Hybrid Control and Switched Systems

Lecture #6
Reachability

João P. Hespanha

University of California
at Santa Barbara

Summary

Review of previous lecture
Reachability

- transition systems
- reachability algorithm
- backward reachability algorithm
- invariance algorithm
- controller design based on backward reachability

Sequence Properties (signals)

$\mathcal{X}_{\text{sig}} \equiv$ set of all piecewise continuous signals $x\colon[0, T) \to \mathbb{R}^n,\ T \in (0, \infty]$
$\mathcal{Q}_{\text{sig}} \equiv$ set of all piecewise constant signals $q\colon[0, T) \to \mathcal{Q},\ T \in (0, \infty]$

Sequence property $\equiv p : \mathcal{Q}_{\text{sig}} \times \mathcal{X}_{\text{sig}} \to \{\text{false},\text{true}\}$

E.g.,

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

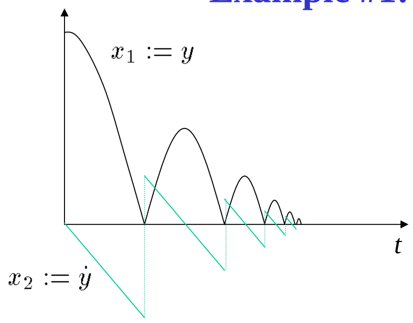
A pair of <u>signals</u> $(q, x) \in \mathcal{Q}_{\text{sig}} \times \mathcal{X}_{\text{sig}}$ ***satisfies*** p if $p(q, x) = \text{true}$

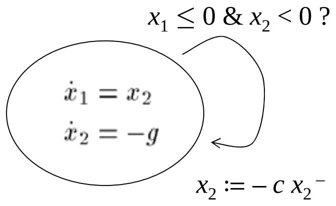
A hybrid <u>automaton</u> H ***satisfies*** p (write $H \vDash p$) if
$p(q, x) = \text{true}$, for every solution (q, x) of H

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

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

Example #1: Bouncing ball





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

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

$(\Box\, p)(t_0) \Leftrightarrow \forall\, t \geq t_0,\ p(t)$
$(\Diamond\, p)(t_0) \Leftrightarrow \exists\, t \geq t_0,\ p(t)$
$(\Box \Diamond\, p)(t_0) \Leftrightarrow \forall\, t_1 \geq t_0,\ \exists\, t \geq t_1\ p(t)$
$(\Diamond \Box\, p)(t_0) \Leftrightarrow \exists\, t_1 \geq t_0,\ \forall\, t \geq t_1\ p(t)$

Safety properties

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

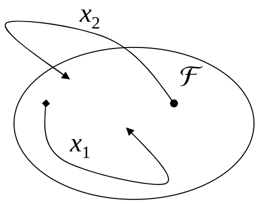
safety property ≡ a sequence property p that is:

1. *nonempty*, i.e., $\exists\ (q,x)$ such that $p(q,x)$ = true
2. *prefix closed*, i.e., given signals (q,x)
   $p(q, x) \Rightarrow p(q^*, x^*)$
   for every prefix (q^*, x^*) to (q, x)
3. *limit closed*, i.e., given an infinite sequence of signals
   (q_1,x_1) , (q_2,x_2), (q_3,x_3), etc.
   each element satisfying p such that
   (q_k,x_k) is a prefix to (q_{k+1},x_{k+1}) $\qquad \forall\ k$
   then $(q,x) \coloneqq \lim_{k\to\infty} (q_k,x_k)$ 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., $p(q, x) = \Box\ \big(q(t),x(t)\big) \in \mathcal{F}$ where $\mathcal{F} \subset Q\times \mathbb{R}^n$ is a nonempty set



x_1 satisfies p
x_2 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:
$p(q, x) = \inf_t x(t) > 0$ (not of the form above; not limit closed, Why?)

Liveness properties

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

liveness property ≡ a sequence property p with the property that for every finite $(q^*, x^*) \in \mathcal{Q}_{\mathrm{sig}} \times \mathcal{X}_{\mathrm{sig}}$ there is some $(q, x) \in \mathcal{Q}_{\mathrm{sig}} \times \mathcal{X}_{\mathrm{sig}}$ 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 $\mathcal{F} \subset Q\times \mathbb{R}^n$ is a nonempty set
$p(q, x) = \Box\, \Diamond\, (q(t),x(t))\in \mathcal{F}$	(always, eventually: $\forall\, t_1\ge t_0$, $\exists\, t\ge t_1$)
$p(q, x) = \Diamond\, \Box\, (q(t),x(t))\in \mathcal{F}$	(eventually, always: $\exists\, t_1\ge t_0$, $\forall\, t\ge t_1$)
$p(q, x) = \exists\, L>0\ \Box\ \|x\| < L$	*what does it mean?*
$p(q, x) = \forall\, \varepsilon>0\ \Diamond\, \Box\ \|x\| < \varepsilon$	*what does it mean?*

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 $(q, x) \in \mathcal{Q}_{\mathrm{sig}} \times \mathcal{X}_{\mathrm{sig}}$ 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***" ≡ property of the whole family of solutions
e.g., stability (continuity with respect to initial conditions) is not a sequence property because by looking a 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

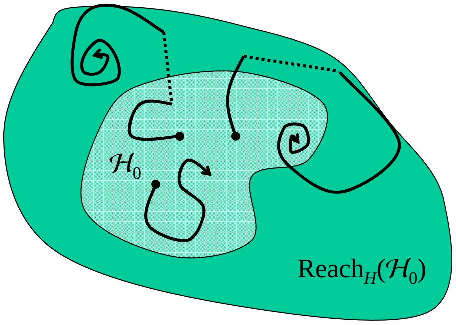
Given: hybrid automaton *H:*

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

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

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

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



Reachability

Given: hybrid automaton *H:*

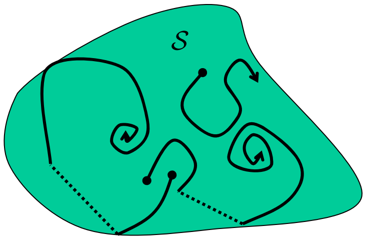
$$\dot{x} = f(q,x) \qquad (q,x) = \Phi(q^-, x^-) \qquad q(t) \in \mathcal{Q},\ x(t) \in \mathbb{R}^n,\ t \geq t_0$$

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

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

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

Invariant set ≡ set $\mathcal{S} \subset \mathcal{Q} \times \mathbb{R}^n$ for which Reach$_H(\mathcal{S}) = \mathcal{S}$



Reachability v.s. Safety

Given: hybrid automaton *H:*

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

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

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

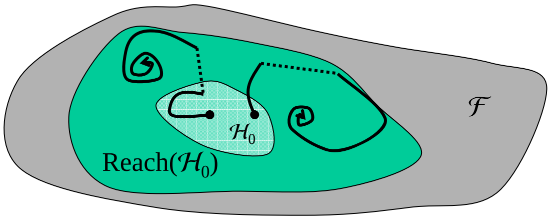
1. $(q(t_0),x(t_0)) \in \mathcal{H}_0$ starts in $\mathcal{H}_0$
2. $\exists\, t \geq t_0 : (q(t),x(t)) = (q_f, x_f)$ passes through (q_f, x_f)

H satisfies a safety property

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

where $\mathcal{F} \subset \mathcal{Q} \times \mathbb{R}^n$ 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:*

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

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

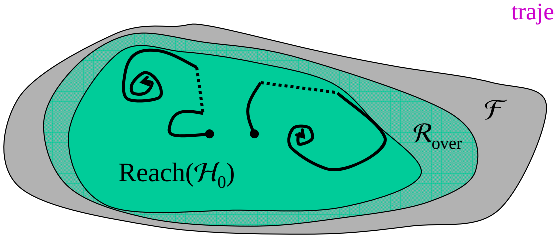
$\text{Reach}_H(\mathcal{H}_0) \equiv$ set of pairs $(q_f, x_f) \in \mathcal{Q} \times \mathbb{R}^n$ for which there is a solution (q,x) to H for which:

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

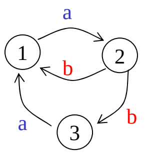


Are under-approximations useful to study reachability?

Transition system

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

$$\begin{array}{c} \text{transition} \\ \text{system} \\ T \end{array} \begin{cases} \mathcal{S} & \equiv \text{set of states (finite or infinite)} \\ \mathcal{E} & \equiv \text{alphabet of events (finite or infinite)} \\ \mathrm{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S} & \equiv \text{transition relation} \end{cases}$$



$\mathcal{S} = \{1,2,3\}$

$\mathcal{E} = \{a,b\}$

$\mathrm{T} \in \{ (1,a,2), (2,b,1), (2,b,3), (3,a,1) \}$

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

Given a set of initial states $\mathcal{S}_0 \subset \mathcal{S}$:

$\mathrm{Reach}_T(\mathcal{S}_0) \equiv$ set of states $s \in \mathcal{S}$ 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

$$\begin{array}{c} \text{automata} \\ M \end{array} \begin{cases} \mathcal{Q} := \{q_1, q_2, \dots, q_n\} & \equiv \text{finite set of states} \\ \Sigma := \{\text{a, b, c,}\dots\} & \equiv \text{finite set of input symbols (alphabet)} \\ \Phi : \mathcal{Q} \times \Sigma \to \mathcal{Q} & \equiv \text{transition function} \end{cases}$$

$$\begin{array}{c} \text{transition} \\ \text{system} \\ T \end{array} \begin{cases} \mathcal{S} = \mathcal{Q} & \equiv \text{set of states (finite)} \\ \mathcal{E} = \Sigma & \equiv \text{alphabet of events (finite)} \\ \mathrm{T} = \{ (\mathrm{s,e,}\Phi(\mathrm{s,e})) : \mathrm{s} \in \mathcal{Q}, \mathrm{e} \in \Sigma \} & \equiv \text{transition relation} \end{cases}$$

for nondeterministic finite automaton

$$\mathrm{T} = \{ (\mathrm{s,e,s'}) : \mathrm{s} \in \mathcal{Q}, \mathrm{e} \in \Sigma, \mathrm{s'} \in \Phi(\mathrm{s,e}) \}$$

Same set of reachable states

Transition systems

As far as reachability goes …

Same set of reachable states

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

hybrid automata H
$\mathcal{Q}$ ≡ set of discrete states
$\mathbb{R}^n$ ≡ continuous state-space
$f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field
$\Phi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q} \times \mathbb{R}^n$ ≡ discrete transition (& reset map)

transition system T
$\mathcal{S} = \mathcal{Q} \times \mathbb{R}^n$ ≡ set of states (infinite)
$\mathcal{E} = \{\tau, (q_i,q_j): q_i,q_j \in \mathcal{Q}\}$ ≡ alphabet of events:
τ called the continuous evolution event
(q_i,q_j) called a jump event
$\mathrm{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S}$ ≡ transition relation

$\big((q_0,x_0), (q_0,q_f)\,, (q_f,x_f) \big) \in \mathrm{T}$ if $(x_f\,, q_f) = \Phi(x_0, q_0)$

$\big((q_0,x_0), \tau, (q_0,x_f) \big) \in \mathrm{T}$ if $\exists\, t_f > 0$ s.t. $\dot{x} = f(q_0, x),\ x(0) = x_0,\ x(t_f) = x_f$
$(x(t), q_0) = \Phi(x^-(t), q_0), \quad \forall t \in (0, t_f)$

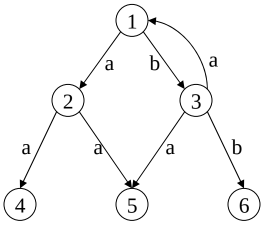
same (q_0,x_0) and τ lead
to many distinct elements in T
(flows modeled as nondetermism)

Reachability algorithm

Reachability algorithms:

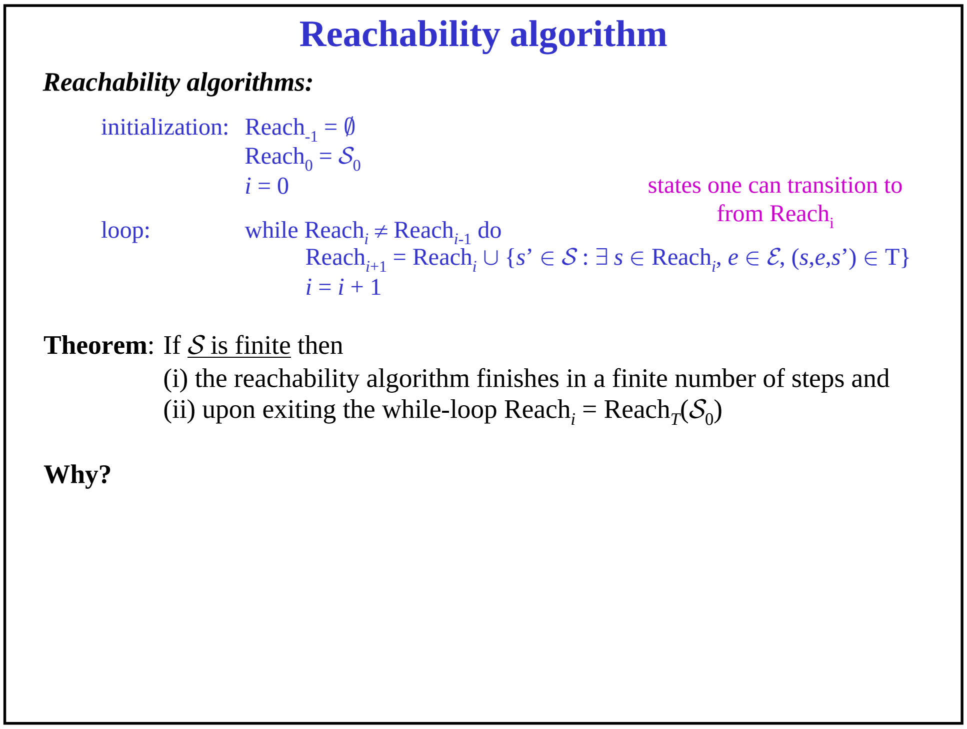
```
initialization:  Reach_{-1} = ∅
                 Reach_0 = $\mathcal{S}_0$
                 $i = 0$
loop:            while Reach_i ≠ Reach_{i-1} do
                     Reach_{i+1} = Reach_i ∪ {s' ∈ $\mathcal{S}$ : ∃ s ∈ Reach_i, e ∈ $\mathcal{E}$, (s,e,s') ∈ T}
                     $i = i + 1$
```

states one can transition to from $\mathrm{Reach}_\mathrm{i}$



$\mathcal{S}_0 = \{3\}$
$\mathrm{Reach}_0 = \{3\}$
$\mathrm{Reach}_1 = \{1,3,5,6\}$
$\mathrm{Reach}_2 = \{1,2,3,5,6\}$
$\mathrm{Reach}_3 = \mathcal{Q}$
$\mathrm{Reach}_4 = \mathcal{Q}$
$\mathrm{Reach}_T(\{3\}) = \mathcal{Q}$

$\mathcal{S}_0 = \{2\}$
$\mathrm{Reach}_0 = \{2\}$
$\mathrm{Reach}_1 = \{2,4,5\}$
$\mathrm{Reach}_2 = \{2,4,5\}$
$\mathrm{Reach}_T(\{2\})=\{2,4,5\}$



Reachability algorithm

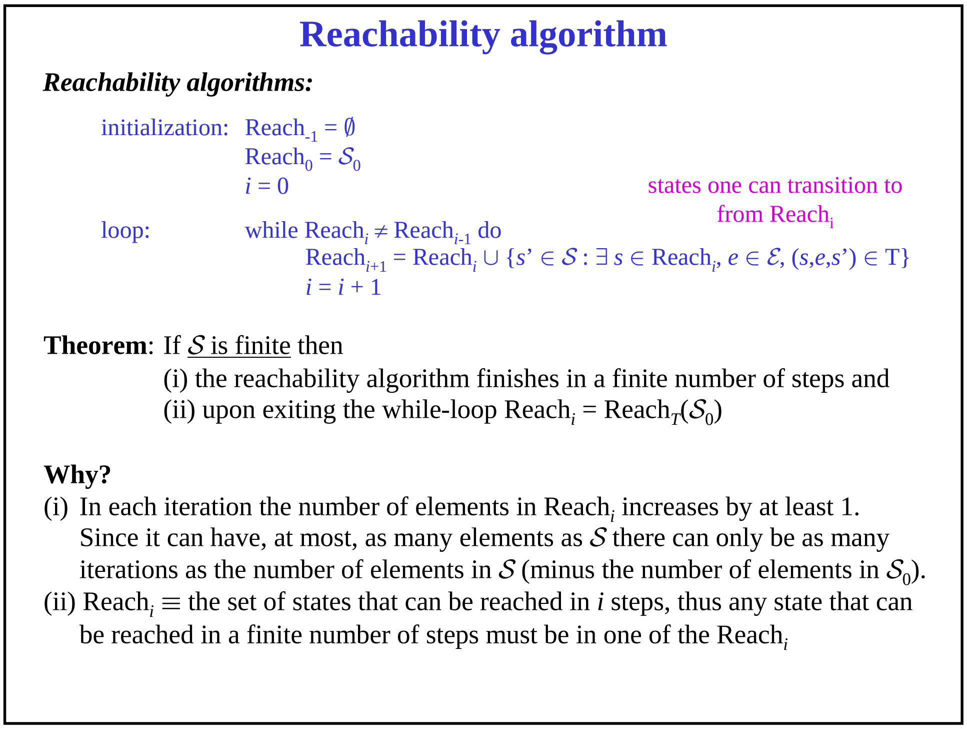
Reachability algorithms:

```
initialization:  $\text{Reach}_{-1} = \emptyset$
                 $\text{Reach}_0 = \mathcal{S}_0$
                 $i = 0$
loop:            while $\text{Reach}_i \neq \text{Reach}_{i-1}$ do
                     $\text{Reach}_{i+1} = \text{Reach}_i \cup \{s' \in \mathcal{S} : \exists\, s \in \text{Reach}_i, e \in \mathcal{E}, (s,e,s') \in \text{T}\}$
                     $i = i + 1$
```

states one can transition to from Reach_i

Theorem: If $\mathcal{S}$ is finite then
(i) the reachability algorithm finishes in a finite number of steps and
(ii) upon exiting the while-loop $\text{Reach}_i = \text{Reach}_T(\mathcal{S}_0)$

Why?



Reachability algorithm

Reachability algorithms:

```
initialization:  $\text{Reach}_{-1} = \emptyset$
                 $\text{Reach}_0 = \mathcal{S}_0$
                 $i = 0$
loop:            while $\text{Reach}_i \neq \text{Reach}_{i-1}$ do
                     $\text{Reach}_{i+1} = \text{Reach}_i \cup \{s' \in \mathcal{S} : \exists\, s \in \text{Reach}_i, e \in \mathcal{E}, (s,e,s') \in \text{T}\}$
                     $i = i + 1$
```

states one can transition to from Reach_i

Theorem: If $\mathcal{S}$ is finite then
(i) the reachability algorithm finishes in a finite number of steps and
(ii) upon exiting the while-loop $\text{Reach}_i = \text{Reach}_T(\mathcal{S}_0)$

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 $\mathcal{S}$ there can only be as many iterations as the number of elements in $\mathcal{S}$ (minus the number of elements in $\mathcal{S}_0$).

(ii) $\text{Reach}_i \equiv$ 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

Reachability algorithm:

```
initialization:  $\text{Reach}_{-1} = \emptyset$
                 $\text{Reach}_0 = \mathcal{S}_0$
                 $i = 0$
loop:            while $\text{Reach}_i \neq \text{Reach}_{i-1}$ do
                     $\text{Reach}_{i+1} = \text{Reach}_i \cup \{s' \in \mathcal{S} : \exists\, s \in \text{Reach}_i,\ e \in \mathcal{E},\ (s,e,s') \in T\}$
                     $i = i + 1$
```

states one can transition to from Reach_i

Two difficulties with hybrid automata

1. the set of states $\mathcal{S} := \mathcal{Q} \times \mathbb{R}^n$ is not finite (algorithm may not terminate)
2. In the while loop: $\text{Reach}_{i+1} = \text{Reach}_i \cup \mathcal{S}_1 \cup \mathcal{S}_2$

   Computation of

   $\mathcal{S}_1 := \{s' \in \mathcal{S} : \exists\, s \in \text{Reach}_i,\ e = (q_i,q_j) \in \mathcal{E},\ (s,e,s') \in T\}$ is simple

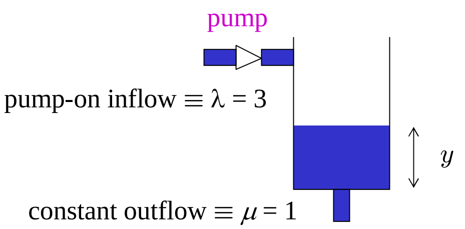
   but

   $\mathcal{S}_2 := \{s' \in \mathcal{S} : \exists\, s \in \text{Reach}_i,\ e = \tau,\ (s,e,s') \in T\}$ is not (in general)

$\mathcal{S}_1 = \{(q_f,x_f) \in \mathcal{S} : \exists\, (q_0,x_0) \in \text{Reach}_i,\ (q_f,x_f) = \Phi\,(q_0,x_0)\} = \Phi(\text{Reach}_i)$

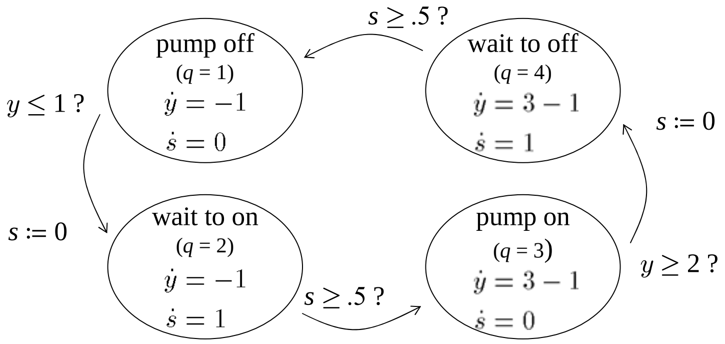
$\mathcal{S}_2 = \{(q_0,x_f) \in \mathcal{S} : \exists\, (q_0,x_0) \in \text{Reach}_i,$ "there is a continuous evolution from x_0 to x_f inside mode q_0" $\}$

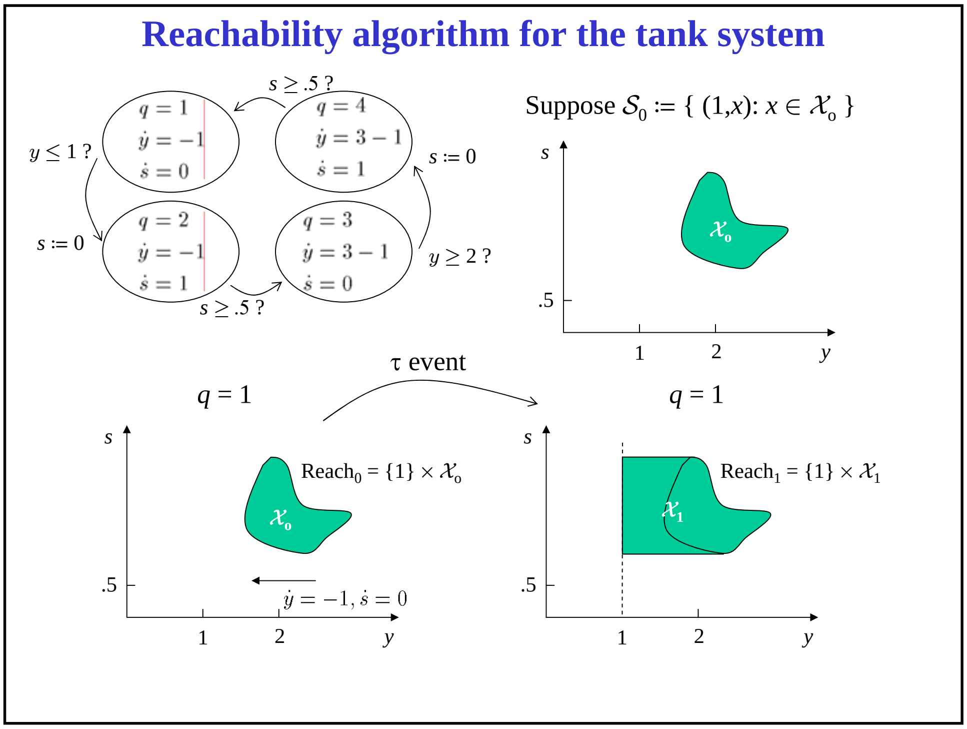
Example #5: Tank system



goal ≡ prevent the tank from emptying or filling up

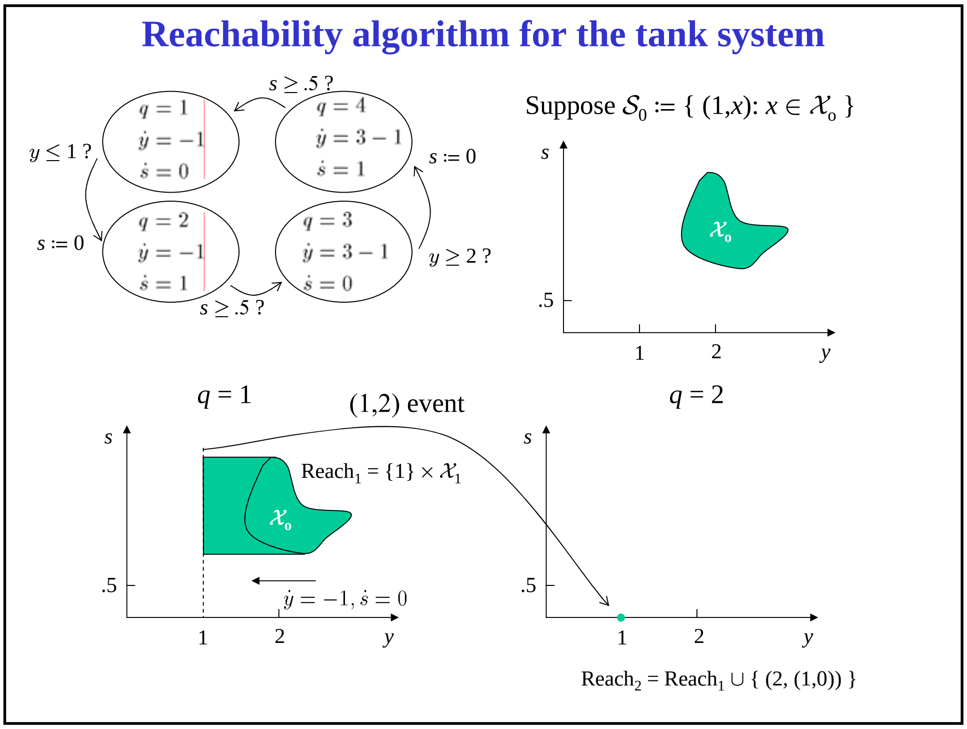
$\delta = .5$ ≡ delay between command is sent to pump and the time it is executed





Reachability algorithm for the tank system

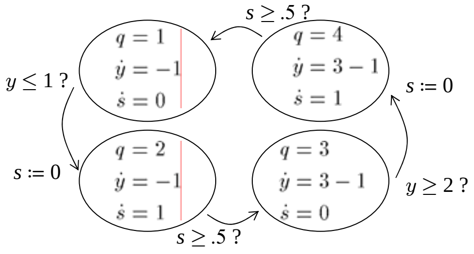
Suppose $\mathcal{S}_0 := \{ (1,x) : x \in \mathcal{X}_o \}$



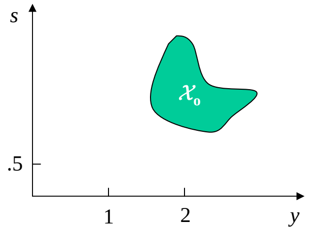
Reachability algorithm for the tank system

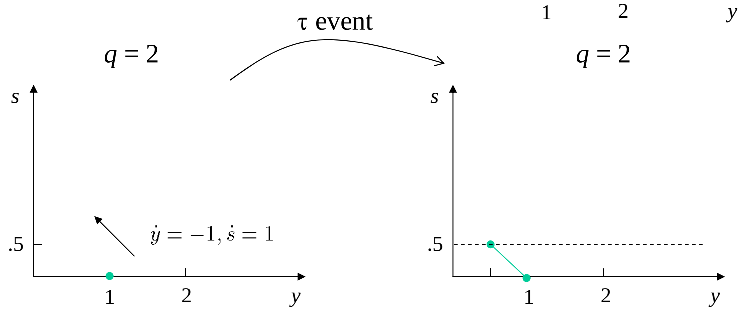
Suppose $\mathcal{S}_0 := \{ (1,x) : x \in \mathcal{X}_o \}$

Reachability algorithm for the tank system



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

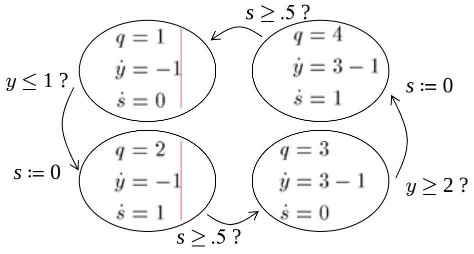




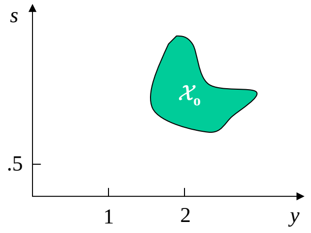
$\text{Reach}_2 = \text{Reach}_1 \cup \{(2, (1,0))\}$

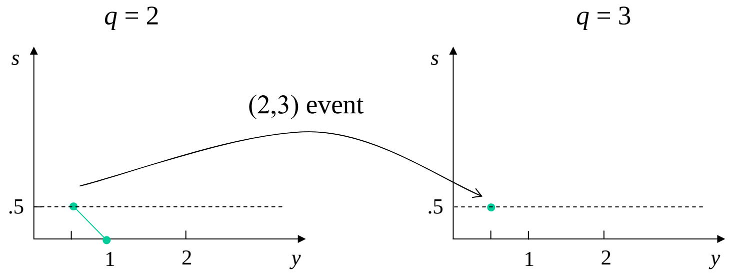
$\text{Reach}_3 = \text{Reach}_1 \cup \{(2, (1-\alpha,\alpha)) : \alpha \in [0,.5]\}$

Reachability algorithm for the tank system



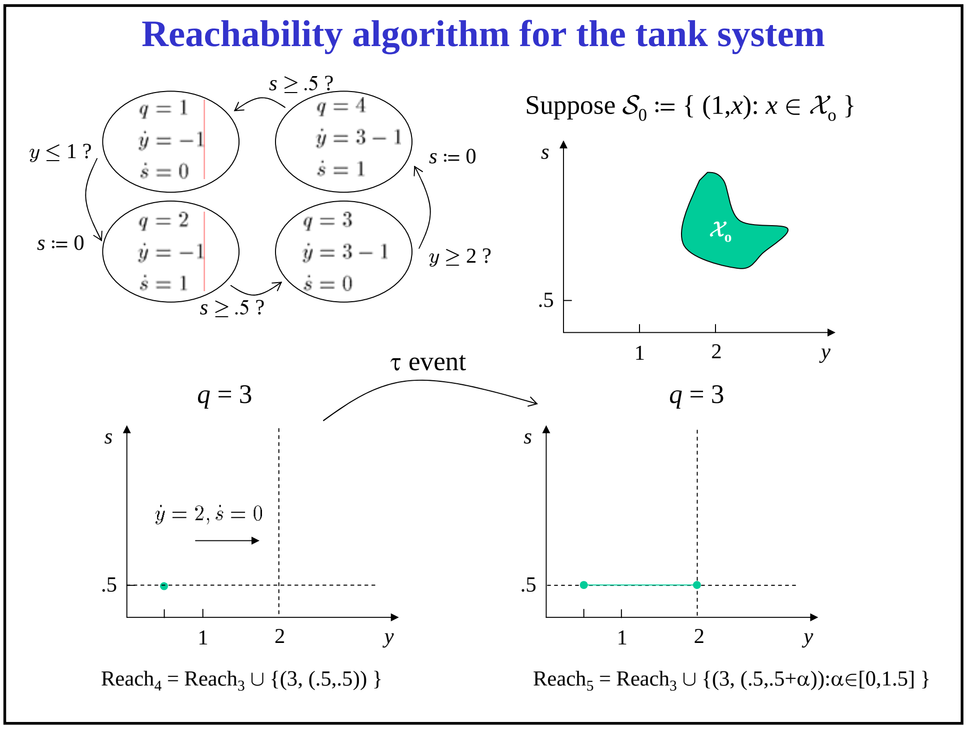
Suppose $\mathcal{S}_0 := \{ (1,x): x \in \mathcal{X}_o \}$





$\text{Reach}_3 = \text{Reach}_1 \cup \{(2, (1-\alpha,\alpha)) : \alpha \in [0,.5]\}$

$\text{Reach}_4 = \text{Reach}_3 \cup \{(3, (.5,.5))\}$

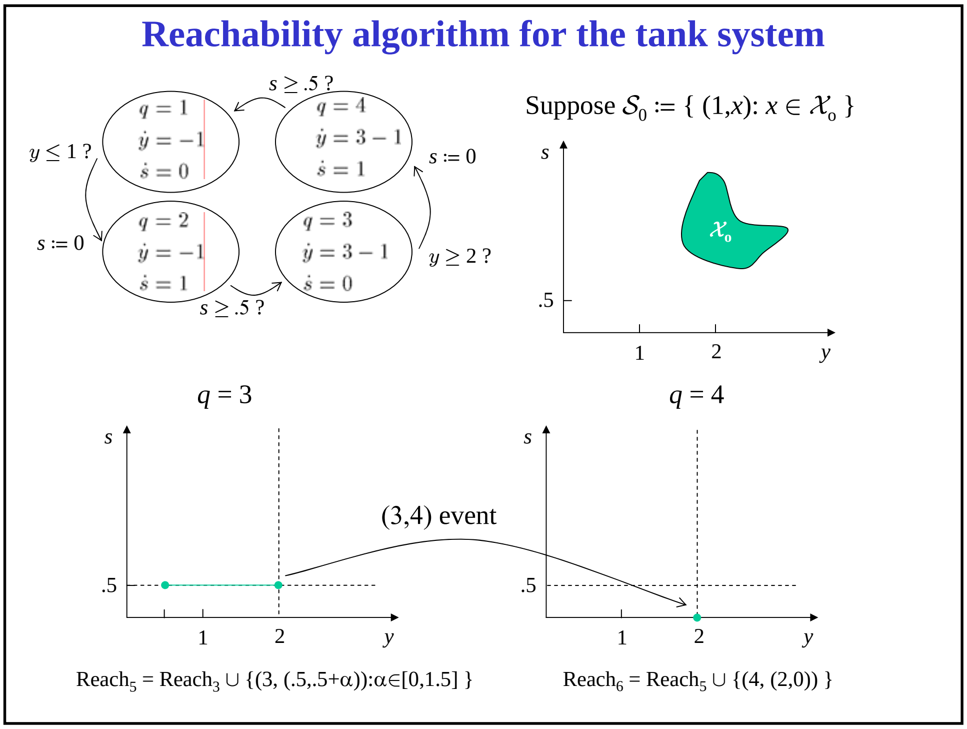


Reachability algorithm for the tank system

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

$\text{Reach}_4 = \text{Reach}_3 \cup \{(3, (.5,.5)) \}$

$\text{Reach}_5 = \text{Reach}_3 \cup \{(3, (.5,.5+\alpha)):\alpha \in [0,1.5] \}$



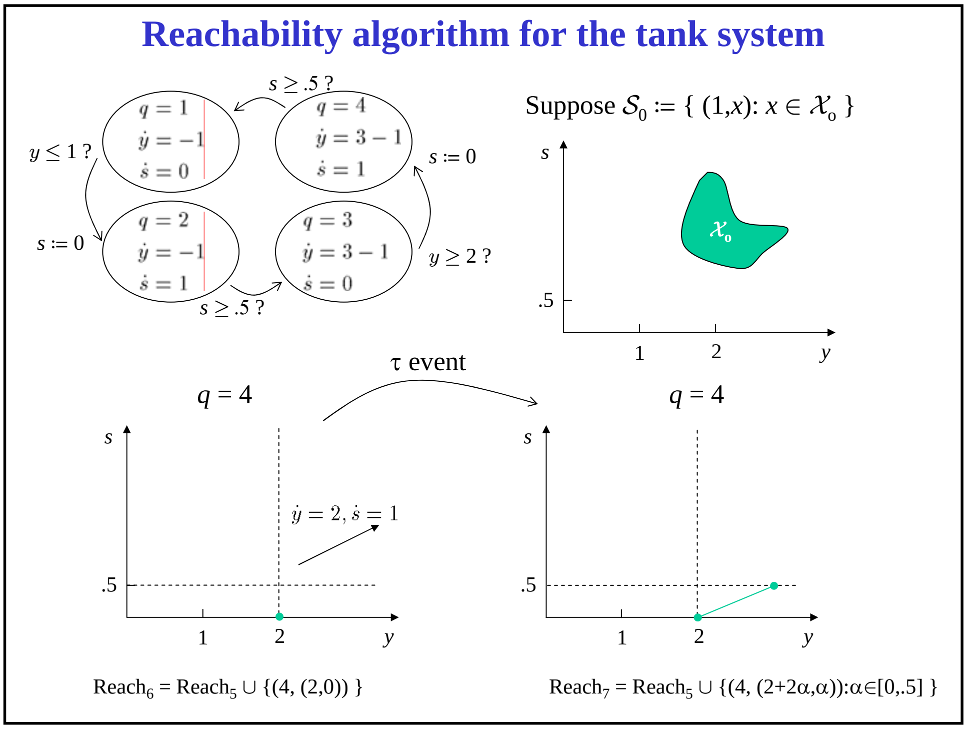
Reachability algorithm for the tank system

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

$\text{Reach}_5 = \text{Reach}_3 \cup \{(3, (.5,.5+\alpha)):\alpha \in [0,1.5] \}$

$\text{Reach}_6 = \text{Reach}_5 \cup \{(4, (2,0)) \}$

Reachability algorithm for the tank system

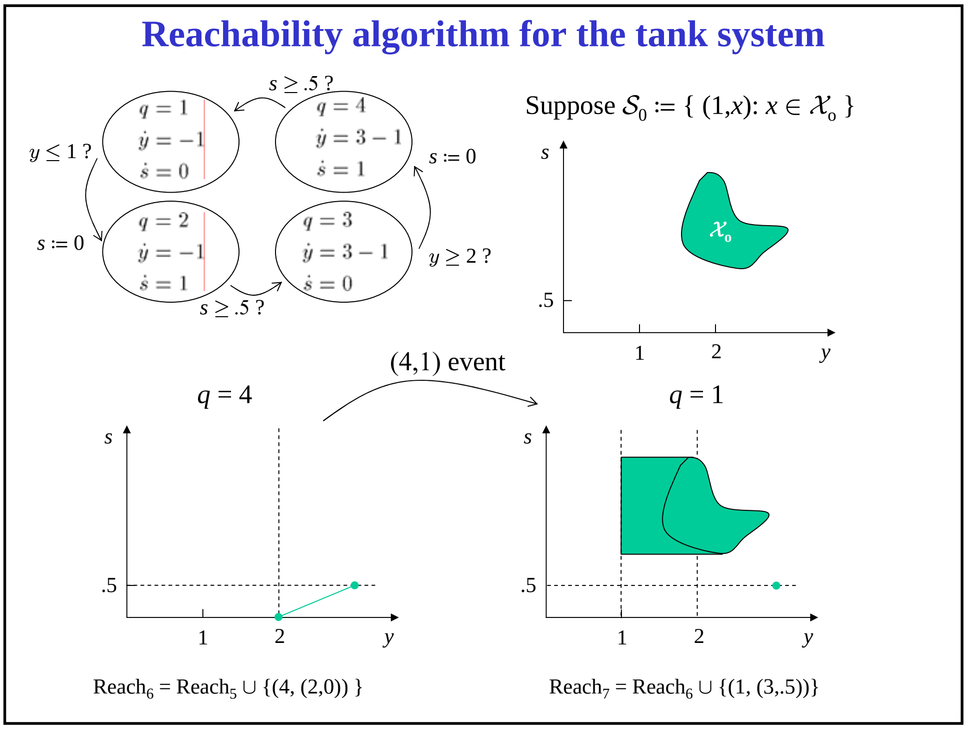


Suppose $\mathcal{S}_0 := \{ (1,x) \colon x \in \mathcal{X}_o \}$

$\text{Reach}_6 = \text{Reach}_5 \cup \{(4, (2,0))\}$

$\text{Reach}_7 = \text{Reach}_5 \cup \{(4, (2+2\alpha,\alpha)) \colon \alpha \in [0,.5]\}$

Reachability algorithm for the tank system

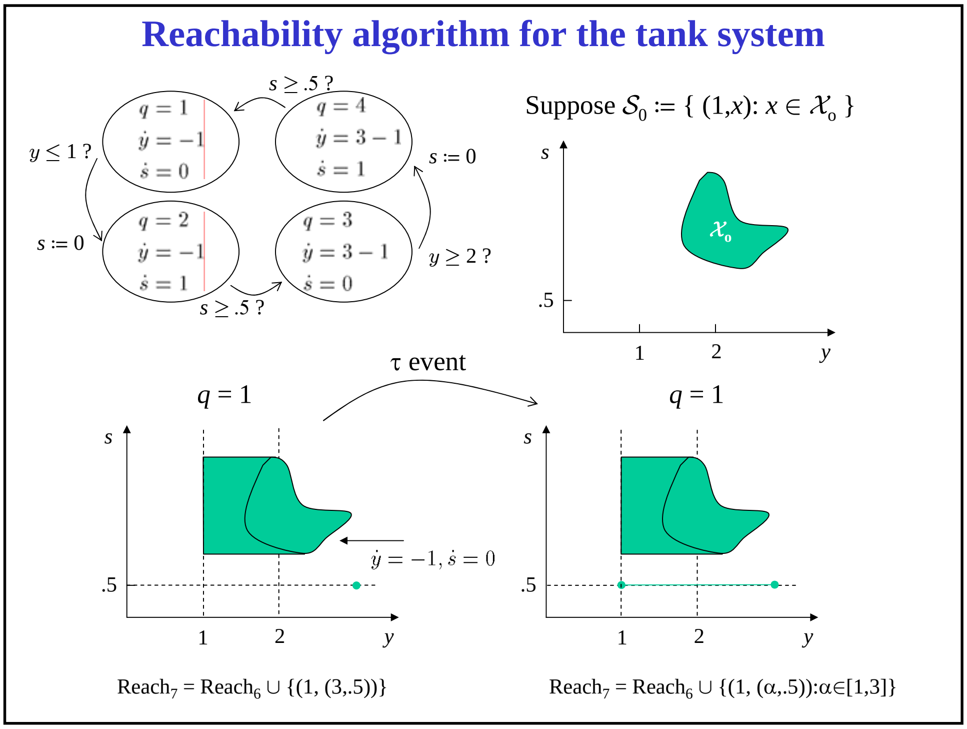


Suppose $\mathcal{S}_0 := \{ (1,x) \colon x \in \mathcal{X}_o \}$

$\text{Reach}_6 = \text{Reach}_5 \cup \{(4, (2,0))\}$

$\text{Reach}_7 = \text{Reach}_6 \cup \{(1, (3,.5))\}$

Reachability algorithm for the tank system

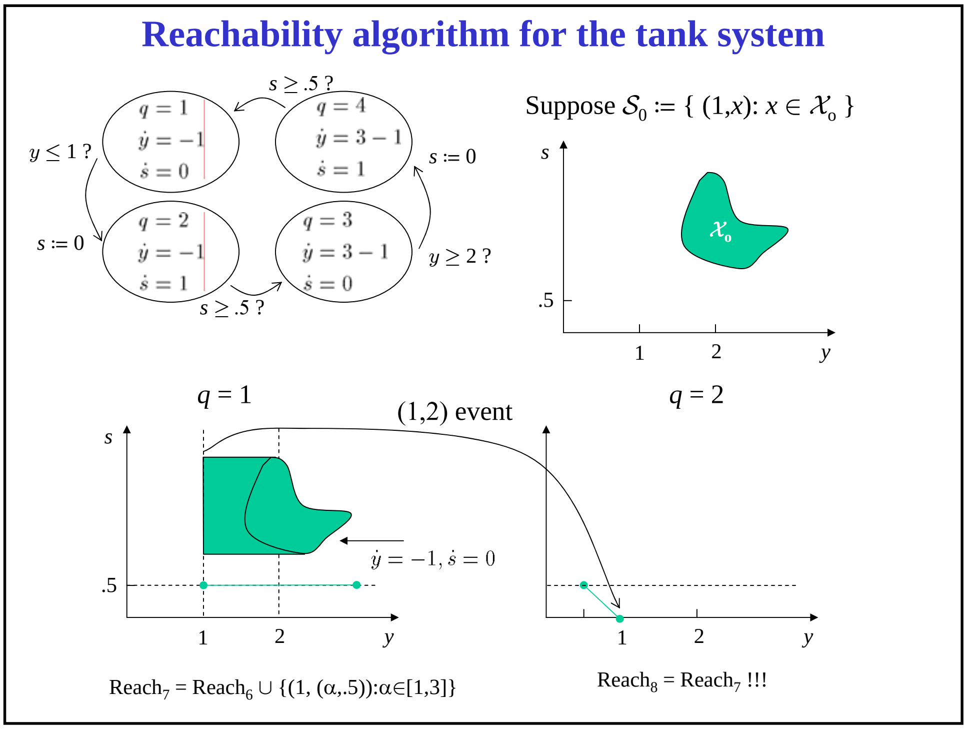


Suppose $\mathcal{S}_0 := \{ (1,x): x \in \mathcal{X}_o \}$

$\text{Reach}_7 = \text{Reach}_6 \cup \{(1, (3,.5))\}$

$\text{Reach}_7 = \text{Reach}_6 \cup \{(1, (\alpha,.5)):\alpha\in[1,3]\}$

Reachability algorithm for the tank system

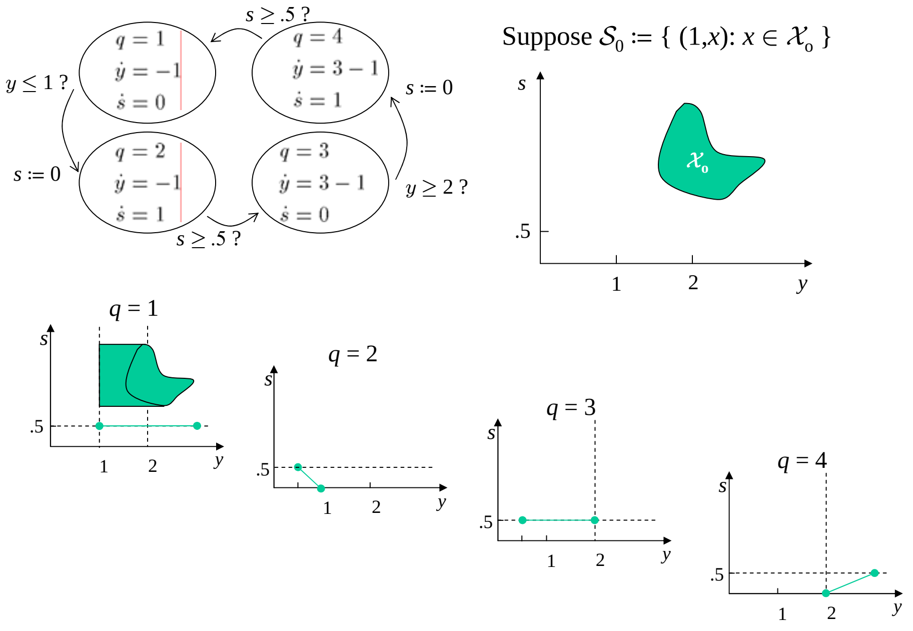


Suppose $\mathcal{S}_0 := \{ (1,x): x \in \mathcal{X}_o \}$

$\text{Reach}_7 = \text{Reach}_6 \cup \{(1, (\alpha,.5)):\alpha\in[1,3]\}$

$\text{Reach}_8 = \text{Reach}_7$!!!

Reachability algorithm for the tank system



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

Initialized Rectangular Automaton

rectangle ≡ set of the form $I_1 \times I_2 \times \dots \times I_n$ where each I_k 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

- $\mathcal{Q}$ ≡ set of discrete states $\mathbb{R}^n$ ≡ continuous state-space
- $f : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ vector field
- $\varphi : \mathcal{Q} \times \mathbb{R}^n \to \mathcal{Q}$ ≡ discrete transition
- $\rho : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$ ≡ reset map

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

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

$$\varphi(q,x) = \begin{cases} q_j & q = q_i,\ x \in R_{ji} \\ \vdots & \end{cases}$$

where all the R_{ji} are rectangles

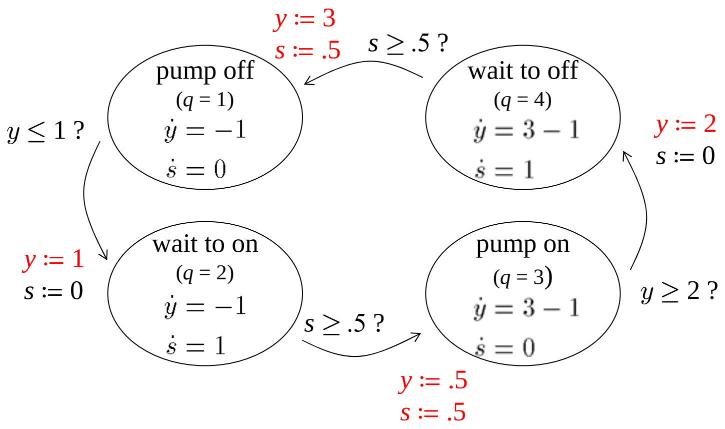
conditions for jumps are expressed by rectangles in *x*

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

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

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 $\mathcal{Q}$ is finite
2. vector field is constant in each discrete mode
3. jump conditions rectangular in *x*
4. resets independent of *x*

(1–3) rectangular automaton

(4) initialized

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

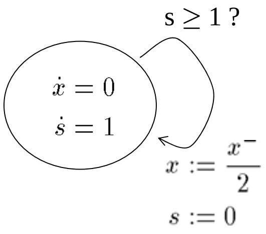
H is an ***initialized rectangular automaton*** if:

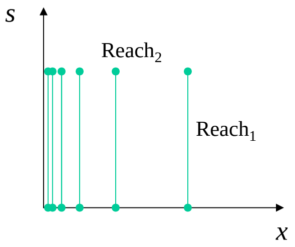
1. the set $\mathcal{Q}$ is finite
2. vector field is constant in each discrete mode
3. jump conditions rectangular in x
4. resets independent of x

} rectangular automaton (1–3)

} initialized (4)

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





$\mathcal{S}_0 := \{1,0\} = \text{Reach}_0$

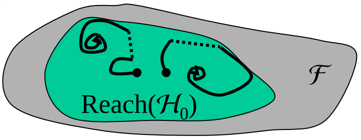
Back to safety…

Given: hybrid automaton H:

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

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

H satisfies a safety property $p(q, x) = \Box\,(q(t),x(t)) \in \mathcal{F}$, $\mathcal{F} \subset \mathcal{Q} \times \mathbb{R}^n$ 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 algorithm:

```
initialization:  Reach_{-1} = $\emptyset$
                 Reach_0 = $\mathcal{S}_0$
                 $i = 0$

loop:            while $\text{Reach}_i \neq \text{Reach}_{i-1}$ or $\text{Reach}_i \not\subset \mathcal{F}$ do
                     $\text{Reach}_{i+1} = \text{Reach}_i \cup \{s' \in \mathcal{S} : \exists\, s \in \text{Reach}_i,\ e \in \mathcal{E},\ (s,e,s') \in T\}$
                     $i = i + 1$

end:             if $\text{Reach}_i = \text{Reach}_{i-1}$ then $H$ satisfies $p$
                                          else $H$ does not satisfy $p$
```

algorithm can terminate immediately if one of the Reach_i is outside of $\mathcal{F}$

Backward reachability

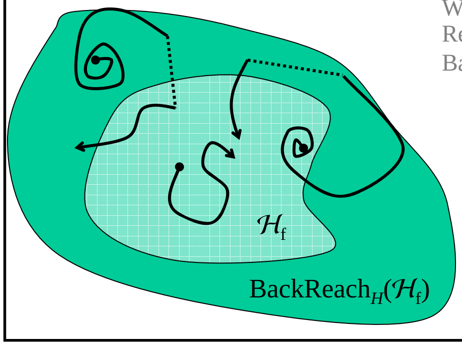
Given: hybrid automaton *H:*

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

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

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

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



What can you say about
$\text{Reach}_H(\text{BackReach}_H(\mathcal{H}_f))$
$\text{BackReach}_H(\text{Reach}_H(\mathcal{H}_0))$?

Backward reachability

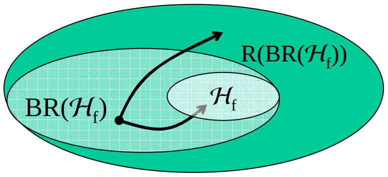
Given: hybrid automaton *H:*

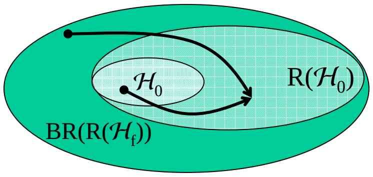
$$\dot{x} = f(q,x) \qquad (q,x) = \Phi(q^-, x^-) \qquad q(t) \in \mathcal{Q},\ x(t) \in \mathbb{R}^n,\ t \geq t_0$$

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

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

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


$\text{R}(\text{BR}(\mathcal{H}_f))$
$\text{BR}(\mathcal{H}_f)$
$\mathcal{H}_f$


$\mathcal{H}_0$
$\text{R}(\mathcal{H}_0)$
$\text{BR}(\text{R}(\mathcal{H}_f))$

In general
$\text{Reach}_H(\text{BackReach}_H(\mathcal{H}_f)) \supset \mathcal{H}_f$
$\text{BackReach}_H(\text{Reach}_H(\mathcal{H}_0)) \supset \mathcal{H}_0$

For deterministic systems
$\text{Reach}_H(\text{BackReach}_H(\mathcal{H}_f)) = \text{Reach}_H(\mathcal{H}_f) \supset \mathcal{H}_f$

For backwards-in-time deterministic systems
$\text{BackReach}_H(\text{Reach}_H(\mathcal{H}_0)) = \text{BackReach}_H(\mathcal{H}_0) \supset \mathcal{H}_0$

Backward reachability

Given: hybrid automaton *H:*

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

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

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

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

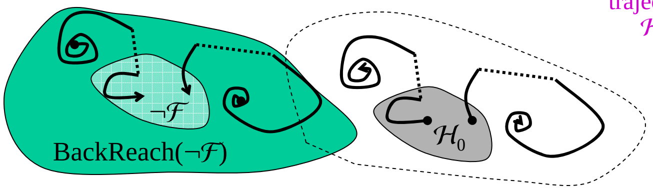
H satisfies a safety property

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

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

$$\text{BackReach}_H(\neg\, \mathcal{F}) \cap \mathcal{H}_0 = \emptyset$$

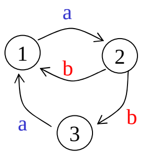
every point in every trajectory starting in $\mathcal{H}_0$ satisfies *p*



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

Transition system

transition system *T* $\begin{cases} \mathcal{S} & \equiv \text{set of states (finite or infinite)} \\ \mathcal{E} & \equiv \text{alphabet of events (finite or infinite)} \\ \mathrm{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S} & \equiv \text{transition relation} \end{cases}$



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

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

Given a set of initial states $\mathcal{S}_0 \subset \mathcal{S}$:

$\text{Reach}_T(\mathcal{S}_0) \equiv$ set of states $s \in \mathcal{S}$ for which there is a finite execution that starts in $\mathcal{S}_0$ and ends at s

Given a set of final states $\mathcal{S}_f \subset \mathcal{S}$:

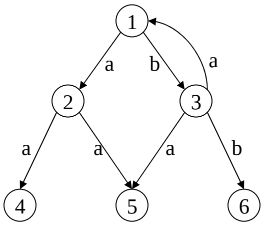
$\text{BackReach}_T(\mathcal{S}_f) \equiv$ set of states $s \in \mathcal{S}$ for which there is a finite execution that starts at s and ends in $\mathcal{S}_f$

Backward reachability algorithm

Backward reachability algorithm:

```
initialization:  $BReach_{-1} = \emptyset$
                 $BReach_0 = \mathcal{S}_f$
                 $i = 0$
loop:            while $BReach_i \neq BReach_{i-1}$ do
                     $BReach_{i+1} = BReach_i \cup \{s \in \mathcal{S} : \exists s' \in BReach_i, e \in \mathcal{E}$ s.t. $(s,e,s') \in T\}$
                     $i = i + 1$
```

states from where one can transition into $BReach_i$



$\mathcal{S}_f = \{5\}$

$BReach_0 = \{5\}$

$BReach_1 = \{2,3,5\}$

$BReach_2 = \{1,2,3,5\}$

$BReach_3 = \{1,2,3,5\}$

$BackReach_T(\{5\}) = \{1,2,3,5\}$

Backward reachability algorithm

Backward reachability algorithm:

```
initialization:  $BReach_{-1} = \emptyset$
                 $BReach_0 = \mathcal{S}_f$
                 $i = 0$
loop:            while $BReach_i \neq BReach_{i-1}$ do
                     $BReach_{i+1} = BReach_i \cup \{s \in \mathcal{S} : \exists s' \in BReach_i, e \in \mathcal{E}$ s.t. $(s,e,s') \in T\}$
                     $i = i + 1$
```

states from where one can transition into $BReach_i$

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 $BReach_i = BackReach_T(\mathcal{S}_f)$

Why?

(i) In each iteration the number of elements in $BReach_i$ 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 $\mathcal{S}$ (minus the number of elements in $\mathcal{S}_0$).

(ii) $BReach_i \equiv$ 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 $Reach_i$

Invariant set algorithm

Invariant set algorithm (backward reachability working with complements):

```
initialization: $\text{Inv}_{-1} = \mathcal{S}$
                $\text{Inv}_0 = \neg\, \mathcal{S}_f$
                $i = 0$
loop:           while $\text{Inv}_i \neq \text{Inv}_{i-1}$ do
                    $\text{Inv}_{i+1} = \text{Inv}_i \cap \{s \in \mathcal{S} : \forall\, s' \notin \text{Inv}_i,\ e \in \mathcal{E} \text{ s.t. } (s,e,s') \notin T\}$
                    $i = i + 1$
```

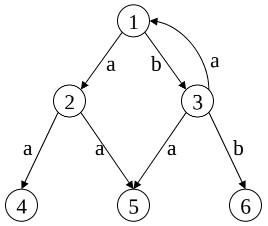
$\text{Inv}_\text{i} := \neg\ \text{BReach}_\text{i}$

complement of previous set

$\{s \in \mathcal{S} : \exists\, s' \notin \text{Inv}_i,\ e \in \mathcal{E} \text{ s.t. } (s,e,s') \in T\}$

(new set can be interpreted as

"states for which there is no transition out of Inv_i")



$\mathcal{F} = \{1,2,3,4,6\}$ $(\mathcal{S}_\text{f} = \neg\, \mathcal{F} = \{5\})$

$\text{Inv}_0 = \{1,2,3,4,6\}$

$\text{Inv}_1 = \{1,4,6\}$

$\text{Inv}_2 = \{4,6\}$

$\text{Inv}_3 = \{4,6\}$

$\text{Inv}_T(\{5\}) = \{4,6\} = \neg\ \text{BackReach}T(\{5\})$

consistent with previous computation:

$\text{BackReach}T(\{5\}) = \{1,2,3,5\}$

Invariant set algorithm

Invariant set algorithm (backward reachability working with complements):

```
initialization: $\text{Inv}_{-1} = \mathcal{S}$
                $\text{Inv}_0 = \neg\, \mathcal{S}_f$
                $i = 0$
loop:           while $\text{Inv}_i \neq \text{Inv}_{i-1}$ do
                    $\text{Inv}_{i+1} = \text{Inv}_i \cap \{s \in \mathcal{S} : \forall\, s' \notin \text{Inv}_i,\ e \in \mathcal{E} \text{ s.t. } (s,e,s') \notin T\}$
                    $i = i + 1$
```

$\text{Inv}_\text{i} := \neg\ \text{BReach}_\text{i}$

states for which there is

no transition out of Inv_i

Theorem: If $\mathcal{S}$ <u>is finite</u> then

(i) the algorithm finishes in a finite number of steps and

(ii) upon exiting the while loop

$\text{Inv}_i = \text{Inv}_T(\neg\, \mathcal{S}_\text{f}) \equiv$ largest invariant set contained in $\neg\, \mathcal{S}_\text{f}$ $(= \mathcal{F})$

Why?

(i) In each iteration the number of elements in Inv_i decreases by at least 1. There can only be as many iterations as the number of elements in $\mathcal{S} \backslash \mathcal{S}_\text{f}$.

(ii) Upon exiting: $\text{Inv} \subset \{s \in \mathcal{S} : \forall\, s' \notin \text{Inv},\ e \in \mathcal{E},\ (s,e,s') \notin T\}$

set of states for which there is no

transition out of Inv

⇓

Inv is invariant set

Invariant sets

Given: hybrid automaton *H:*

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

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

$\text{Inv}_H(\mathcal{H}_f) \equiv$ 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

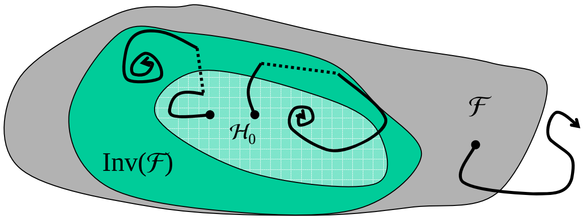
$$p(q, x) = \Box\, \big(q(t),x(t)\big) \in \mathcal{F}$$

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

$$\text{BackReach}_H(\neg\, \mathcal{F}) \cap \mathcal{H}_0 = \emptyset$$

or equivalently

$$\neg\, \text{Inv}_H(\mathcal{F}) \cap \mathcal{H}_0 = \emptyset \ \Leftrightarrow\ \mathcal{H}_0 \subset \text{Inv}_H(\mathcal{F})$$



$\text{Inv}_H(\mathcal{F}) \equiv$ largest set of initial states for which the property is satisfied

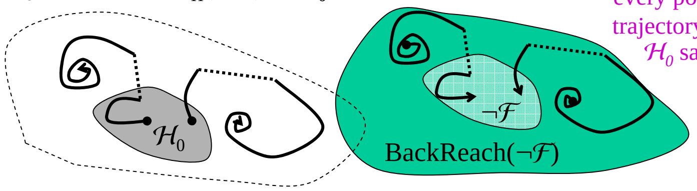
Back to safety (again)…

Given: hybrid automaton *H:*

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

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

H satisfies a safety property $p(q, x) = \Box\, \big(q(t),x(t)\big) \in \mathcal{F} \subset Q \times \mathbb{R}^n$ (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:  $\text{BReach}_{-1} = \emptyset$
                 $\text{BReach}_0 = \mathcal{S}_f := \neg\, \mathcal{F}$
                 $i = 0$

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

end:             if $\text{Reach}_i = \text{Reach}_{i-1}$ then $H$ satisfies $p$ else $H$ does not satisfy $p$
```

algorithm can terminate immediately if one of the BReach_i intersects $\mathcal{H}_0$

Controller design based on backward reachability

Backwards Reachability algorithm:

```
initialization:  $\text{BReach}_{-1} = \emptyset$
                 $\text{BReach}_0 = \mathcal{S}_f := \neg\mathcal{F}$
                 $i = 0$

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

end:             if $\text{Reach}_i = \text{Reach}_{i-1}$ then $H$ satisfies $p$ else $H$ does not satisfy $p$
```

algorithm can terminate immediately if one of the BReach_i intersects $\mathcal{H}_0$

When one obtains $\text{BReach}_{i+1} \cap \mathcal{H}_0 \neq \emptyset$ it is because

$$\{s \in \mathcal{S} : \exists\, s' \in \text{BReach}_i,\ e \in \mathcal{E},\ (s,e,s') \in \text{T}\} \cap \mathcal{H}_0 \neq \emptyset$$

therefore

$$\exists\, s \in \mathcal{H}_0,\ s' \in \text{BReach}_i,\ e \in \mathcal{E} : (s,e,s') \in \text{T}$$

transition from $\mathcal{H}_0$ to BReach_i

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 $s' \in \text{BReach}_i,\ e \in \mathcal{E},\ s \in \mathcal{H}_0$

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