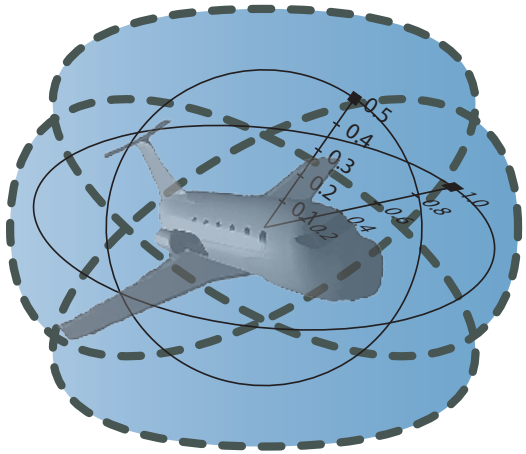
15-819/18-879: Logical Analysis of Hybrid Systems

10: Hybrid Systems Model Checking

André Platzer

aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



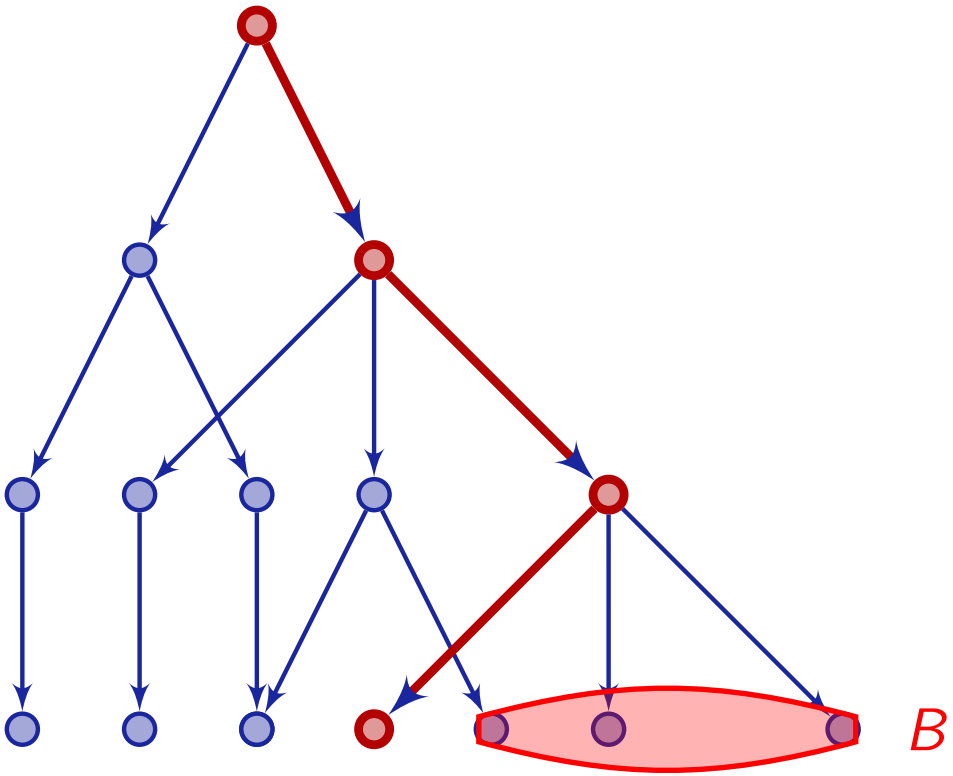
Outline

Outline

Model Checking in a Nutshell

Definition (Model Checking Problem)

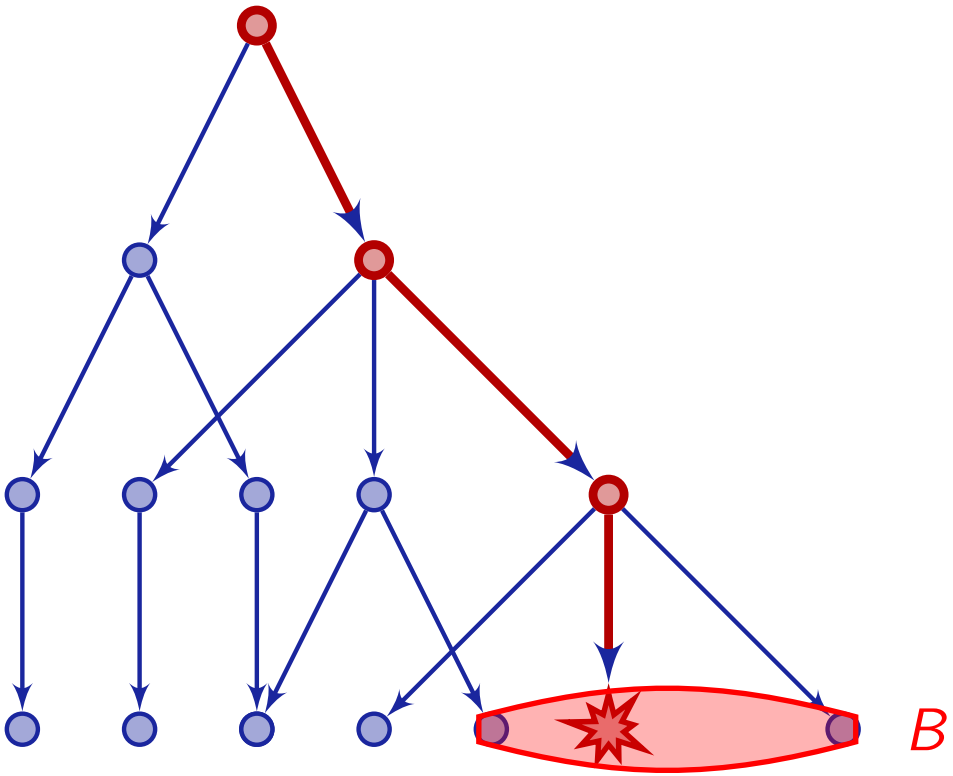
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



Model Checking in a Nutshell

Definition (Model Checking Problem)

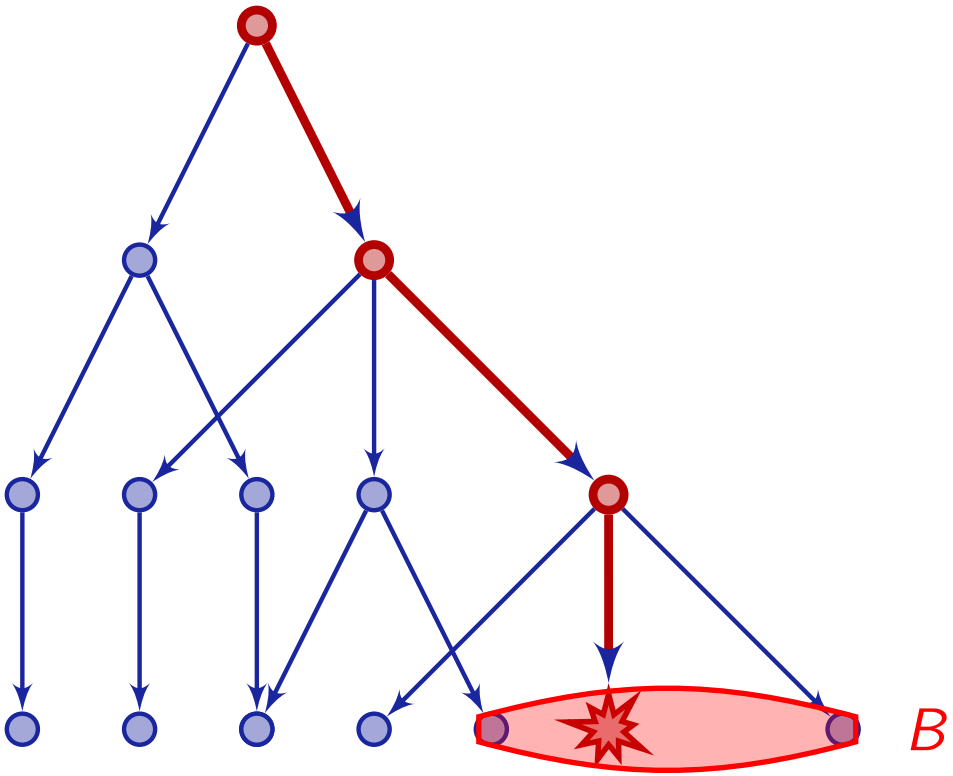
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



Model Checking in a Nutshell

Definition (Image Computation)

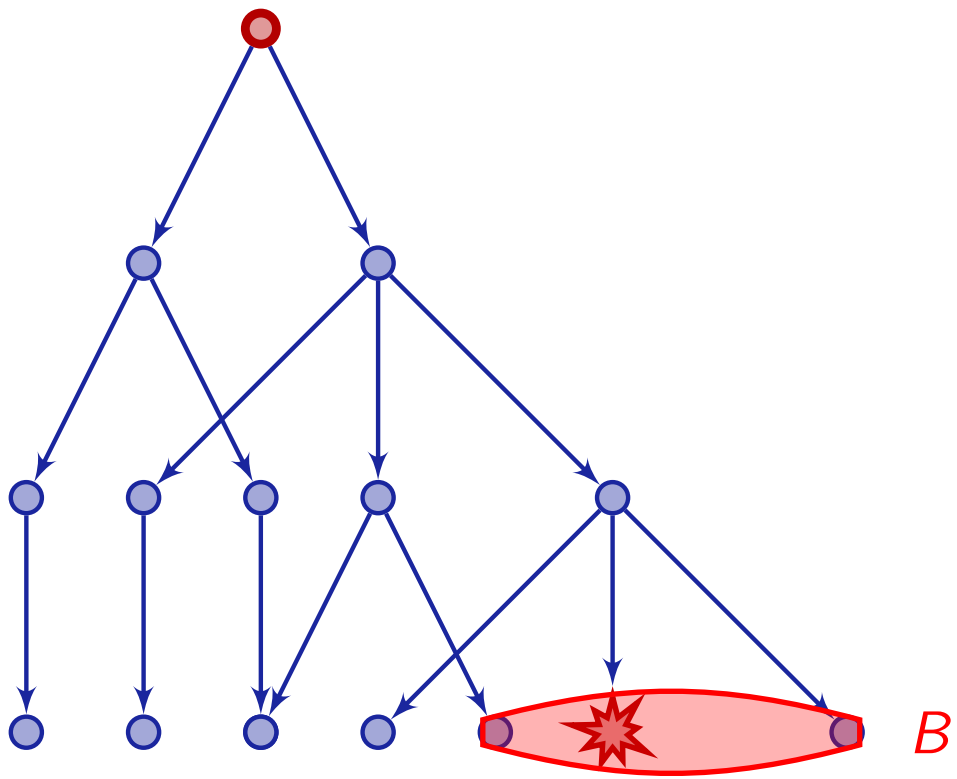
$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Model Checking in a Nutshell

Definition (Image Computation)

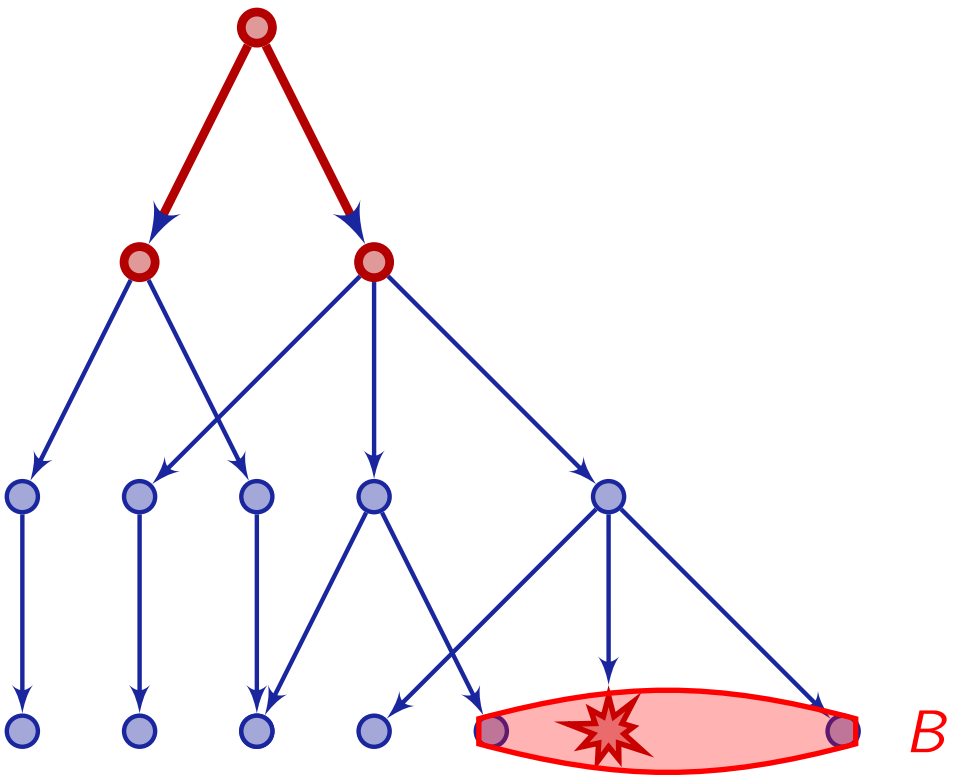
$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Q_0

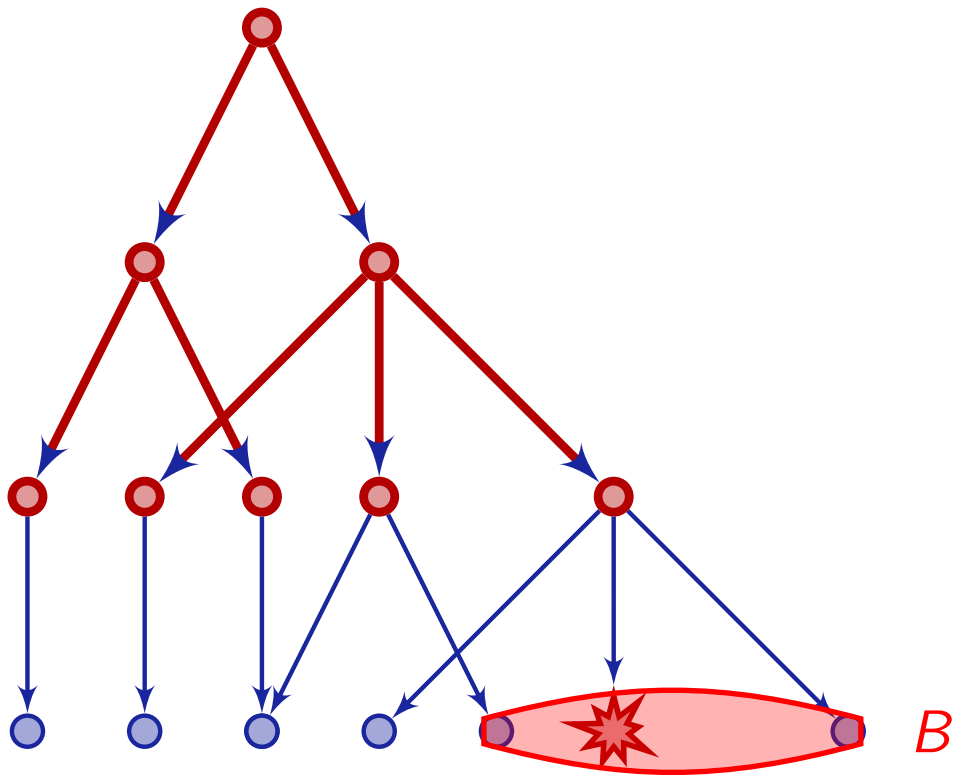
$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Q_0

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

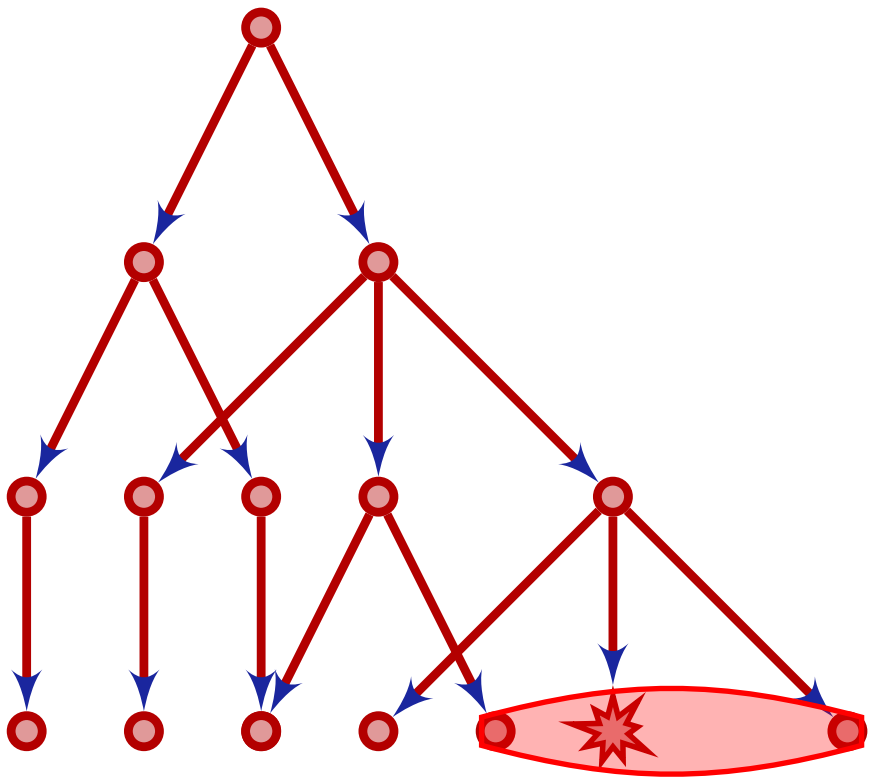
$Post_A(Q_1)$

$Q_2 = Post_A^2(Q_0)$

Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



$$Q_0$$

$$Post_A(Q_0)$$

$$Q_1 = Post_A(Q_0)$$

$$Post_A(Q_1)$$

$$Q_2 = Post_A^2(Q_0)$$

$$Post_A(Q_2)$$

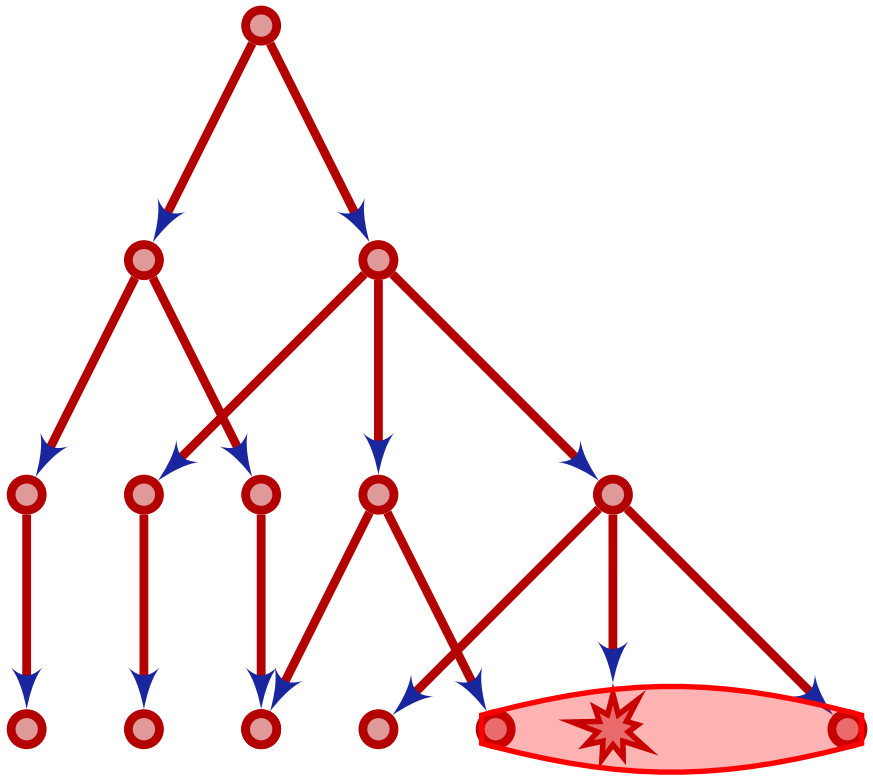
$$Q_3 = Post_A^3(Q_0)$$

Model Checking in a Nutshell

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$

$$Post_A^*(Y) := \bigcup_{n \in \mathbb{N}} Post_A^n(Y) = \mu Z.(Y \cup Z \cup Post_A(Z))$$



Q_0

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

$Post_A(Q_1)$

$Q_2 = Post_A^2(Q_0)$

$Post_A(Q_2)$

$Q_3 = Post_A^3(Q_0)$

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Can we use this for hybrid systems?

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure.

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Proposition (Semidecision)

For (computable) countably infinite-state systems, naïve MC gives a (slow) semidecision procedure.

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Proposition (Semidecision)

For (computable) countably infinite-state systems, naïve MC gives a (slow) semidecision procedure. Faster algorithms depend on problem

Model Checking in a Nutshell

Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.

Proposition (Decision)

For finite-state systems, this naïve MC algorithm gives a (slow) decision procedure. Faster algorithms exist with OBDD, BMC, ...

Proposition (Semidecision)

For (computable) countably infinite-state systems, naïve MC gives a (slow) semidecision procedure. Faster algorithms depend on problem

Hybrid systems have uncountable state spaces

(Uncountably) infinite state spaces require extra care

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q, x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q, x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q, x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q, x) \in Y, x^+ = \varphi_q(r, x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r, x)$ is the state reached after r time when starting in x.

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q,x) \in Y, x^+ = \varphi_q(r,x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r,x)$ is the state reached after r time when starting in x.
- φ needs to be computable for this!

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q, x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q, x) \in Y, x^+ = \varphi_q(r, x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r, x)$ is the state reached after r time when starting in x.
- φ needs to be computable for this!
- A needs to be finite for this!

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q, x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q, x) \in Y, x^+ = \varphi_q(r, x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r, x)$ is the state reached after r time when starting in x.
- φ needs to be computable for this!
- A needs to be finite for this! But $A = E \cup \mathbb{R}_{\geq 0}$.

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q,x) \in Y, x^+ = \varphi_q(r,x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r,x)$ is the state reached after r time when starting in x.
- φ needs to be computable for this!
- A needs to be finite for this! But $A = E \cup \mathbb{R}_{\geq 0}$.
  Discretize $A := E \cup \Delta$ for $\Delta := \{0, 0.5, 1\}$ or $\Delta := \{0.5\}$

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$ What if nondeterministic $x_1 := *$?
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q,x) \in Y, x^+ = \varphi_q(r,x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r,x)$ is the state reached after r time when starting in x.
- φ needs to be computable for this!
- A needs to be finite for this! But $A = E \cup \mathbb{R}_{\geq 0}$.
  Discretize $A := E \cup \Delta$ for $\Delta := \{0, 0.5, 1\}$ or $\Delta := \{0.5\}$
- Y rarely stays finite when repeating $Y := Post_A(Y)$

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is finite (a lot easier!)
- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
  What if nondeterministic $x_1 := *$?
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q,x) \in Y, x^+ = \varphi_q(r,x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r,x)$ is the state reached after r time when starting in x.
- φ needs to be computable for this!
- A needs to be finite for this! But $A = E \cup \mathbb{R}_{\geq 0}$.
  Discretize $A := E \cup \Delta$ for $\Delta := \{0, 0.5, 1\}$ or $\Delta := \{0.5\}$
- Y rarely stays finite when repeating $Y := Post_A(Y)$
- How check evolution domain restriction on $\varphi_q(t,x)$ for all $0 \leq t \leq r$?

Simple Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

× Consider the case where $Y \subseteq Q$ is finite (a lot easier!)

- Compute $Post_e(Y)$ for discrete action $e \in A$?
- $Post_e(Y) = \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$
  easy to compute if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
  What if nondeterministic $x_1 := *$?
- Compute $Post_r(Y)$ for continuous action $r \in \mathbb{R}_{\geq 0}$?
- $Post_r(Y) = \{(q, x^+) \ : \ \exists (q,x) \in Y, x^+ = \varphi_q(r,x)\}$
  when φ is the solution (flow) for mode q, i.e., $\varphi_q(r,x)$ is the state reached after r time when starting in x.

× φ needs to be computable for this!

× A needs to be finite for this! But $A = E \cup \mathbb{R}_{\geq 0}$.
Discretize $A := E \cup \Delta$ for $\Delta := \{0, 0.5, 1\}$ or $\Delta := \{0.5\}$

× Y rarely stays finite when repeating $Y := Post_A(Y)$

× How check evolution domain restriction on $\varphi_q(t,x)$ for all $0 \leq t \leq r$?

- Simulation-style $Post_A(Y)$ simple but more problems than solutions.

HS BMC: Numerical Finite Discrete-Time

Discrete-time finite-state numerical bounded model checking

Approximate $Post_A^*(Y)$ for finite $Y \subseteq Q \times \mathbb{Q}^n$ on a grid $A := E \cup \{\Delta\}$

1. $Post_e(Y) := \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x \models guard_e, (x, x^+) \models reset_e\}$ for edge $e \in E$ going from q to q^+
2. $Post_\Delta(Y) := \{(q, \varphi_q(\Delta, x)) \ : \ (q,x) \in Y, \varphi_q(\Delta, x) \models inv_q\}$ hoping that $\varphi_q(t,x) \models inv_q$ for all $0 \leq t \leq \Delta$ assuming that $\varphi_q(t,x)$ is computable
3. $Y := Post_A(Y) := Post_\Delta(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

HS BMC: Numerical Finite Discrete-Time

Discrete-time finite-state numerical bounded model checking

Approximate $Post_A^*(Y)$ for finite $Y \subseteq Q \times \mathbb{Q}^n$ on a grid $A := E \cup \{\Delta\}$

1. $Post_e(Y) := \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x \models guard_e, (x, x^+) \models reset_e\}$ for edge $e \in E$ going from q to q^+
2. $Post_\Delta(Y) := \{(q, \varphi_q(\Delta, x)) \ : \ (q,x) \in Y, \varphi_q(\Delta, x) \models inv_q\}$ hoping that $\varphi_q(t,x) \models inv_q$ for all $0 \leq t \leq \Delta$ assuming that $\varphi_q(t,x)$ is computable
3. $Y := Post_A(Y) := Post_\Delta(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

✓ Very easy to implement

HS BMC: Numerical Finite Discrete-Time

Discrete-time finite-state numerical bounded model checking

Approximate $Post^*_A(Y)$ for finite $Y \subseteq Q \times \mathbb{Q}^n$ on a grid $A := E \cup \{\Delta\}$

1. $Post_e(Y) := \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x \models guard_e, (x, x^+) \models reset_e\}$ for edge $e \in E$ going from q to q^+
2. $Post_\Delta(Y) := \{(q, \varphi_q(\Delta, x)) \ : \ (q,x) \in Y, \varphi_q(\Delta, x) \models inv_q\}$ hoping that $\varphi_q(t,x) \models inv_q$ for all $0 \leq t \leq \Delta$ assuming that $\varphi_q(t,x)$ is computable
3. $Y := Post_A(Y) := Post_\Delta(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

✓ Very easy to implement

× Not sound (no problem found does not mean safe)

HS BMC: Numerical Finite Discrete-Time

Discrete-time finite-state numerical bounded model checking

Approximate $Post^*_A(Y)$ for finite $Y \subseteq Q \times \mathbb{Q}^n$ on a grid $A := E \cup \{\Delta\}$

1. $Post_e(Y) := \{(q^+, x^+) \ : \ \exists (q,x) \in Y, x \models guard_e, (x, x^+) \models reset_e\}$ for edge $e \in E$ going from q to q^+
2. $Post_\Delta(Y) := \{(q, \varphi_q(\Delta, x)) \ : \ (q,x) \in Y, \varphi_q(\Delta, x) \models inv_q\}$ hoping that $\varphi_q(t,x) \models inv_q$ for all $0 \leq t \leq \Delta$ assuming that $\varphi_q(t,x)$ is computable
3. $Y := Post_A(Y) := Post_\Delta(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

✓ Very easy to implement

× Not sound (no problem found does not mean safe)

× Not complete (does not find all bugs)

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?
- Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?
- Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y
- Let Y be defined by $\text{FOL}_{\mathbb{R}}$ formula F_Y, i.e., $Y = \{a \in \mathbb{R}^n \ : \ a \models F_Y\}$ and evolution domain D by formula F_D

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?
- Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y
- Let Y be defined by $\text{FOL}_{\mathbb{R}}$ formula F_Y, i.e., $Y = \{a \in \mathbb{R}^n \ : \ a \models F_Y\}$ and evolution domain D by formula F_D
- Consider the case where $\varphi_t(x)$ is a polynomial.

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?
- Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y
- Let Y be defined by $\text{FOL}_{\mathbb{R}}$ formula F_Y, i.e., $Y = \{a \in \mathbb{R}^n \ : \ a \models F_Y\}$ and evolution domain D by formula F_D
- Consider the case where $\varphi_t(x)$ is a polynomial.
- "$z \in Post_{p|_D}(Y)$" for evolution domain $D := inv_q$ is definable as:

$$\exists x \, \exists t \geq 0 \, (F_Y(x) \wedge \forall 0 \leq s \leq t \, F_D(s, p(s,x)) \wedge z = p(t,x))$$

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?
- Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y
- Let Y be defined by $\text{FOL}_{\mathbb{R}}$ formula F_Y, i.e., $Y = \{a \in \mathbb{R}^n \ : \ a \models F_Y\}$ and evolution domain D by formula F_D
- Consider the case where $\varphi_t(x)$ is a polynomial.
- "$z \in Post_{p|_D}(Y)$" for evolution domain $D := inv_q$ is definable as:

$$\exists x \, \exists t{\geq}0 \, (F_Y(x) \wedge \forall 0{\leq}s{\leq}t \, F_D(s, p(s,x)) \wedge z = p(t,x))$$

- $Post_e(Y) = \{(q^+, x^+) \ : \ (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$ easy to define if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?
- Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y
- Let Y be defined by $\text{FOL}_{\mathbb{R}}$ formula F_Y, i.e., $Y = \{a \in \mathbb{R}^n \ : \ a \models F_Y\}$ and evolution domain D by formula F_D
- Consider the case where $\varphi_t(x)$ is a polynomial.
- "$z \in Post_{p|_D}(Y)$" for evolution domain $D := inv_q$ is definable as:

$$\exists x \, \exists t{\geq}0 \, (F_Y(x) \wedge \forall 0{\leq}s{\leq}t \, F_D(s, p(s,x)) \wedge z = p(t,x))$$

- $Post_e(Y) = \{(q^+, x^+) \ : \ (q,x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$ easy to define if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Thus quantifier elimination can compute $Post_A(Y)$ in this case.

Hybrid Systems Model Checking

How to compute $Post_A(Y)$? Then $Post_A^*(Y)$ won't be long?

- Consider the case where $Y \subseteq Q$ is (infinite) set.
- How to represent such sets of states computationally?

? Consider polyhedra, ellipsoid, zonotope, semialgebraic set Y

- Let Y be defined by $\mathrm{FOL}_{\mathbb{R}}$ formula F_Y, i.e., $Y = \{a \in \mathbb{R}^n \ : \ a \models F_Y\}$ and evolution domain D by formula F_D

? Consider the case where $\varphi_t(x)$ is a polynomial.

- "$z \in Post_{p|_D}(Y)$" for evolution domain $D := inv_q$ is definable as:

$$\exists x \, \exists t{\geq}0 \, (F_Y(x) \wedge \forall 0{\leq}s{\leq}t \, F_D(s, p(s, x)) \wedge z = p(t, x))$$

- $Post_e(Y) = \{(q^+, x^+) \ : \ (q, x) \in Y, x_1 \geq 2, {x_1}^+ = x_1 + 5\}$ easy to define if e edge from q to q^+ labelled with guard $x_1 \geq 2$ and reset $x_1 := x_1 + 5$
- Thus quantifier elimination can compute $Post_A(Y)$ in this case.

HS BMC: Symbolic Continuous-Time

Continuous-time symbolic bounded model checking

Compute $Post_A^N(F_Y)$ for symbolic representation $F_Y : Q \to \mathrm{FOL}_{\mathbb{R}}$

1. $Post_{\mathbb{R}_{\geq 0}}(F_Y) := \{(q, G_q(F_Y(q))) \ : \ q \in Q\}$

   $G_q(L) := QE(\exists z\, \exists t{\geq}0\, (L(z) \wedge \forall 0{\leq}s{\leq}t\, F_D(s, \varphi_q(s,z)) \wedge x = \varphi_q(t,z)))$

   assuming that $\varphi_q(t,z)$ is polynomial and F_D represents inv_q.
2. $Post_e(F_Y) := \{(q^+, G_e(F_Y(q)))\}$ for edge $e \in E$ going from q to q^+

   $$G_e(L) := QE(\exists z\, (L(z) \wedge guard_e(z) \wedge reset_e(z,x)))$$
3. $Y := Post_A(Y) := Post_{\mathbb{R}_{\geq 0}}(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

HS BMC: Symbolic Continuous-Time

Continuous-time symbolic bounded model checking

Compute $Post_A^N(F_Y)$ for symbolic representation $F_Y : Q \to \mathrm{FOL}_{\mathbb{R}}$

1. $Post_{\mathbb{R}_{\geq 0}}(F_Y) := \{(q, G_q(F_Y(q))) \ : \ q \in Q\}$

   $G_q(L) := QE(\exists z\, \exists t{\geq}0\, (L(z) \wedge \forall 0{\leq}s{\leq}t\, F_D(s, \varphi_q(s,z)) \wedge x = \varphi_q(t,z)))$

   assuming that $\varphi_q(t,z)$ is polynomial and F_D represents inv_q.
2. $Post_e(F_Y) := \{(q^+, G_e(F_Y(q)))\}$ for edge $e \in E$ going from q to q^+

   $$G_e(L) := QE(\exists z\, (L(z) \wedge guard_e(z) \wedge reset_e(z,x)))$$
3. $Y := Post_A(Y) := Post_{\mathbb{R}_{\geq 0}}(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

✓ Not too terrible to implement

HS BMC: Symbolic Continuous-Time

Continuous-time symbolic bounded model checking

Compute $Post_A^N(F_Y)$ for symbolic representation $F_Y : Q \to \mathrm{FOL}_{\mathbb{R}}$

1. $Post_{\mathbb{R}_{\geq 0}}(F_Y) := \{(q, G_q(F_Y(q))) \ : \ q \in Q\}$

   $G_q(L) := QE(\exists z\, \exists t{\geq}0\, (L(z) \wedge \forall 0{\leq}s{\leq}t\, F_D(s, \varphi_q(s,z)) \wedge x = \varphi_q(t,z)))$

   assuming that $\varphi_q(t,z)$ is polynomial and F_D represents inv_q.
2. $Post_e(F_Y) := \{(q^+, G_e(F_Y(q)))\}$ for edge $e \in E$ going from q to q^+

   $$G_e(L) := QE(\exists z\, (L(z) \wedge guard_e(z) \wedge reset_e(z,x)))$$
3. $Y := Post_A(Y) := Post_{\mathbb{R}_{\geq 0}}(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

✓ Not too terrible to implement

× high complexity QE used very often

HS BMC: Symbolic Continuous-Time

Continuous-time symbolic bounded model checking

Compute $Post_A^N(F_Y)$ for symbolic representation $F_Y : Q \to \mathrm{FOL}_{\mathbb{R}}$

1. $Post_{\mathbb{R}_{\geq 0}}(F_Y) := \{(q, G_q(F_Y(q))) \ : \ q \in Q\}$

   $G_q(L) := QE(\exists z\, \exists t{\geq}0\, (L(z) \wedge \forall 0{\leq}s{\leq}t\, F_D(s, \varphi_q(s,z)) \wedge x = \varphi_q(t,z)))$

   assuming that $\varphi_q(t,z)$ is polynomial and F_D represents inv_q.
2. $Post_e(F_Y) := \{(q^+, G_e(F_Y(q)))\}$ for edge $e \in E$ going from q to q^+

   $$G_e(L) := QE(\exists z\, (L(z) \wedge guard_e(z) \wedge reset_e(z,x)))$$
3. $Y := Post_A(Y) := Post_{\mathbb{R}_{\geq 0}}(Y) \cup \bigcup_{e \in E} Post_e(Y)$
4. Repeat until some finite number of steps (bounded model checking)

✓ Not too terrible to implement

× high complexity QE used very often

× Not sound (no problem found does not mean safe)