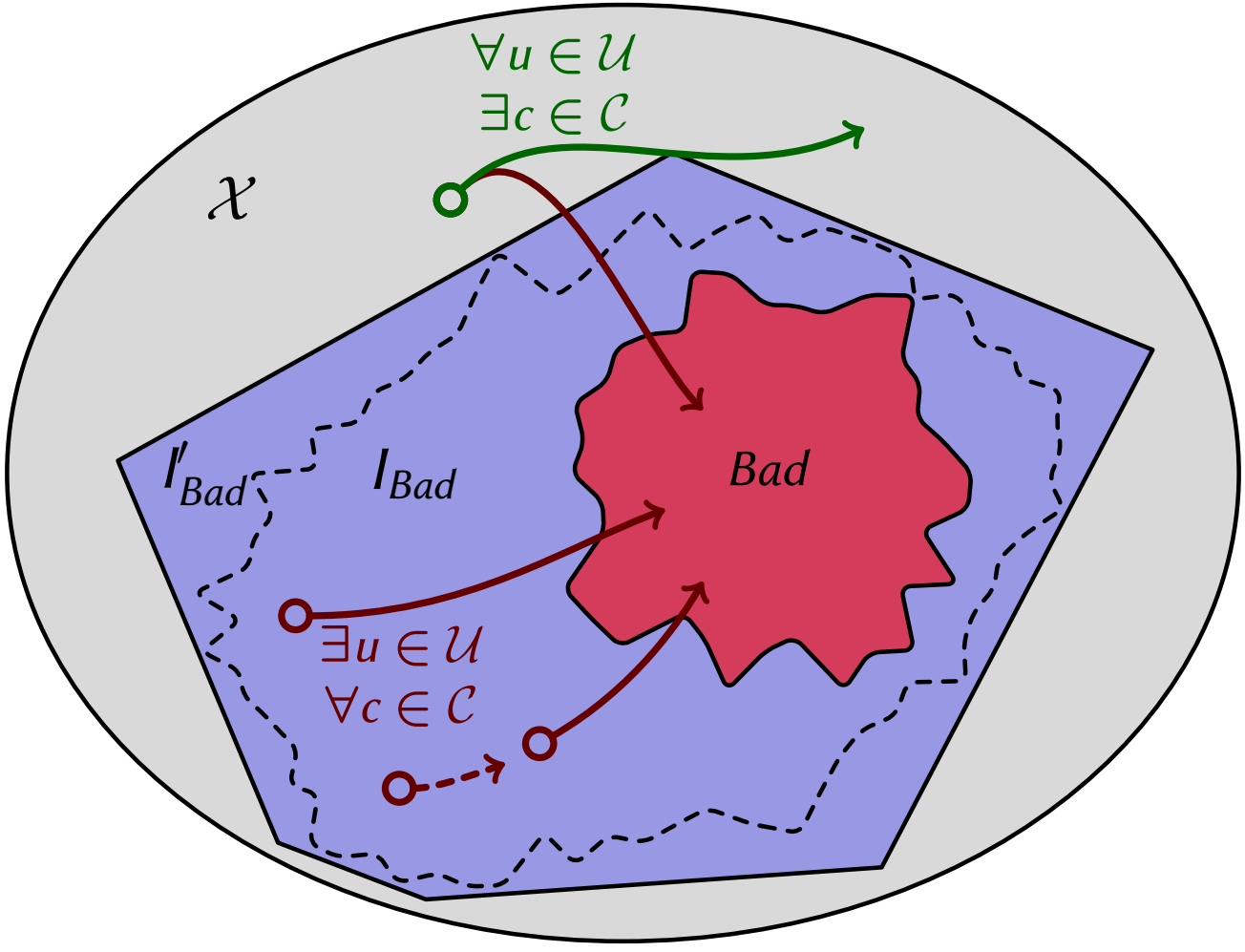
$$\text{pre}_u(B) \stackrel{\text{def}}{=} \left\{x \in \mathcal{X} \,\middle|\, \exists u \in \mathcal{U}, \forall c \in \mathcal{C}, (x,u,c) \in \mathcal{T}_S^{-1}(B) \cap \mathcal{A}_S\right\}$$

Infinite Case: Computing $I'_{Bad}(\supseteq I_{Bad})$



$$I'_{Bad} = \gamma \circ \text{coreach}^{\sharp}_{u} \circ \alpha(Bad)$$

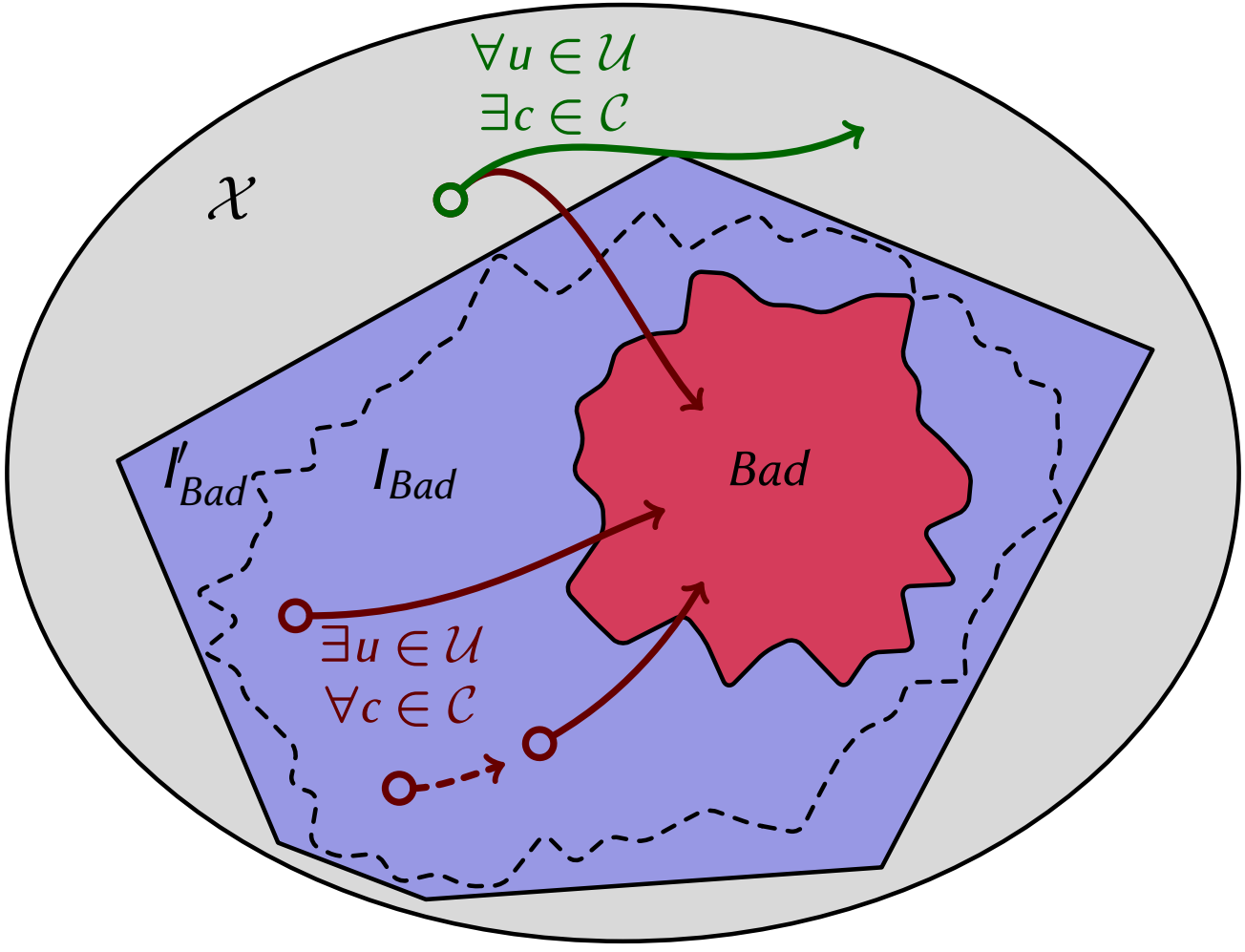
Infinite Case: Computing $I'_{Bad}(\supseteq I_{Bad})$



$$I'_{Bad} = \gamma \circ \text{coreach}^{\sharp}_{u} \circ \alpha(Bad) \qquad \text{coreach}^{\sharp}_{u}(B) \stackrel{\text{def}}{=} \text{lfp}(\lambda\beta.B \sqcup \text{pre}^{\sharp}_{u}(\beta))$$

$$\text{pre}^{\sharp}_{u}(B) \stackrel{\text{def}}{=} \exists^{\sharp}_{\mathcal{U}} \left(\forall^{\sharp}_{\mathcal{C}} \left(\mathcal{T}_{S}^{\sharp -1}(B) \sqcap \alpha(\mathcal{A}_S) \right) \right)$$

Infinite Case: Computing $I'_{Bad}(\supseteq I_{Bad})$



$$I'_{Bad} = \gamma \circ \text{coreach}_u^{\nabla} \circ \alpha(Bad) \qquad \text{coreach}_u^{\nabla}(B) \stackrel{\text{def}}{=} \text{lfp}(\lambda\beta.B \nabla \text{pre}_u^{\sharp}(\beta))$$

$$\text{pre}_u^{\sharp}(B) \stackrel{\text{def}}{=} \exists_{\mathcal{U}}^{\sharp}\left(\forall_{\mathcal{C}}^{\sharp}\left(\mathcal{T}_S^{\sharp -1}(B) \sqcap \alpha(\mathcal{A}_S)\right)\right)$$

Outline

- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
- ReaX: Technical Choices, Implementation & Evaluations
  - Over-approximating Logico-numerical State Spaces
  - Implementation Details
  - Evaluations of Over-approximating Synthesis
- Conclusions

Over-approximating Logico-numerical State Spaces

Over-approximating Numerical Spaces (*e.g.*, $\mathcal{X} = \mathbb{Z}^n \times \mathbb{R}^m$)

Numerical Abstract Domains

- Provide α, γ and $\mathcal{N}$ such that $\wp(\mathbb{Z}^n \times \mathbb{R}^m) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{N}$

Over-approximating Logico-numerical State Spaces

Over-approximating Numerical Spaces ($e.g., \mathcal{X} = \mathbb{Z}^n \times \mathbb{R}^m$)

Numerical Abstract Domains

- Provide α, γ and $\mathcal{N}$ such that $\wp(\mathbb{Z}^n \times \mathbb{R}^m) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{N}$

e.g.,

- *Intervals*: Conjunction of $n + m$ Bound Constraints of the form:
$$(a_i \leqslant v_i \leqslant b_i)$$
- *Convex Polyhedra*: Conjunction of k Linear Constraints of the form:
$$\left(\sum_{i \in [1, n+m]} a_i v_i\right) \leqslant b$$

Over-approximating Logico-numerical State Spaces

Over-approximating Numerical Spaces ($e.g., \mathcal{X} = \mathbb{Z}^n \times \mathbb{R}^m$)

Numerical Abstract Domains

- Provide α, γ and $\mathcal{N}$ such that $\wp(\mathbb{Z}^n \times \mathbb{R}^m) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{N}$

e.g.,

- *Intervals*: Conjunction of $n + m$ Bound Constraints of the form:
$$(a_i \leqslant v_i \leqslant b_i)$$
- *Convex Polyhedra*: Conjunction of k Linear Constraints of the form:
$$\left(\sum_{i \in [1, n+m]} a_i v_i\right) \leqslant b$$

Over-approximating Logico-numerical Spaces ($e.g., \mathcal{X} = \mathbb{B}^n \times \mathbb{Z}^m \times \mathbb{R}^o$)

Combining a Boolean Domain and a Numerical Abstract Domain $\mathcal{N}$

Over-approximating Logico-numerical State Spaces

Over-approximating Numerical Spaces ($e.g., \mathcal{X} = \mathbb{Z}^n \times \mathbb{R}^m$)

Numerical Abstract Domains

- Provide α, γ and $\mathcal{N}$ such that $\wp(\mathbb{Z}^n \times \mathbb{R}^m) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{N}$

e.g.,

- *Intervals*: Conjunction of $n + m$ Bound Constraints of the form:
$$(a_i \leqslant v_i \leqslant b_i)$$
- *Convex Polyhedra*: Conjunction of k Linear Constraints of the form:
$$\left(\sum_{i \in [1,n+m]} a_i v_i\right) \leqslant b$$

Over-approximating Logico-numerical Spaces ($e.g., \mathcal{X} = \mathbb{B}^n \times \mathbb{Z}^m \times \mathbb{R}^o$)

Combining a Boolean Domain and a Numerical Abstract Domain $\mathcal{N}$

e.g.,

- *Power*: $\wp(\mathbb{B}^n \times \mathbb{Z}^m \times \mathbb{R}^o) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathbb{B}^n \to \mathcal{N}$ $(= \mathcal{N}^{\mathbb{B}^n})$

ReaX: Implementation Details

Extension of ReaVer[2]

- Combining *Decision Diagrams* and *Numerical Abstract Domains* with BddApron[3]
  - Numerical Abstract Domains: APRON[4]
    - Intervals, Convex Polyhedra...
  - Decision Diagrams: CUDD[5]
- Heptagon/BZR Backend[6]

---

[2] Peter Schrammel. "Logico-Numerical Verification Methods for Discrete and Hybrid Systems". PhD thesis. University of Grenoble, 2012.

[3] Bertrand Jeannet. *BddApron: A logico-numerical abstract domain library*. 2009. URL: http://pop-art.inrialpes.fr/~bjeannet/bjeannet-forge/bddapron/.

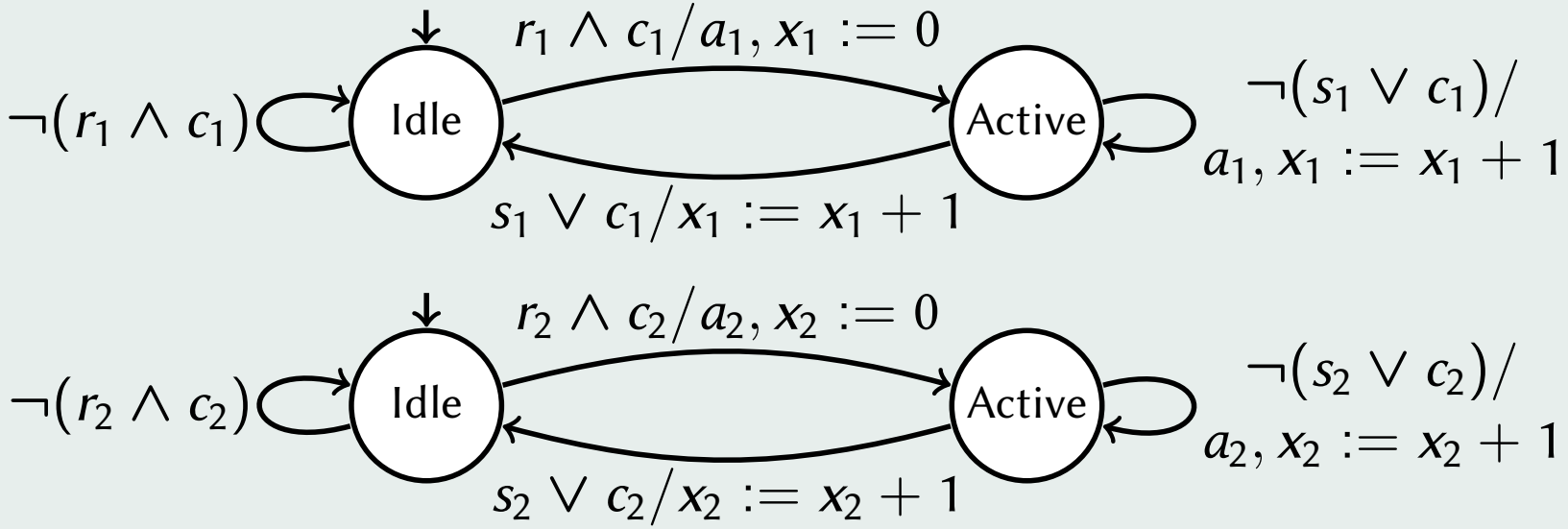
[4] Bertrand Jeannet and Antoine Miné. "APRON: A Library of Numerical Abstract Domains for Static Analysis". In: *Proceedings of the 21st International Conference on Computer Aided Verification*. CAV '09. Grenoble, France: Springer-Verlag, 2009, pp. 661–667. URL: http://apron.cri.ensmp.fr/library/.

[5] http://vlsi.colorado.edu/~fabio/CUDD/

[6] http://bzr.inria.fr/

Example Safety Control Problems for Infinite System

Example Infinite State System & Invariants

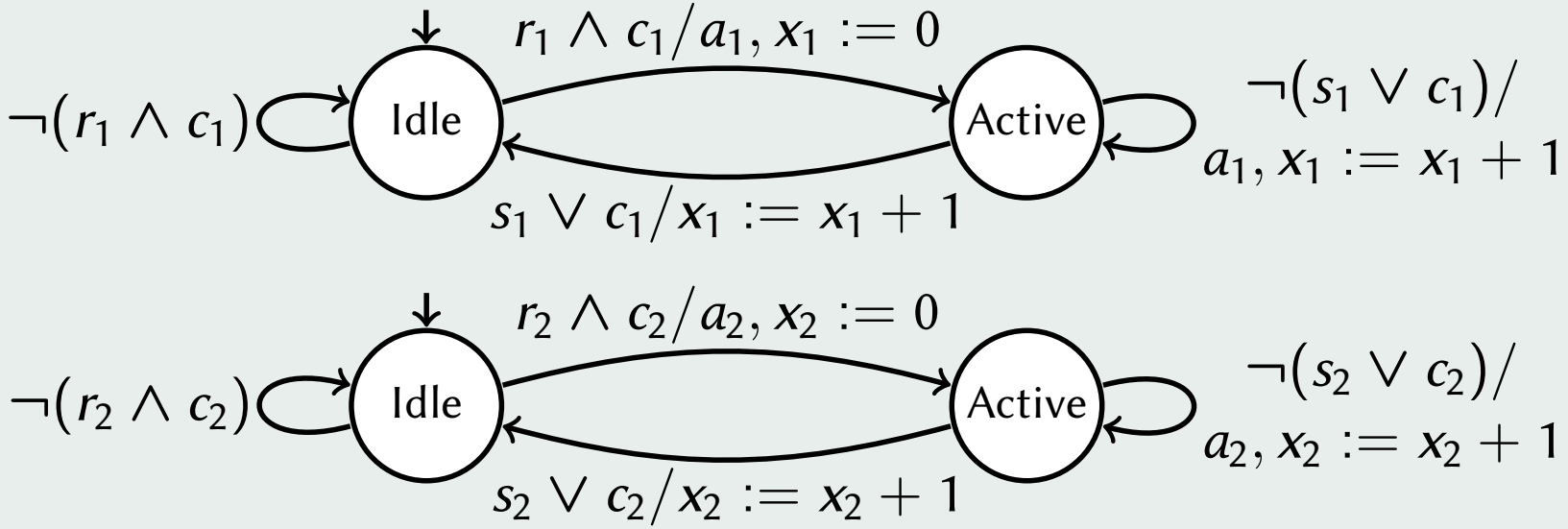


- $X = \langle t_1, t_2, x_1, x_2, a_1, a_2 \rangle$ $\quad \mathcal{D}_X = \{\text{Idle}, \text{Active}\}^2 \times \mathbb{Z}^2 \times \mathbb{B}^2$
- $I_{uc} = \langle r_1, r_2, s_1, s_2 \rangle, I_c = \langle c_1, c_2 \rangle$ $\quad \mathcal{D}_{I_{uc}} = \mathbb{B}^4, \mathcal{D}_{I_c} = \mathbb{B}^2$
- Enforcing Mutual Exclusion Between Active States

$$\Phi(\langle t_1, t_2, x_1, x_2 \ldots \rangle) = (t_1 \neq t_2 \vee t_1 = \text{Idle})$$

Example Safety Control Problems for Infinite System

Example Infinite State System & Invariants

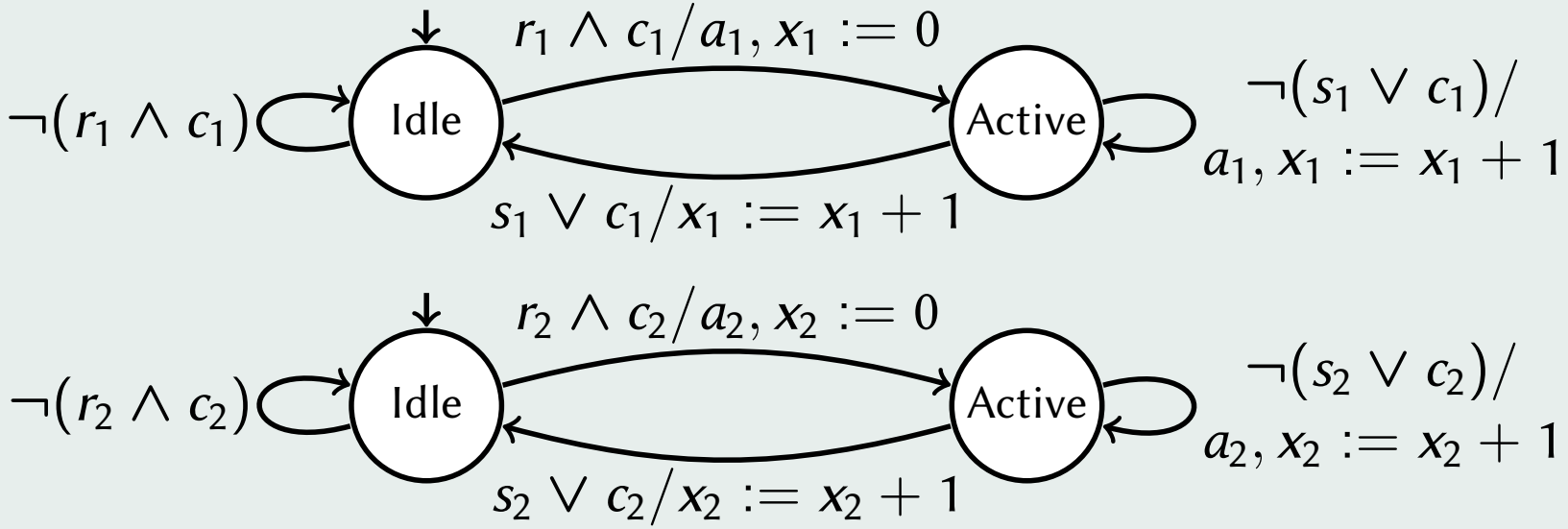


- $X = \langle t_1, t_2, x_1, x_2, a_1, a_2 \rangle$ $\quad \mathcal{D}_X = \{\text{Idle}, \text{Active}\}^2 \times \mathbb{Z}^2 \times \mathbb{B}^2$
- $I_{uc} = \langle r_1, r_2, s_1, s_2 \rangle, I_c = \langle c_1, c_2 \rangle$ $\quad \mathcal{D}_{I_{uc}} = \mathbb{B}^4, \mathcal{D}_{I_c} = \mathbb{B}^2$
- Enforcing Mutual Exclusion Between Active States & Bounds on x_i's

$$\Phi(\langle t_1, t_2, x_1, x_2 \ldots \rangle) = (t_1 \neq t_2 \vee t_1 = \text{Idle}) \wedge (x_1 \leqslant 10 \wedge x_2 \leqslant 10)$$

Example Safety Control Problems for Infinite System

Example Infinite State System & Invariants

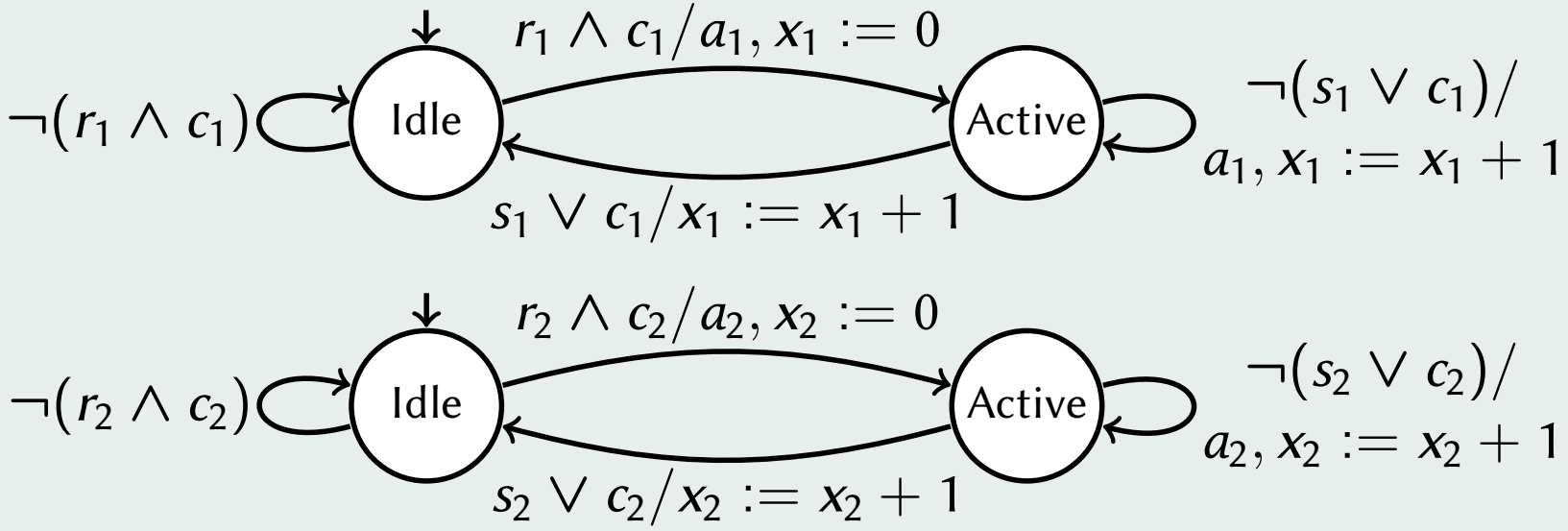


- $X = \langle t_1, t_2, x_1, x_2, a_1, a_2 \rangle$ $\quad \mathcal{D}_X = \{\text{Idle}, \text{Active}\}^2 \times \mathbb{Z}^2 \times \mathbb{B}^2$
- $I_{uc} = \langle r_1, r_2, s_1, s_2 \rangle$, $I_c = \langle c_1, c_2 \rangle$ $\quad \mathcal{D}_{I_{uc}} = \mathbb{B}^4, \mathcal{D}_{I_c} = \mathbb{B}^2$
- Enforcing Mutual Exclusion Between Active States & Bounds on x_i's & Relational Constraints on x_i's

$$\Phi(\langle t_1, t_2, x_1, x_2 \ldots \rangle) = (t_1 \neq t_2 \vee t_1 = \text{Idle}) \wedge (x_1 \leqslant 10 \wedge x_2 \leqslant 10) \wedge (x_1 \leqslant x_2)$$

Results of Over-approximating Algorithm

System



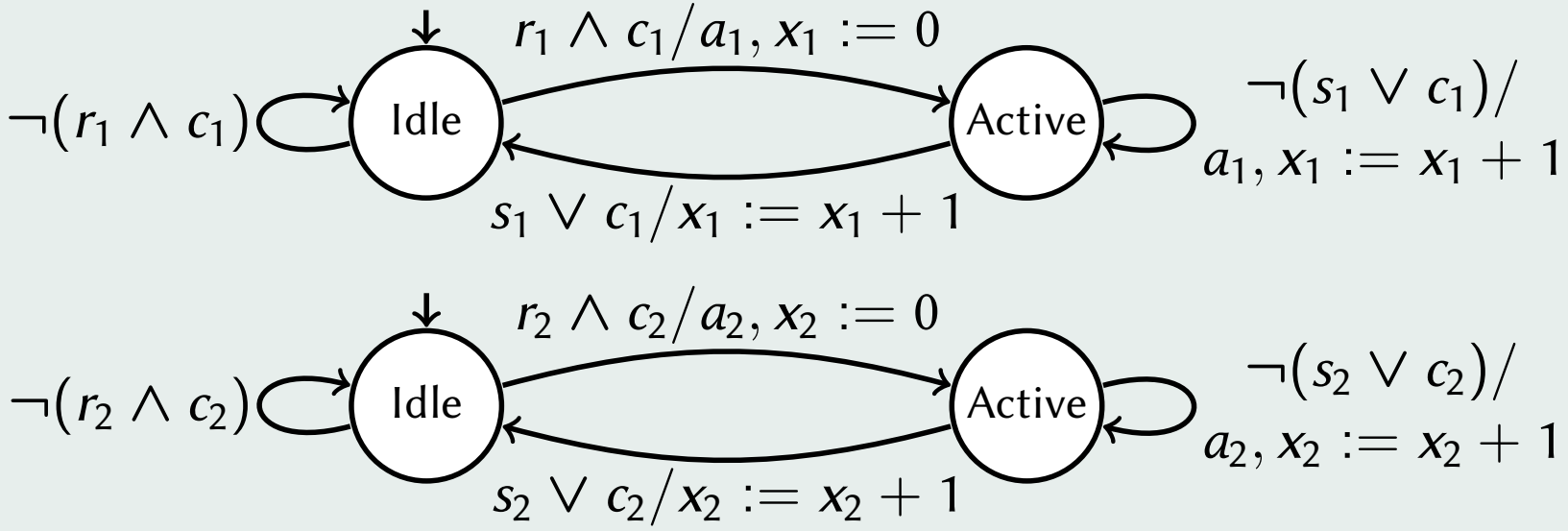
$\Phi(\langle t_1, t_2, x_1 \ldots \rangle) = (t_1 \neq t_2 \vee t_1 = \text{Idle}) \wedge (x_1 \leqslant 10 \wedge x_2 \leqslant 10) \wedge (x_1 \leqslant x_2)$

- Power Domain, Intervals

$$I'_{Bad} \supseteq \{t_1 = \text{Idle} \wedge t_2 = \text{Idle} \wedge x_1 < 11 \wedge x_2 < 10\}$$
$$\supseteq \{t_1 = \text{Idle} \wedge t_2 = \text{Idle} \wedge x_1 = 0 \wedge x_2 = 0\} = \mathcal{X}_0$$

Results of Over-approximating Algorithm

System



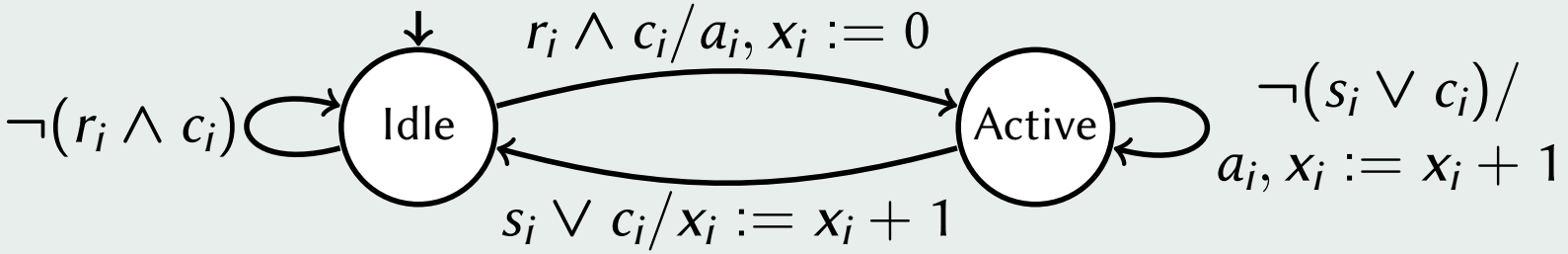
$\Phi(\langle t_1, t_2, x_1 \ldots \rangle) = (t_1 \neq t_2 \vee t_1 = \text{Idle}) \wedge (x_1 \leqslant 10 \wedge x_2 \leqslant 10) \wedge (x_1 \leqslant x_2)$

- Power Domain, Convex Polyhedra

$$I'^{c}_{Bad} = \left\{ \begin{array}{l} (t_1 = \text{Idle} \wedge t_2 = \text{Idle} \wedge x_1 \leqslant x_2 \leqslant 10) \vee \\ (t_1 = \text{Idle} \wedge t_2 = \text{Active} \wedge x_1 \leqslant x_2 \leqslant 9) \vee \\ (t_1 = \text{Active} \wedge t_2 = \text{Idle} \wedge x_1 < x_2 \leqslant 10) \end{array} \right\}$$

Performance Evaluation

Example Infinite State System & Invariants



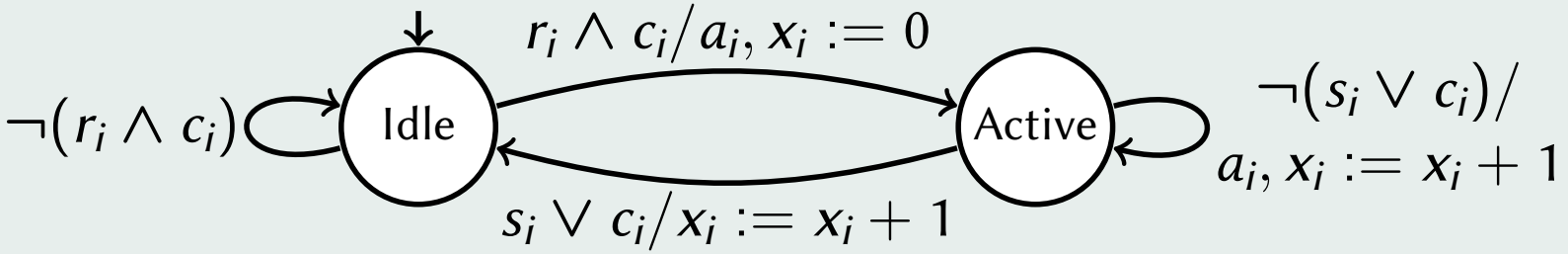
- $X = \langle t_1 \dots t_n, x_1 \dots x_n, a_1 \dots a_n \rangle$ $\quad \mathcal{D}_X = \{\text{Idle}, \text{Active}\}^n \times \mathbb{Z}^n \times \mathbb{B}^n$
- $I_{uc} = \langle r_1 \dots r_n, s_1 \dots s_n \rangle, I_c = \langle c_1 \dots c_n \rangle$ $\quad \mathcal{D}_{I_{uc}} = \mathbb{B}^{2n}, \mathcal{D}_{I_c} = \mathbb{B}^n$
- Enforcing Mutual Exclusion Between Active States & Bounds on x_i's

$$\Phi(\langle t_1 \dots t_n, x_1 \dots x_n, \dots \rangle) =$$

$$\bigoplus_{i \in [1,n]} (t_i = \text{Active}) \wedge \bigwedge_{i \in [1,n]} (x_i \leqslant 10)$$

Performance Evaluation

Example Infinite State System & Invariants



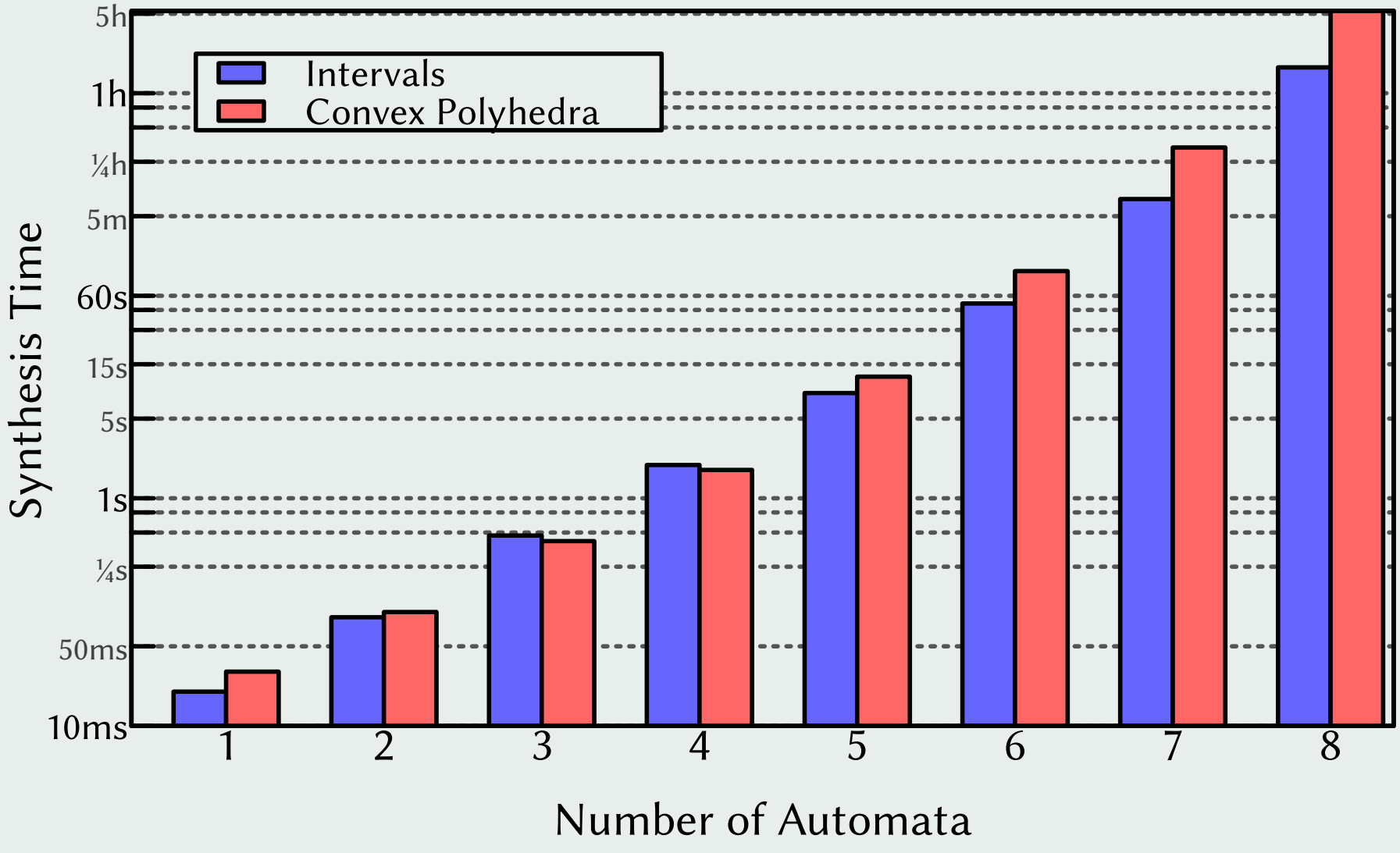
- $X = \langle t_1 \dots t_n, x_1 \dots x_n, a_1 \dots a_n \rangle$ $\quad \mathcal{D}_X = \{\text{Idle}, \text{Active}\}^n \times \mathbb{Z}^n \times \mathbb{B}^n$
- $I_{uc} = \langle r_1 \dots r_n, s_1 \dots s_n \rangle, I_c = \langle c_1 \dots c_n \rangle$ $\quad \mathcal{D}_{I_{uc}} = \mathbb{B}^{2n}, \mathcal{D}_{I_c} = \mathbb{B}^n$
- Enforcing Mutual Exclusion Between Active States & Bounds on x_i's & Relational Constraints on x_i's

$$\Phi(\langle t_1 \dots t_n, x_1 \dots x_n, \dots \rangle) =$$

$$\bigoplus_{i \in [1,n]} (t_i = \text{Active}) \wedge \bigwedge_{i \in [1,n]} (x_i \leqslant 10) \wedge \bigwedge_{i \in [1,n[} (x_i \leqslant x_{i+1})$$

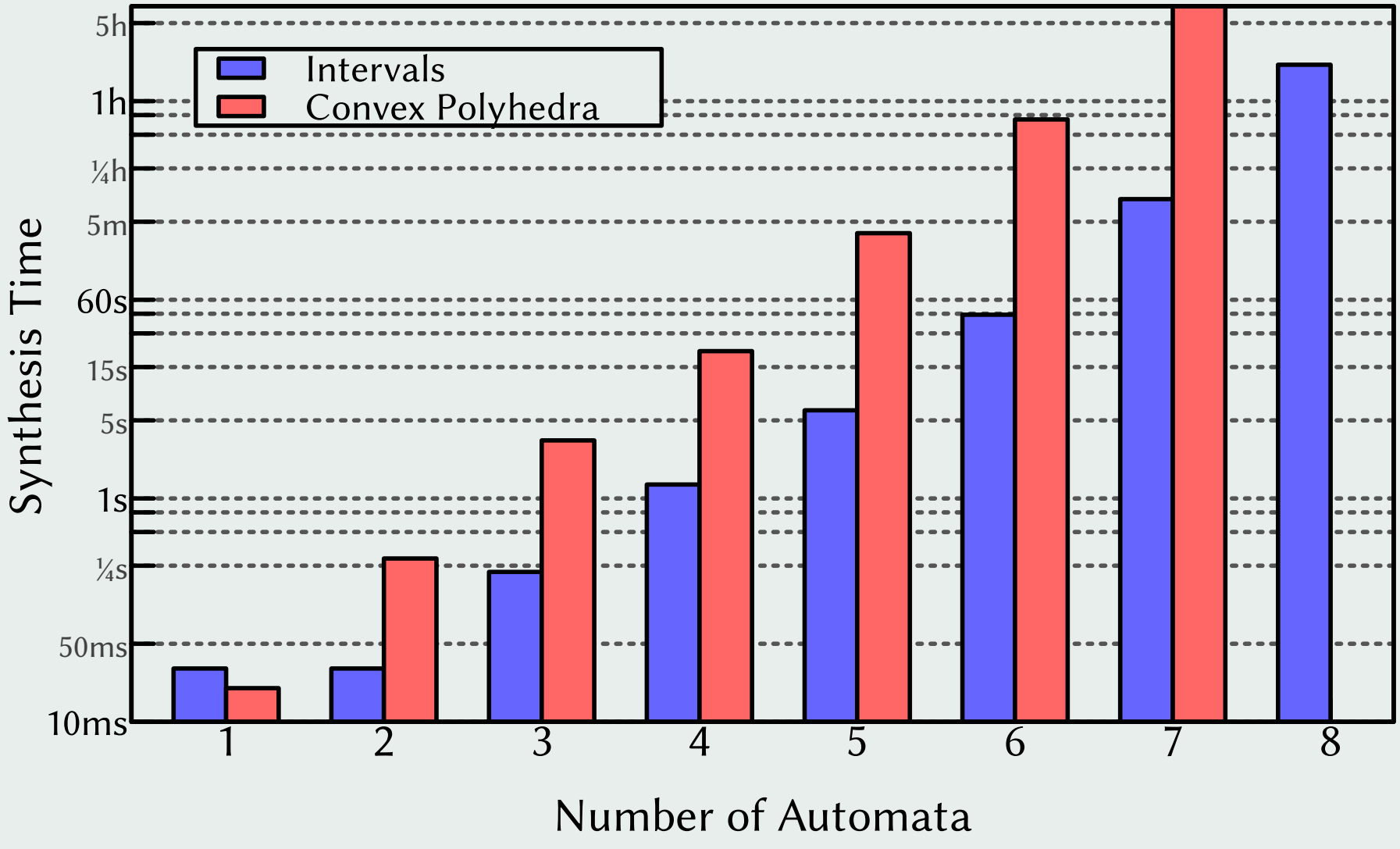
Performance Evaluation (cont'd)

Results for Mutual Exclusion & Bounds



Performance Evaluation (cont'd)

Results for Mutual Exclusion & Bounds & Relational Constraints



Outline

- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
- ReaX: Technical Choices, Implementation & Evaluations
- Conclusions

Conclusion & Further Works

Overall Contributions

- Algorithm for the Safety Control Problem of Infinite State Systems
  - Using Abstract Interpretation Techniques
- Efficient Synthesis in the Finite Case
- Heptagon/BZR Backend

$\rightsquigarrow$ Favorably Replaces SIGALI

Conclusion & Further Works

Overall Contributions

- Algorithm for the Safety Control Problem of Infinite State Systems
  - Using Abstract Interpretation Techniques
- Efficient Synthesis in the Finite Case
- Heptagon/BZR Backend

$\rightsquigarrow$ Favorably Replaces SIGALI

Forthcoming Challenges

- Synthesis Failure Diagnosis
- Avoiding Deadlocks
- Improving Precision
  - Dynamic Partitioning
  - Abstract Acceleration (Avoids Widening)

Outline

- Infinite State Systems (ASTSs)
- Safety Control Problem for ASTSs
- Principles of the Solution
- ReaX: Technical Choices, Implementation & Evaluations
- Conclusions