Hybrid Control and Switched Systems

Lecture #6
Reachability

João P. Hespanha
University of California
at Santa Barbara

UCSB

Summary

Review of previous lecture
Reachability
• transition systems
• reachability algorithm
• backward reachability algorithm
• invariance algorithm
• controller design based on backward reachability
**Sequence Properties** (signals)

\[ X_{\text{sig}} \equiv \text{set of all piecewise continuous signals}\]
\[ x : [0, T) \rightarrow \mathbb{R}^n, \ T \in (0, \infty) \]

\[ Q_{\text{sig}} \equiv \text{set of all piecewise constant signals}\]
\[ q : [0, T) \rightarrow \mathbb{Q}, \ T \in (0, \infty) \]

**Sequence property** \( \equiv p : Q_{\text{sig}} \times X_{\text{sig}} \rightarrow \{\text{false, true}\} \)

E.g.,
\[
p(q, x) = \begin{cases} 
\text{true} & q(t) \in \{1, 3\}, \ x(t) \geq x(t + 3), \ \forall t \\
\text{false} & \text{otherwise} 
\end{cases}
\]

A pair of signals \((q, x) \in Q_{\text{sig}} \times X_{\text{sig}}\) satisfies \(p\) if \(p(q, x) = \text{true}\)

A hybrid automaton \(H\) satisfies \(p\) (write \(H \models p\)) if
\[ p(q, x) = \text{true}, \quad \text{for every solution } (q, x) \text{ of } H \]

**Sequence analysis** \(\equiv\) Given a hybrid automaton \(H\) and a sequence property \(p\)
show that \(H \models p\)
When this is not the case, find a witness
\((q, x) \in Q_{\text{sig}} \times X_{\text{sig}}\) such that \(p(q, x) = \text{false}\)

(in general for solution starting on a given set of initial states \(\mathcal{H}_0 \subset Q \times \mathbb{R}^n\))

---

**Example #1: Bouncing ball**

\[ x_1 := y \]
\[ x_2 := \dot{y} \]
\[ x_1 \leq 0 \land x_2 < 0 \]
\[ \dot{x}_1 = x_2, \quad \dot{x}_2 = -g \]
\[ x_2 := -c x_2 \]

Assuming that \(x_1(0) \geq 0\), the hybrid automaton satisfies:

- \(\square \{ x_1 \geq 0 \}\) (short for \(\square \{ x_1(t) \geq 0 \}(0)\))
- \(\Diamond \{ x_1 = 0 \}\)
- \(\Box \Diamond \{ x_1 = 0 \}\)
- \(\Diamond \Box \{ x_1 < 1 \}\)

- \((\Box p)(t_0) \iff \forall t \geq t_0, p(t)\)
- \((\square p)(t_0) \iff \exists t \geq t_0, p(t)\)
- \((\square \Diamond p)(t_0) \iff \forall t_1 \geq t_0, \exists t \geq t_1, p(t)\)
- \((\Diamond \Box p)(t_0) \iff \exists t_1 \geq t_0, \forall t \geq t_1, p(t)\)
Safety properties

Given a signal \( x: [0, T) \rightarrow \mathbb{R}^n, T \in (0, \infty) \), \( x^*: [0, T^*) \rightarrow \mathbb{R}^n \) is called a prefix to \( x \)
if \( T^* \leq T \) \& \( x^*(t) = x(t) \ \forall \ t \in [0, T^*) \)

\textbf{safety property} \equiv \ a \\text{sequence property} \ p \ \text{that is:}

1. \textit{nonempty}, i.e., \( \exists (q, x) \) such that \( p(q, x) = \text{true} \)
2. \textit{prefix closed}, i.e., given signals \( (q, x) \)
   \[ p(q, x) \Rightarrow p(q^*, x^*) \]
   for every prefix \( (q^*, x^*) \) to \( (q, x) \)
3. \textit{limit closed}, i.e., given an infinite sequence of signals
   \( (q_1, x_1), (q_2, x_2), (q_3, x_3), \text{etc.} \)
each element satisfying \( p \) such that
   \( (q_k, x_k) \) is a prefix to \( (q_{k+1}, x_{k+1}) \) \ \forall \ k
then \( (q, x) := \lim_{k \to \infty} (q_k, x_k) \) also satisfies \( p \)

"Something bad never happens:"
1. nontrivial
2. a prefix to a good signal is always good
3. if something bad happens, it will happen in finite time

Examples

E.g., \( p(q, x) = \square (q(t), x(t)) \in \mathcal{F} \) where \( \mathcal{F} \subset Q \times \mathbb{R}^n \) is a nonempty set

\[ x_1 \text{ satisfies } p \]
\[ x_2 \text{ does not} \]

this is a safety property: nonempty, prefix closed, limit closed

Other safety properties:
\( p(q, x) = x(t) \geq 0 \ \forall \ t \) (closed \( \mathcal{F} \))
\( p(q, x) = x(t) > 0 \ \forall \ t \) (open \( \mathcal{F} \))

Nonsafety property:
\( p(q, x) = \inf x(t) > 0 \) (not of the form above; not limit closed, Why?)
Liveness properties

Given a signal $x:[0,T) \rightarrow \mathbb{R}^n$, $T \in (0,\infty]$, $x^*:[0,T^*) \rightarrow \mathbb{R}^n$ is called a prefix to $x$ if $T^* \leq T$ & $x^*(t) = x(t) \ \forall t \in [0,T^*)$

**liveness property** $\equiv$ a sequence property $p$ with the property that for every finite $(q^*,x^*) \in \Sigma_{\text{sig}} \times \Sigma_{\text{sig}}$ there is some $(q,x) \in \Sigma_{\text{sig}} \times \Sigma_{\text{sig}}$ such that:
1. $(q^*,x^*)$ is a prefix to $(q,x)$
2. $(q,x)$ satisfies $p$

“Something good can eventually happen:” for any sequence there is a good continuation.

E.g., $p(q,x) = \Diamond (q(t),x(t)) \in \mathcal{F}$ where $\mathcal{F} \subseteq \mathbb{Q} \times \mathbb{R}^n$ is a nonempty set
$p(q,x) = \Box \Diamond (q(t),x(t)) \in \mathcal{F}$ (always, eventually: $\forall t \geq t_0, \exists t \geq t$)
$p(q,x) = \Diamond \Box (q(t),x(t)) \in \mathcal{F}$ (eventually, always: $\exists t \geq t_0, \forall t \geq t$)

What does it mean?
$p(q,x) = \exists L > 0 \diamond \|x\| < L$
$p(q,x) = \forall \varepsilon > 0 \diamond \|x\| < \varepsilon$

very rich class, more difficult to verify

Completeness of liveness/safety

**Theorem 1**: If $p$ is both a liveness and a safety property then every $(q,x) \in \Sigma_{\text{sig}} \times \Sigma_{\text{sig}}$ satisfies $p$, i.e., $p$ is always true (trivial property)

**Theorem 2**: For every nonempty (not always false) sequence property $p$ there is a safety property $p_1$ and a liveness property $p_2$ such that:

$(q,x)$ satisfies $p$ if and only if $(q,x)$ satisfies both $p_1$ an $p_2$

Thus if we are able to verify safety and liveness properties we are able to verify any sequence property.

But sequence properties are not all we may be interested in…

“**ensemble properties**” $\equiv$ property of the whole family of solutions

e.g., stability (continuity with respect to initial conditions) is not a sequence property because by looking at each solution $(q,x)$ individually we cannot decide if the system is stable. Much more on this later…

Can one find sequence properties that guarantee that the system is stable or unstable?
Reachability

Given: hybrid automaton $H$:
\[
\dot{x} = f(q, x) \quad (q, x) = \Phi(q^-, x^-) \quad q(t) \in Q, \; x(t) \in \mathbb{R}^n, \; t \geq t_0
\]
set of initial states $\mathcal{H}_0 \subset Q \times \mathbb{R}^n$

$\text{Reach}_H(\mathcal{H}_0) \equiv \text{set of pairs } ((q_0, x_0)) \in Q \times \mathbb{R}^n \text{ for which there is a solution } (q, x) \text{ to } H \text{ for which:}$
1. $(q(t_0), x(t_0)) \in \mathcal{H}_0$ \hspace{2cm} starts in $\mathcal{H}_0$
2. $\exists \; t \geq t_0 : (q(t), x(t)) = (q_f, x_f)$ \hspace{7pt} passes through $(q_f, x_f)$

Invariant set $\equiv \text{set } S \subset Q \times \mathbb{R}^n$ for which $\text{Reach}_H(S) = S$
Reachability v.s. Safety

Given: hybrid automaton $H$:

$$\dot{x} = f(q, x) \quad (q, x) = \Phi(q^-, x^-) \quad q(t) \in Q, \ x(t) \in \mathbb{R}^n, \ t \geq t_0$$

set of initial states $\mathcal{H}_0 \subset Q \times \mathbb{R}^n$

$\text{Reach}_H(\mathcal{H}_0) \equiv$ set of pairs $(q_f, x_f) \in Q \times \mathbb{R}^n$ for which there is a solution $(q, x)$ to $H$ for which:

1. $(q(t_0), x(t_0)) \in \mathcal{H}_0$ \hspace{1cm} starts in $\mathcal{H}_0$
2. $\exists \ t \geq t_0 : (q(t), x(t)) = (q_f, x_f)$ \hspace{1cm} passes through $(q_f, x_f)$

$H$ satisfies a safety property

$p(q, x) = \square \ (q(t), x(t)) \in \mathcal{F}$

where $\mathcal{F} \subset Q \times \mathbb{R}^n$ is a nonempty set if and only if $

\text{Reach}_H(\mathcal{H}_0) \subset \mathcal{F}$

every point in every trajectory starting in $\mathcal{H}_0$ satisfies $p$

---

Reachability v.s. Safety

Given: hybrid automaton $H$:

$$\dot{x} = f(q, x) \quad (q, x) = \Phi(q^-, x^-) \quad q(t) \in Q, \ x(t) \in \mathbb{R}^n, \ t \geq t_0$$

set of initial states $\mathcal{H}_0 \subset Q \times \mathbb{R}^n$

$\text{Reach}_H(\mathcal{H}_0) \equiv$ set of pairs $(q_f, x_f) \in Q \times \mathbb{R}^n$ for which there is a solution $(q, x)$ to $H$ for which:

1. $(q(t_0), x(t_0)) \in \mathcal{H}_0$ \hspace{1cm} starts in $\mathcal{H}_0$
2. $\exists \ t \geq t_0 : (q(t), x(t)) = (q_f, x_f)$ \hspace{1cm} passes through $(q_f, x_f)$

Over-approximation to the reach set $\equiv$ any set $\mathcal{R}_{\text{over}}$ such that $\text{Reach}_H(\mathcal{H}_0) \subset \mathcal{R}_{\text{over}}$

To prove safety is enough to show that: $\mathcal{R}_{\text{over}} \subset \mathcal{F}$

every point in every trajectory starting in $\mathcal{H}_0$ satisfies $p$

Are under-approximations useful to study reachability?
**Transition system**

*generalization of finite automaton, differential equations, hybrid automaton, etc.*

\[
\begin{align*}
S & \equiv \text{set of states (finite or infinite)} \\
E & \equiv \text{alphabet of events (finite or infinite)} \\
T & \subseteq S \times E \times S \equiv \text{transition relation}
\end{align*}
\]

\[
\begin{array}{c}
1 \xrightarrow{a} 2 \\
2 \xrightarrow{b} 3 \\
3 \xrightarrow{a} 1
\end{array}
\]

\(S = \{1, 2, 3\}\)  
\(E = \{a, b\}\)  
\(T = \{(1,a,2), (2,b,1), (2,b,3), (3,a,1)\}\)

**execute** of a transition system \(\equiv\) sequence of states \(\{s_0, s_1, s_2, \ldots\}\) such that

- there exists a sequence of events \(\{e_0, e_1, e_2, \ldots\}\)
  - for which \((s_i, e_i, s_{i+1}) \in T\) \(\forall i\)

Given a set of initial states \(S_0 \subseteq S\):

\(\text{Reach}_T(S_0) \equiv\) set of states \(s \in S\) for which there is a finite execution that

- starts in \(S_0\) and ends at \(s\)

---

**Transition systems**

As far as reachability goes …

1. A finite automaton (deterministic or not) can be viewed as a transition system

\[
\begin{align*}
M & := \left\{ Q := \{q_1, q_2, \ldots, q_n\} \equiv \text{finite set of states} \\
& \quad \Sigma := \{a, b, c, \ldots\} \equiv \text{finite set of input symbols (alphabet)} \\
& \quad \Phi : Q \times \Sigma \to Q \equiv \text{transition function}
\end{align*}
\]

\[
\begin{align*}
T & := \left\{ (s, e, \Phi(s,e)) : s \in Q, e \in \Sigma \right\} \equiv \text{transition relation}
\end{align*}
\]

for nondeterministic finite automaton

\[
T = \left\{ (s, e, s') : s \in Q, e \in \Sigma, s' \in \Phi(s,e) \right\}
\]

Same set of reachable states
Transition systems

As far as reachability goes …

1. A hybrid automaton can be viewed as a transition system

\[ H \]

\[ \mathcal{Q} \equiv \text{set of discrete states} \]

\[ \mathbb{R}^n \equiv \text{continuous state-space} \]

\[ f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n \equiv \text{vector field} \]

\[ \Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n \equiv \text{discrete transition (\& reset map)} \]

**transition system**

\[ T \subseteq \mathcal{S} \times \mathcal{E} \times \mathcal{S} \equiv \text{transition relation} \]

\[ \{(q_0, x_0), (q_0, q_f), (q_f, x_f)\} \in T \text{ if } (x_f, q_f) = \Phi(x_0, q_0) \]

\[ \{(q_0, x_0), \tau, (q_0, y_j)\} \in T \text{ if } \exists t_f > 0 \text{ s.t. } \dot{x} = f(q_0, x), x(0) = x_0, x(t_f) = x_f \]

same \( (q_0, x_0) \) and \( \tau \) lead to many distinct elements in \( T \)

(flows modeled as nondeterminism)

---

Reachability algorithm

**Reachability algorithms:**

**initialization:**

\[ \text{Reach}_{i-1} = \emptyset \]

\[ \text{Reach}_0 = \mathcal{S}_0 \]

\[ i = 0 \]

\[ \text{states one can transition to from } \text{Reach}_i \]

**loop:**

while \( \text{Reach}_i \neq \text{Reach}_{i-1} \) do

\[ \text{Reach}_{i+1} = \text{Reach}_i \cup \{s' \in \mathcal{S} : \exists s \in \text{Reach}_i, e \in \mathcal{E}, (s, e, s') \in T\} \]

\[ i = i + 1 \]

---

\[ \mathcal{S}_0 = \{3\} \]

\[ \text{Reach}_0 = \{3\} \]

\[ \text{Reach}_1 = \{1, 3, 5, 6\} \]

\[ \text{Reach}_2 = \{1, 2, 3, 5, 6\} \]

\[ \text{Reach}_3 = \emptyset \]

\[ \text{Reach}_4 = \emptyset \]

\[ \text{Reach}_4(\{3\}) = \emptyset \]

\[ \mathcal{S}_0 = \{2\} \]

\[ \text{Reach}_0 = \{2\} \]

\[ \text{Reach}_1 = \{2, 4, 5\} \]

\[ \text{Reach}_2 = \{2, 4, 5\} \]

\[ \text{Reach}_3 = \{2, 4, 5\} \]

\[ \text{Reach}_4(\{2\}) = \{2, 4, 5\} \]
Reachability algorithm

Reachability algorithms:

initialization: Reach\(_i\) = \(\emptyset\)
Reach\(_0\) = \(S_0\)
i = 0

states one can transition to from Reach\(_i\)

loop: while Reach\(_i\) ≠ Reach\(_i\)-1 do

Reach\(_{i+1}\) = Reach\(_i\) \(\cup\) \{s' ∈ \(S\): \(\exists\) s ∈ Reach\(_i\), e ∈ \(E\), (s,e,s') ∈ \(T\)\}
i = i + 1

Theorem: If \(S\) is finite then
(i) the reachability algorithm finishes in a finite number of steps and (ii) upon exiting the while-loop Reach\(_i\) = Reach\(_i\)(\(S_0\))

Why?

(i) In each iteration the number of elements in Reach\(_i\) increases by at least 1. Since it can have, at most, as many elements as \(S\) there can only be as many iterations as the number of elements in \(S\) (minus the number of elements in \(S_0\)).
(ii) Reach\(_i\) \(\equiv\) the set of states that can be reached in \(i\) steps, thus any state that can be reached in a finite number of steps must be in one of the Reach\(_i\)
**Reachability algorithm**

**initialization:** Reach\(_i\) = \(\emptyset\)
Reach\(_0\) = \(S_0\)

**loop:**
while Reach\(_i\) ≠ Reach\(_i\)-1 do
Reach\(_i\) = Reach\(_i\) \cup \{s' \in S : \exists s \in \text{Reach}_i, e \in E, (s,e,s') \in T\}
i = i + 1

Two difficulties with hybrid automata
1. the set of states \(S := Q \times \mathbb{R}^n\) is not finite (algorithm may not terminate)
2. In the while loop: Reach\(_{i+1}\) = Reach\(_i\) \cup S_1 \cup S_2

Computation of
\(S_1 := \{s' \in S : \exists s \in \text{Reach}_i, e = (q_i,q_j) \in E, (s,e,s') \in T\}\) is simple
but
\(S_2 := \{s' \in S : \exists s \in \text{Reach}_i, e = \tau, (s,e,s') \in T\}\) is not (in general)

\(S_1 = \{(q_0{x}_f) \in S : \exists (q_0,{x}_0) \in \text{Reach}_i, (q_0,x_f) = \Phi(q_0,x_0)\}\)
\(S_2 = \{(q_0{x}_f) \in S : \exists (q_0,x_0) \in \text{Reach}_i, \text{there is a continuous evolution from } x_0 \text{ to } x_f \text{ inside mode } q_0\}\)

---

**Example #5: Tank system**

**goal** \(\equiv\) prevent the tank from emptying or filling up

\(\delta = .5 \equiv\) delay between command is sent to pump and the time it is executed

constant outflow \(\equiv\) \(\mu = 1\)
pump-on inflow \(\equiv\) \(\lambda = 3\)
pump

\(y \leq 1\) \(\implies\) pump off \((q = 1)\)

\[\dot{y} = -1, \quad \dot{s} = 0\]

wait to off \((q = 4)\)

\[\dot{y} = 3 - 1, \quad \dot{s} = 1\]

\(s := 0\)

\(y \geq 2\) \(\implies\) pump on \((q = 3)\)

\[\dot{y} = 3 - 1, \quad \dot{s} = 0\]

\(s \geq .5\) \(\implies\) wait to on \((q = 2)\)

\[\dot{y} = -1, \quad \dot{s} = 1\]

\(s := 0\)

\(y \geq .5\) \(\implies\) wait to off \((q = 4)\)

\[\dot{y} = 3 - 1, \quad \dot{s} = 1\]

\(s := 0\)

\(y \leq 1\) \(\implies\) pump off \((q = 1)\)

\[\dot{y} = -1, \quad \dot{s} = 0\]

wait to off \((q = 4)\)

\[\dot{y} = 3 - 1, \quad \dot{s} = 1\]

\(s := 0\)
Reachability algorithm for the tank system

Suppose $S_0 := \{(1,x) : x \in X_o\}$

Reachability algorithm for the tank system

Suppose $S_0 := \{(1,x) : x \in X_o\}$
Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{(1,x) : x \in \mathcal{X}_d\}$

Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{(1,x) : x \in \mathcal{X}_d\}$

Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{(1,x) : x \in \mathcal{X}_d\}$
Reachability algorithm for the tank system

Suppose \( \mathcal{S}_0 := \{ (1,x) : x \in \mathcal{X}_0 \} \)

\[
\begin{align*}
q &= 1 \\
y &= 1 \\
s &= 1 \\
\dot{y} &= 1 \\
\dot{s} &= 0
\end{align*}
\]

\[
\begin{align*}
q &= 4 \\
y &= 3 \\
s &= 0 \\
\dot{y} &= 3 \\
\dot{s} &= 1
\end{align*}
\]

Reach_4 = Reach_3 \cup \{(3, (5,5))\}

Reach_5 = Reach_4 \cup \{(3, (5,5+\alpha)) : \alpha \in [0,1.5] \}

Reach_6 = Reach_5 \cup \{(4, (2,0))\}

Suppose \( \mathcal{S}_0 := \{ (1,x) : x \in \mathcal{X}_0 \} \)

\[
\begin{align*}
q &= 3 \\
\dot{y} &= 2, \dot{s} = 0 \\
\end{align*}
\]

Reach_3 = Reach_2 \cup \{(3, (5,5))\}

Reach_4 = Reach_3 \cup \{(3, (5,5+\alpha)) : \alpha \in [0,1.5] \}

Reach_5 = Reach_4 \cup \{(3, (5,4))\}

\[
\begin{align*}
q &= 4 \\
\dot{y} &= 4, \dot{s} = 0 \\
\end{align*}
\]

Reach_6 = Reach_5 \cup \{(4, (2,0))\}
Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{ (1, x) : x \in \mathcal{X}_0 \}$

Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{ (1, x) : x \in \mathcal{X}_0 \}$
Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{ (1, x) : x \in \mathcal{X}_o \}$

Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{ (1, x) : x \in \mathcal{X}_o \}$
Reachability algorithm for the tank system

Suppose $\mathcal{S}_0 := \{ (1, x) : x \in \mathcal{X}_0 \}$

### Initialized Rectangular Automaton

**rectangle** $\equiv$ set of the form $I_1 \times I_2 \times \ldots \times I_n$ where each $I_i$ is an interval whose finite end-points are rational (in $\mathbb{Q}$)

e.g., $[3,4] \times [5,6]$ or $(-\infty, 1) \times (1,2)$ or $\mathbb{R} \times (1/2, 5/4)$

but not $[1,2] \cup [3,4] \times [5,6]$ or $[1,2^{1/2}] \times [3,4]$ 

**hybrid automata** $H$ $\begin{cases} 
\mathcal{Q} \equiv \text{set of discrete states} & \mathbb{R}^n \equiv \text{continuous state-space} \\
 f : \mathcal{Q} \times \mathbb{R}^n \rightarrow \mathbb{R}^n \equiv \text{vector field} \\
 \varphi : \mathcal{Q} \times \mathbb{R}^n \rightarrow \mathcal{Q} \equiv \text{discrete transition} \\
 \rho : \mathcal{Q} \times \mathbb{R}^n \rightarrow \mathbb{R}^n \equiv \text{reset map} 
\end{cases}$

$H$ is an **initialized rectangular automaton** if:

1. The set $\mathcal{Q}$ is finite
2. $f(q,x) = k(q) \in \mathcal{Q} \forall x \in \mathbb{R}^n$ (constant rational vector fields in each discrete mode)
3. The discrete transitions are of the form

\[ \varphi(q, x) = \begin{cases} 
q_j & q = q_i, x \in R_{ji} \\
: & \text{where all the } R_{ji} \text{ are rectangles} 
\end{cases} \]

4. There is a function $\nu : \mathcal{Q} \rightarrow \mathbb{Q}^n$ such that

\[ \varphi(q, x) \neq q \quad \Rightarrow \quad \rho(q, x) = \nu(q) \forall x \in \mathbb{R}^n \]

conditions for jumps are expressed by rectangles in $x$

the resets are independent of $x$ (and rectangles for nondeterministic case)
Example #5: Tank system

By adding “no-effect” resets one obtains an initialized rectangular automaton

Decidability

$H$ is an initialized rectangular automaton if:

1. the set $Q$ is finite
2. vector field is constant in each discrete mode
3. jump conditions rectangular in $x$
4. resets independent of $x$

Theorem: The reachability algorithm finishes in finite time for any initialized rectangular automaton (deterministic or not).

Moreover, one can implement the reachability algorithm exactly using finite memory and finite computation

- finite number of discrete states & constant resets ⇒ finite termination (only needs to compute a finite number of reach sets inside each mode)
- rational numbers needed for exact representation with finite memory
- constant vector fields & rectangular jump conditions make possible exact computation of reach sets inside each mode
Decidability

$H$ is an initialized rectangular automaton if:
1. the set $Q$ is finite
2. vector field is constant in each discrete mode
3. jump conditions rectangular in $x$
4. resets independent of $x$

Perhaps the most restrictive condition is the "initialization" because it clears any memory regarding the previous continuous evolution (other than what was encoded in the discrete state) but without it we may not have finite termination.

\[
\begin{align*}
\dot{s} &= 0 \\
\dot{x} &= s \\
x &:= \frac{x^-}{2} \\
s &:= 0
\end{align*}
\]

\[S_0 := \{1,0\} = \text{Reach}_0\]

Back to safety...

Given: hybrid automaton $H$:
\[
\dot{x} = f(q,x) \quad (q,x) = \Phi(q^-, x^-) \quad q(t), x(t) \in Q, x(t) \in \mathbb{R}^n, t \geq t_0
\]
set of initial states $H_0 \subseteq Q \times \mathbb{R}^n$

$H$ satisfies a safety property $p(q,x) = \Box (q(t), x(t)) \in \mathcal{F}, \mathcal{F} \subseteq Q \times \mathbb{R}^n$ if and only if $\text{Reach}_i(H_0) \subseteq \mathcal{F}$

\[\text{Reach}(H_0) \subseteq \mathcal{F}\]

Reachability algorithm:

\begin{align*}
\text{initialization:} & \quad \text{Reach}_{-1} = \emptyset \\
& \quad \text{Reach}_0 = S_0 \\
& \quad i = 0 \\
\text{loop:} & \quad \text{while} \ \text{Reach}_i \neq \text{Reach}_{i+1} \ \text{or} \ \text{Reach}_i \notin \mathcal{F} \ \text{do} \\
& \quad \text{Reach}_{i+1} = \text{Reach}_i \cup \{s' \in S : \exists s \in \text{Reach}_i, e \in E, (s,e,s') \in T\} \\
& \quad i = i + 1 \\
\text{end:} & \quad \text{if} \ \text{Reach}_i = \text{Reach}_{i+1}, \text{then} \ H \text{ satisfies } p \\
& \quad \text{else} \ H \text{ does not satisfy } p
\end{align*}
Backward reachability

Given: hybrid automaton $H$

\[ \dot{x} = f(q, x), \quad (q, x) = \Phi(q^-, x^-) \]
\[ q(t) \in Q, \quad x(t) \in \mathbb{R}^n \]
set of final states $\mathcal{H}_f \subseteq Q \times \mathbb{R}^n$

$\text{BackReach}_H(\mathcal{H}_f) \equiv \text{set of pairs } (q_0, x_0) \in Q \times \mathbb{R}^n \text{ for which there is a solution } (q, x) \text{ to } H \text{ for which:}$

1. \[(q(t_0), x(t_0)) = (q_0, x_0) \text{ starts at } (q_0, x_0)\]
2. \[\exists t \geq t_0 : (q(t), x(t)) \in \mathcal{H}_f \text{ passes through } \mathcal{H}_f\]

What can you say about

$\text{Reach}_H(\text{BackReach}_H(\mathcal{H}_f))$ \quad $\text{BackReach}_H(\text{Reach}_H(\mathcal{H}_0))$?
Backward reachability

Given: hybrid automaton $H$:

$$\dot{x} = f(q, x) \quad (q, x) = \Phi(q^-, x^-) \quad q(t) \in Q, \quad x(t) \in \mathbb{R}^n, \quad t \geq t_0$$

set of final states $\mathcal{H}_f \subset Q \times \mathbb{R}^n$

$$\text{BackReach}_H(\mathcal{H}_f) \equiv \text{set of pairs } (q_0, x_0) \in Q \times \mathbb{R}^n \text{ for which there is a solution } (q(t), x(t)) \text{ to } H \text{ for which:}$$

1. $(q(t_0), x(t_0)) = (q_0, x_0)$ \hspace{1cm} starts at $(q_0, x_0)$
2. $\exists t \geq t_0 : (q(t), x(t)) \in \mathcal{H}_f$ \hspace{1cm} passes through $\mathcal{H}_f$

$H$ satisfies a safety property

$$p(q, x) = \square \left( q(t), x(t) \right) \in \mathcal{F}$$

where $\mathcal{F} \subset Q \times \mathbb{R}^n$ is a nonempty set if and only if

$$\text{BackReach}_H(\neg \mathcal{F}) \cap \mathcal{H}_f = \emptyset$$

every point in every trajectory starting in $\mathcal{H}_f$ satisfies $p$

$\neg \mathcal{F}$ means $Q \times \mathbb{R}^n \setminus \mathcal{F}$

Transition system

transition system $T$ \hspace{1cm} $S$ \equiv set of states (finite or infinite) \hspace{1cm} $\mathcal{E}$ \equiv alphabet of events (finite or infinite) \hspace{1cm} $T \subset S \times \mathcal{E} \times S$ \equiv transition relation

$$S = \{1, 2, 3\} \quad \mathcal{E} = \{a, b\} \quad T \in \{(1, a, 2), (2, b, 1), (2, b, 3), (3, a, 1)\}$$

execution of a transition system \equiv sequence of states $\{s_0, s_1, s_2, \ldots\}$ such that there exists a sequence of events $\{e_0, e_1, e_2, \ldots\}$ for which $(s_i, e_i, s_{i+1}) \in T \quad \forall \quad i$

Given a set of initial states $S_0 \subset S$:

$$\text{Reach}_T(S_0) \equiv \text{set of states } s \in S \text{ for which there is a finite execution that starts in } S_0 \text{ and ends at } s$$

Given a set of final states $S_f \subset S$:

$$\text{BackReach}_T(S_f) \equiv \text{set of states } s \in S \text{ for which there is a finite execution that starts at } s \text{ and ends in } S_f$$
Backward reachability algorithm

**Backward reachability algorithm:**

initialization: \( B\text{Reach}_{-1} = \emptyset \)
\( B\text{Reach}_0 = \mathcal{S}_f \)
\( i = 0 \)

loop: while \( B\text{Reach}_i \neq B\text{Reach}_{i-1} \) do
\( B\text{Reach}_{i+1} = B\text{Reach}_i \cup \{ s \in \mathcal{S} : \exists s' \in B\text{Reach}_i, e \in E \text{ s.t. } (s,e,s') \in T \} \)
\( i = i + 1 \)

states from where one can transition into \( B\text{Reach}_i \)

\( \mathcal{S}_f = \{5\} \)
\( B\text{Reach}_0 = \{5\} \)
\( B\text{Reach}_1 = \{2,3,5\} \)
\( B\text{Reach}_2 = \{1,2,3,5\} \)
\( B\text{Reach}_3 = \{1,2,3,5\} \)
\( \text{BackReach}_i(\{5\}) = \{1,2,3,5\} \)

Theorem: If \( \mathcal{S} \) is finite then

(i) the backwards reachability algorithm finishes in a finite number of steps and
(ii) upon exiting the while-loop \( B\text{Reach}_i = \text{BackReach}_i(\mathcal{S}_f) \)

Why?

(i) In each iteration the number of elements in \( B\text{Reach}_i \) increases by at least 1.
Since it can have, at most, as many elements as \( \mathcal{S} \) there can only be as many iterations as the number of elements in \( \mathcal{S}_f \) minus the number of elements in \( \mathcal{S}_0 \).
(ii) \( B\text{Reach}_i \equiv \) the set of states that can reach \( \mathcal{S}_f \) in \( i \) steps, thus any state from which \( \mathcal{S}_f \) can be reached in a finite number of steps must be in one of the \( \text{Reach}_i \)}
Invariant set algorithm

Invariant set algorithm (backward reachability working with complements):

initialization: \[ \text{Inv}_i := \text{Inv}_0 = S \setminus \mathcal{F} \]
\( i = 0 \)

loop: while \( \text{Inv}_i \neq \text{Inv}_{i-1} \) do
\[ \text{Inv}_{i+1} = \text{Inv}_i \cap \{ s \in S : \forall s' \notin \text{Inv}_{i-1}, e \in \mathcal{E} \text{ s.t. } (s,e,s') \notin \mathcal{T} \} \]
\( i = i + 1 \)

Theorem: If \( S \) is finite then
(i) the algorithm finishes in a finite number of steps and
(ii) upon exiting the while loop
\[ \text{Inv}_i = \text{Inv}_i(\neg \mathcal{F}) = \text{largest invariant set contained in } \neg \mathcal{F} \]

Why?
(i) In each iteration the number of elements in \( \text{Inv}_i \) decreases by at least 1.
There can only be as many iterations as the number of states in \( S \setminus \mathcal{F} \).
(ii) Upon exiting:
\[ \text{Inv} \subseteq \{ s \in S : \forall s' \notin \text{Inv}, e \in \mathcal{E} \text{ s.t. } (s,e,s') \notin \mathcal{T} \} \]
set of states for which there is no transition out of \( \text{Inv} \)
\[ \downarrow \]
\( \text{Inv} \) is invariant set
Invariant sets

Given: hybrid automaton $H$:
\[ \dot{x} = f(q, x) \quad (q, x) = \Phi(q^-, x^-) \quad q(t) \in Q, \ x(t) \in \mathbb{R}^n, \ t \geq t_0 \]
set of final states $\mathcal{H}_f \subseteq Q \times \mathbb{R}^n$

$\text{Inv}_H(\mathcal{H}_f) \equiv \text{largest invariant set contained in } \mathcal{H}_f$
As just seen, $\text{Inv}_H(\mathcal{H}_f) = \neg \text{BackReach}(\neg \mathcal{H}_f)$

$H$ satisfies a safety property
\[ p(q, x) = \Box (q(t), x(t)) \in \mathcal{F} \]
where $\mathcal{F} \subseteq Q \times \mathbb{R}^n$ is a nonempty set if and only if
$\text{BackReach}_H(\neg \mathcal{F}) \cap \mathcal{H}_0 = \emptyset$
or equivalently
\[ \neg \text{Inv}_H(\mathcal{F}) \cap \mathcal{H}_0 = \emptyset \Leftrightarrow \mathcal{H}_0 \subset \text{Inv}_H(\mathcal{F}) \]

Back to safety (again)...

Given: hybrid automaton $H$:
\[ \dot{x} = f(q, x) \quad (q, x) = \Phi(q^-, x^-) \quad q(t) \in Q, \ x(t) \in \mathbb{R}^n, \ t \geq t_0 \]
set of final states $\mathcal{H}_f \subseteq Q \times \mathbb{R}^n$

$H$ satisfies a safety property $p(q, x) = \Box (q(t), x(t)) \in \mathcal{F} \subseteq Q \times \mathbb{R}^n$ (nonempty set)
if and only if $\text{BackReach}_H(\neg \mathcal{F}) \cap \mathcal{H}_0 = \emptyset$
every point in every trajectory starting in $\mathcal{H}_0$ satisfies $p$

Backwards Reachability algorithm:

initialization: $\text{BReach}_{-1} = \emptyset$, $\text{BReach}_0 = S_f := \neg \mathcal{F}$
$I = 0$

loop: while $\text{BReach}_I \neq \text{BReach}_{I-1}$ or $\text{BReach}_I \cap \mathcal{H}_0 \neq \emptyset$ do
$\text{BReach}_{I+1} = \text{BReach}_I \cup \{s \in S : \exists s' \in \text{BReach}_I, e \in E, (s, e, s') \in T\}$
$I = I + 1$

end: if $\text{Reach}_I = \text{Reach}_{I-1}$ then $H$ satisfies $p$ else $H$ does not satisfy $p$
Controller design based on backward reachability

**Backwards Reachability algorithm:**

initialization: \( \text{BReach}_{-1} = \emptyset \)  
\( \text{BReach}_0 = S_f := \neg F \)  
\( i = 0 \)  
algorithm can terminate 
\( \text{BReach}_i \) intersects \( \mathcal{H}_0 \)  

loop:  
while \( \text{BReach}_i \neq \text{BReach}_{i-1} \) or \( \text{BReach}_i \cap \mathcal{H}_0 \neq \emptyset \) do 
\( \text{BReach}_{i+1} = \text{BReach}_i \cup \{ s \in S : \exists s' \in \text{BReach}_i, e \in E, (s,e,s') \in T \} \) 
\( i = i + 1 \)  
end:  
if \( \text{Reach}_i = \text{Reach}_{i-1} \) then \( H \) satisfies \( p \) else \( H \) does not satisfy \( p \)

When one obtains \( \text{BReach}_{i+1} \cap \mathcal{H}_0 \neq \emptyset \) it is because 
\( \{ s \in S : \exists s' \in \text{BReach}_i, e \in E, (s,e,s') \in T \} \cap \mathcal{H}_0 \neq \emptyset \)  
therefore  
transition from \( \mathcal{H}_0 \) to \( \text{BReach}_i \)  
\( \exists s \in \mathcal{H}_0, s' \in \text{BReach}_i, e \in E : (s,e,s') \in T \)  

**Safety could be recovered if the transition** \((s,e,s') \in T\) **was removed**

**Control design based on backward reachability:**

inhibit any transition \((s,e,s')\) for which \( s' \in \text{BReach}_i, e \in E, s \in \mathcal{H}_0 \)

Typically amounts to

1. removing a discrete transition  
2. adding a discrete transition to prevent continuous evolution

Next lecture...

Lyapunov stability of ODEs  
- epsilon-delta and beta-function definitions  
- Lyapunov’s stability theorem  
- LaSalle’s invariance principle  
Lyapunov stability of hybrid systems