

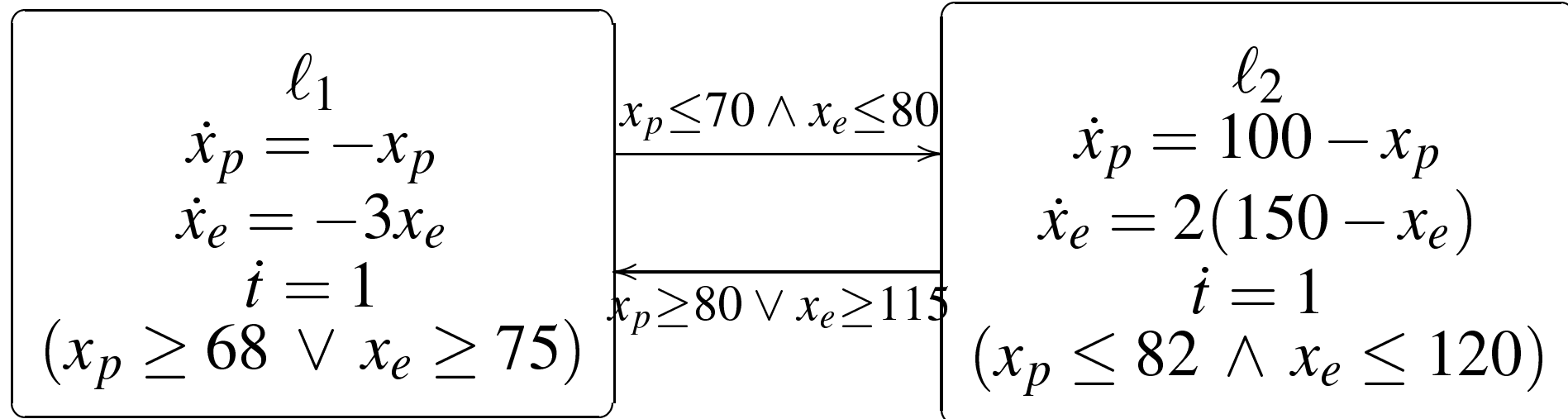


Stability Proofs for Hybrid Systems

Silke Wagner
Andreas Podelski

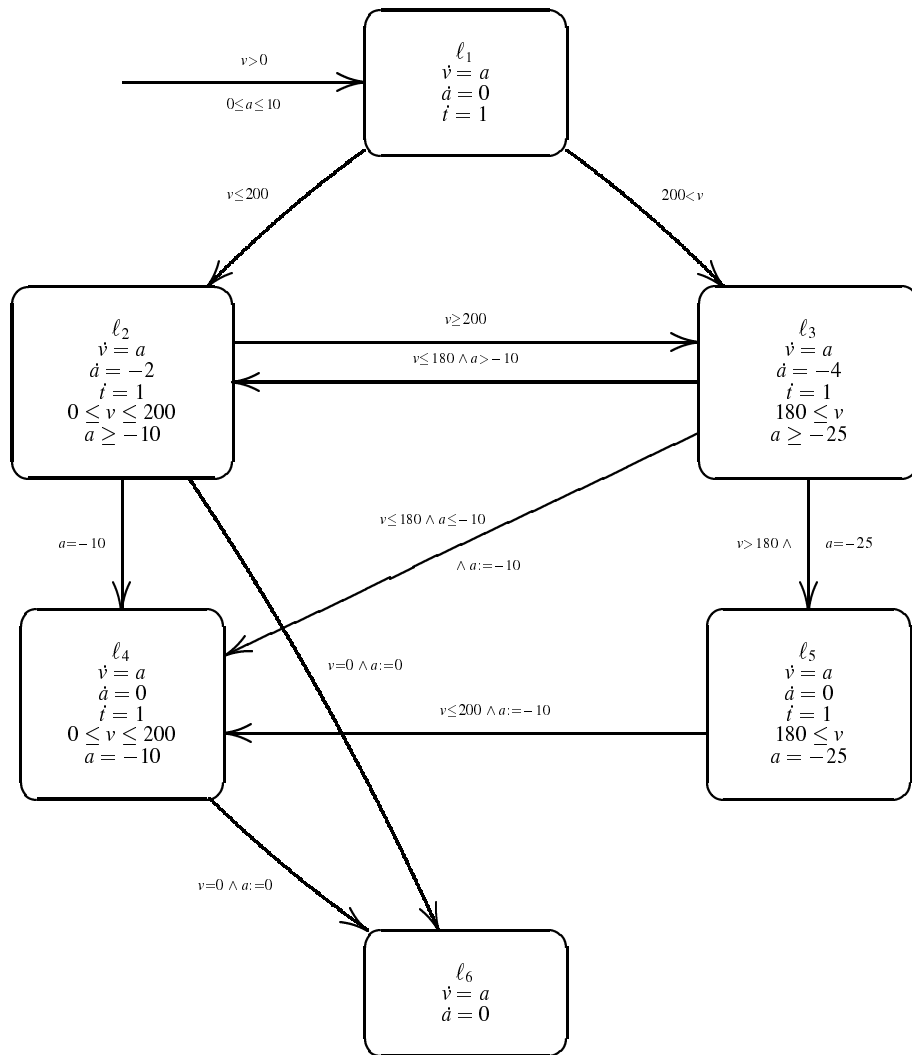


Motivation



- heater + internal engine
- internal engine may overheat
- **correctness property:** $x_p \in [65, 82]$

Motivation



- speed v
- two brakes
- time delay between ordering brake application and reaching brake effort
- **correctness property:** $v \leq 0$



Outline

1. Region stability

- Definition
- Related notions

2. Method for linear dynamics

- Sound and complete condition
- Algorithm
- Correctness of the algorithm

3. Proof rule for arbitrary hybrid systems

4. Extensions

5. Benchmarks



Stability w.r.t. Interval Regions

- DEFINITION:

A hybrid system A is stable w.r.t. a region φ if for every trajectory τ there exists a time point t_0 such that from then on the trajectory is always in the region φ .

$$\forall \tau \exists t_0 \forall t \geq t_0 : \tau(t) \in \varphi$$



Stability w.r.t. Interval Regions

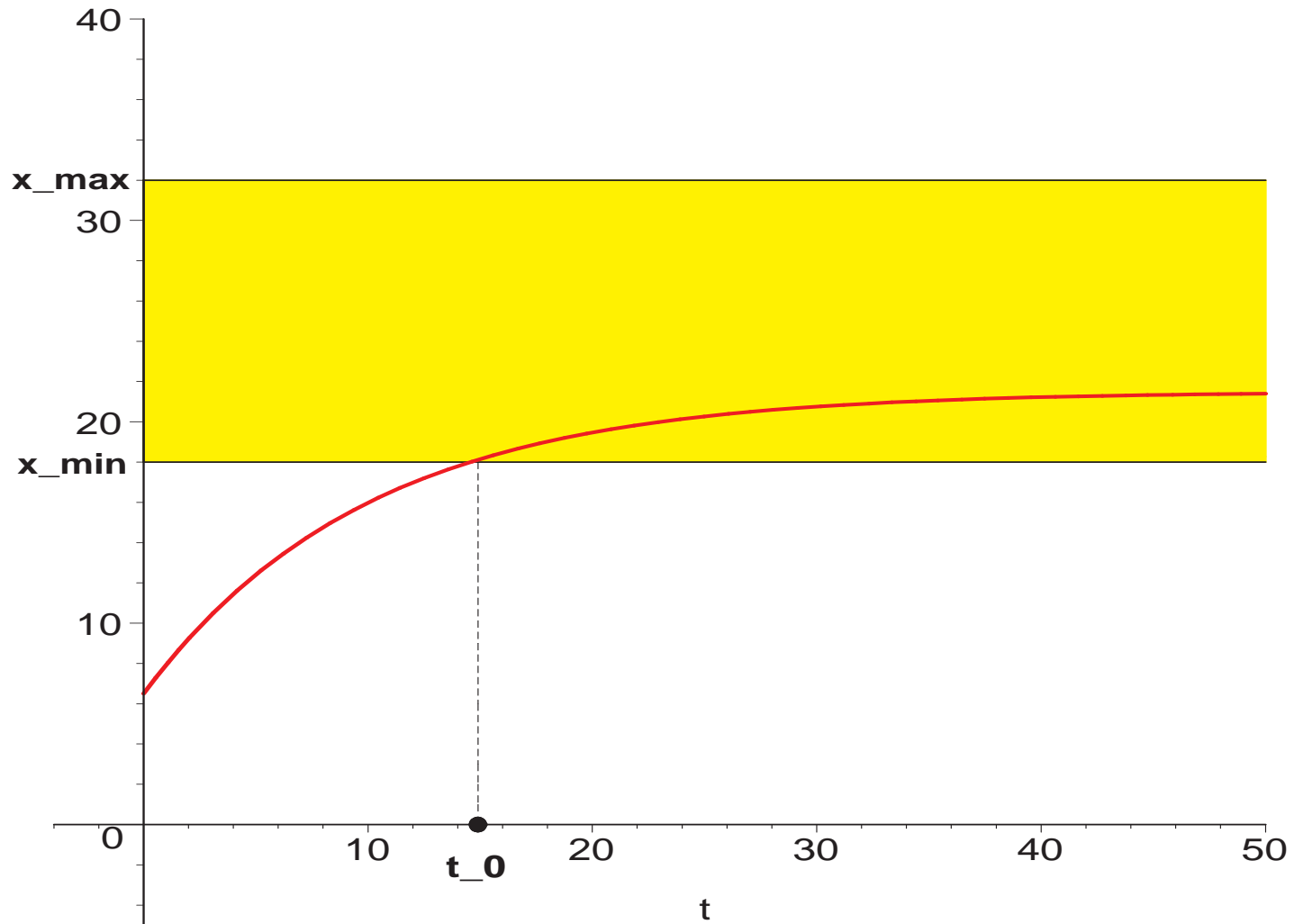
- DEFINITION:

A hybrid system A is stable w.r.t. a region φ if for every trajectory τ there exists a time point t_0 such that from then on the trajectory is always in the region φ .

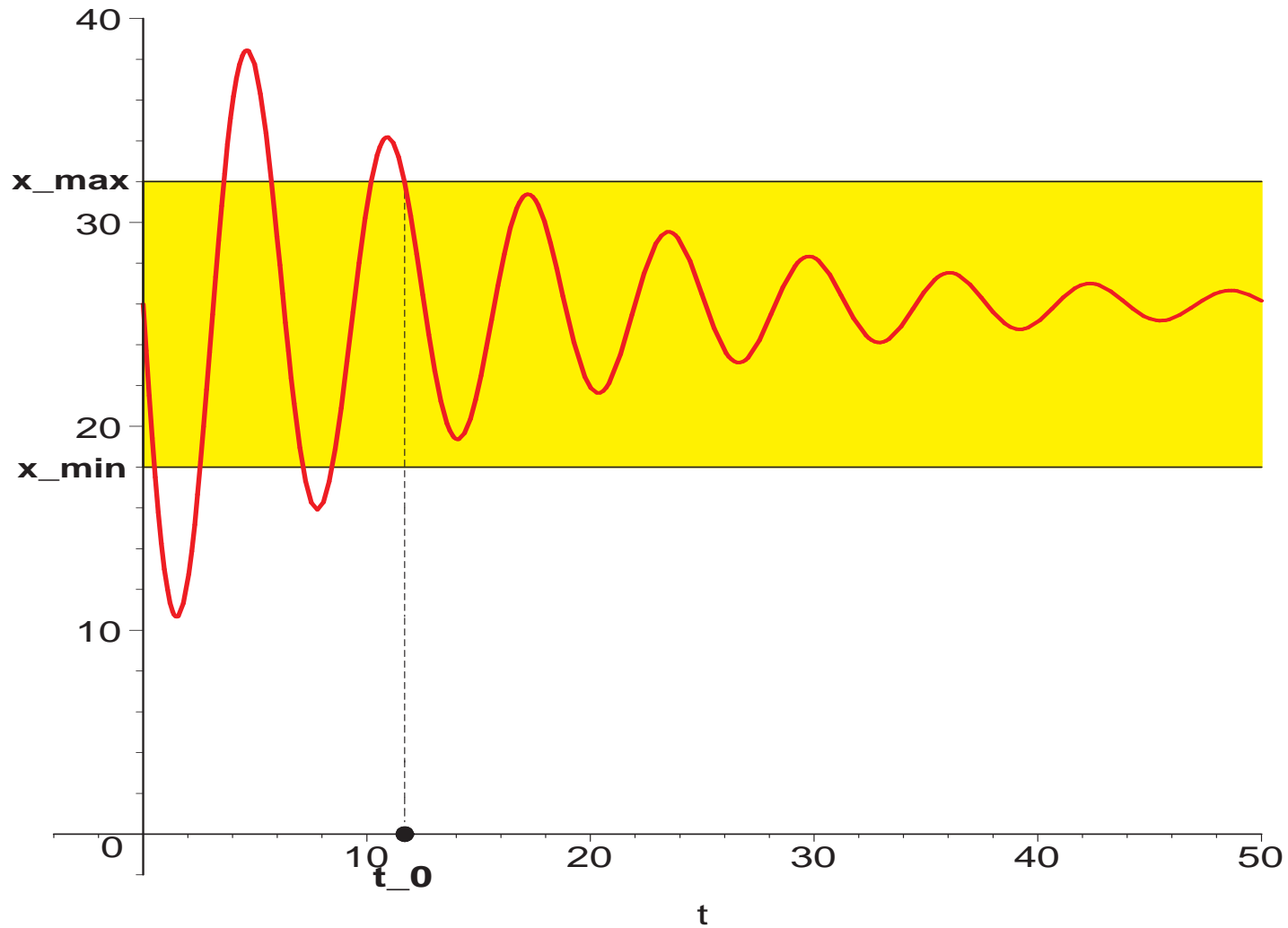
$$\forall \tau \exists t_0 \forall t \geq t_0 : \tau(t) \in \varphi$$

- interval region: $\varphi \equiv x \in [x_{\min}, x_{\max}]$

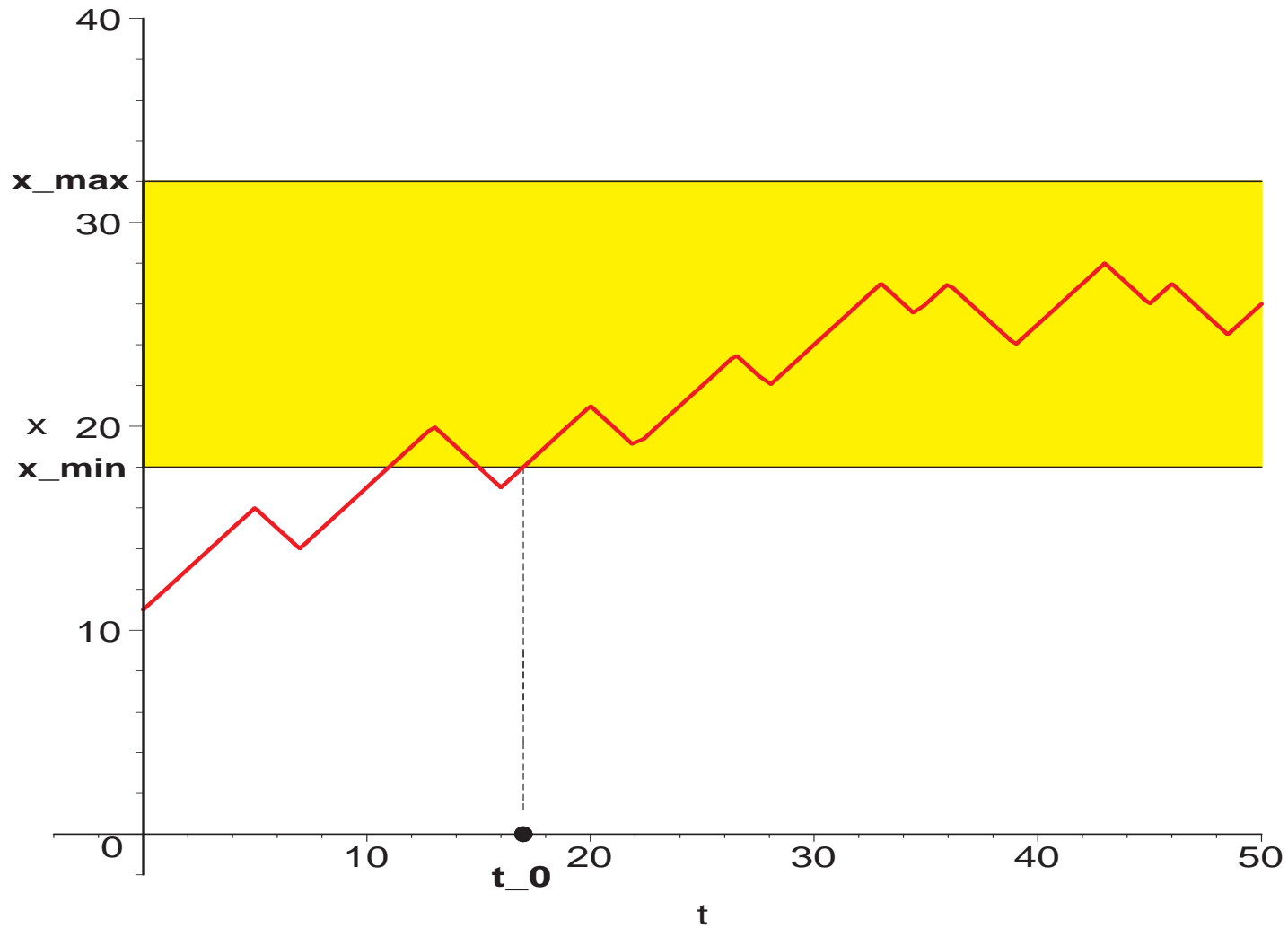
Stability w.r.t. Interval Regions



Stability w.r.t. Interval Regions



Stability w.r.t. Interval Regions





Stability w.r.t. Equilibrium States

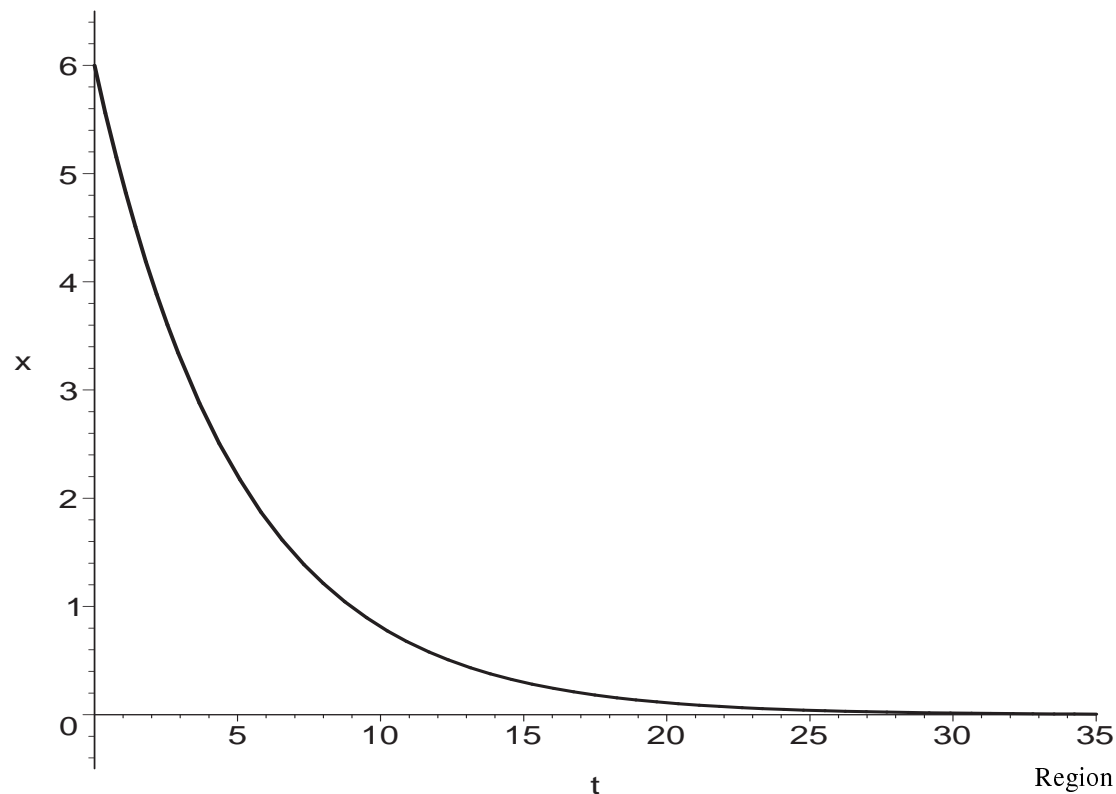
- Asymptotic stability:

“Each trajectory of the hybrid system that starts within a small region around the equilibrium point will converge to the equilibrium point.”

Stability w.r.t. Equilibrium States

- Asymptotic stability:

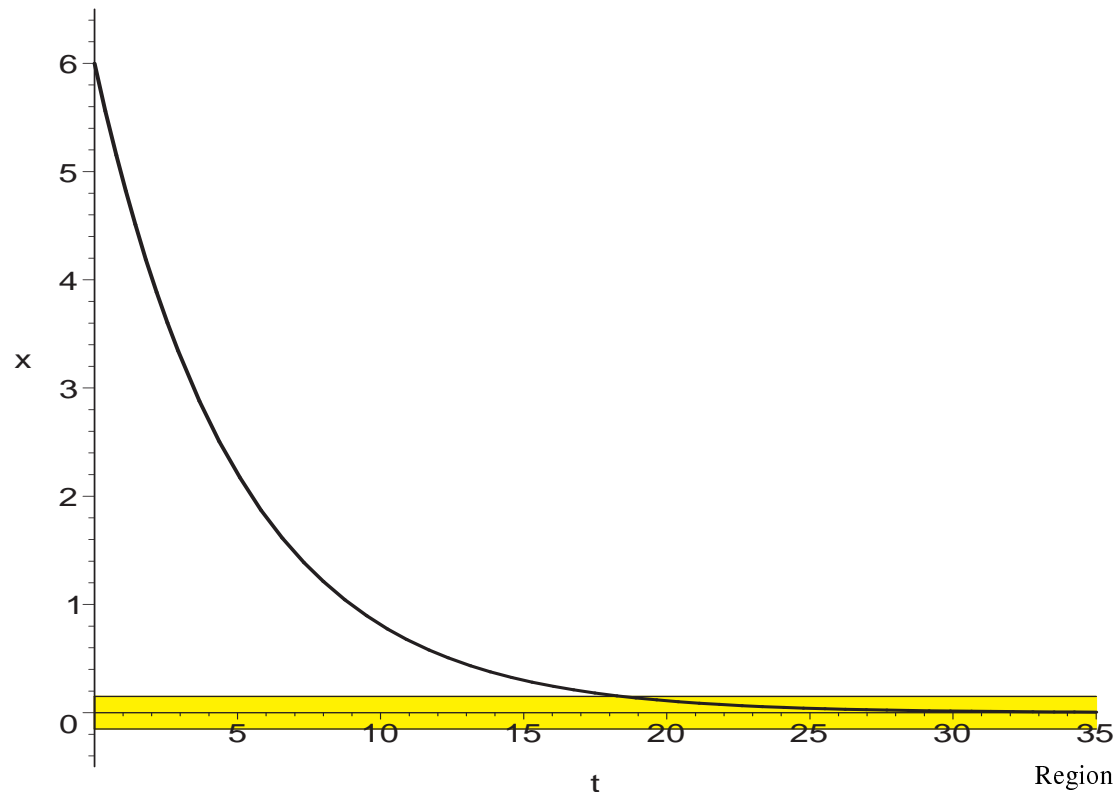
“Each trajectory of the hybrid system that starts within a small region around the equilibrium point will converge to the equilibrium point.”



Stability w.r.t. Equilibrium States

- Asymptotic stability:

“Each trajectory of the hybrid system that starts within a small region around the equilibrium point will converge to the equilibrium point.”





Stability w.r.t. Equilibrium States

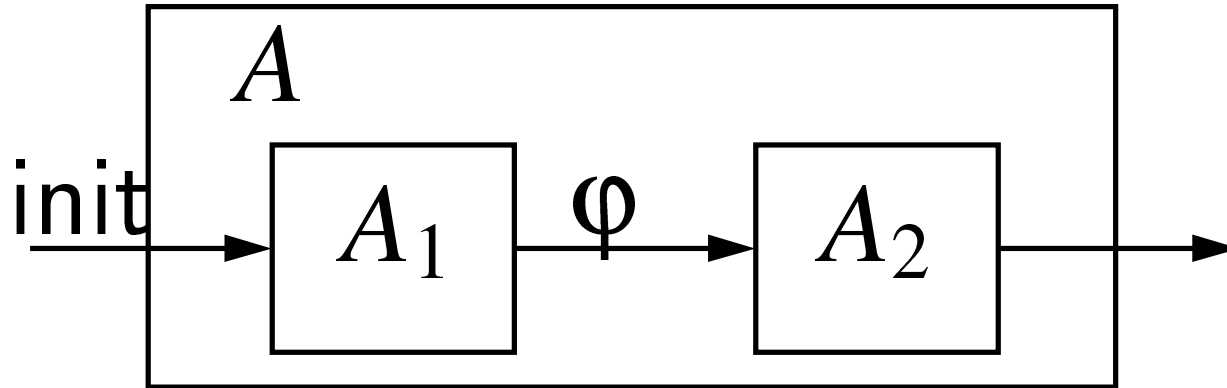
- Asymptotic stability:
“Each trajectory of the hybrid system that starts within a small region around the equilibrium point will converge to the equilibrium point.”
- Asymptotic stability w.r.t. x_0 corresponds to region stability w.r.t. $(x_0 - \varepsilon, x_0 + \varepsilon)$.



Stability w.r.t. Equilibrium States

- Asymptotic stability:
“Each trajectory of the hybrid system that starts within a small region around the equilibrium point will converge to the equilibrium point.”
- Asymptotic stability w.r.t. x_0 corresponds to region stability w.r.t. $(x_0 - \varepsilon, x_0 + \varepsilon)$.
- Region stability is not always expressible in terms of asymptotic stability.

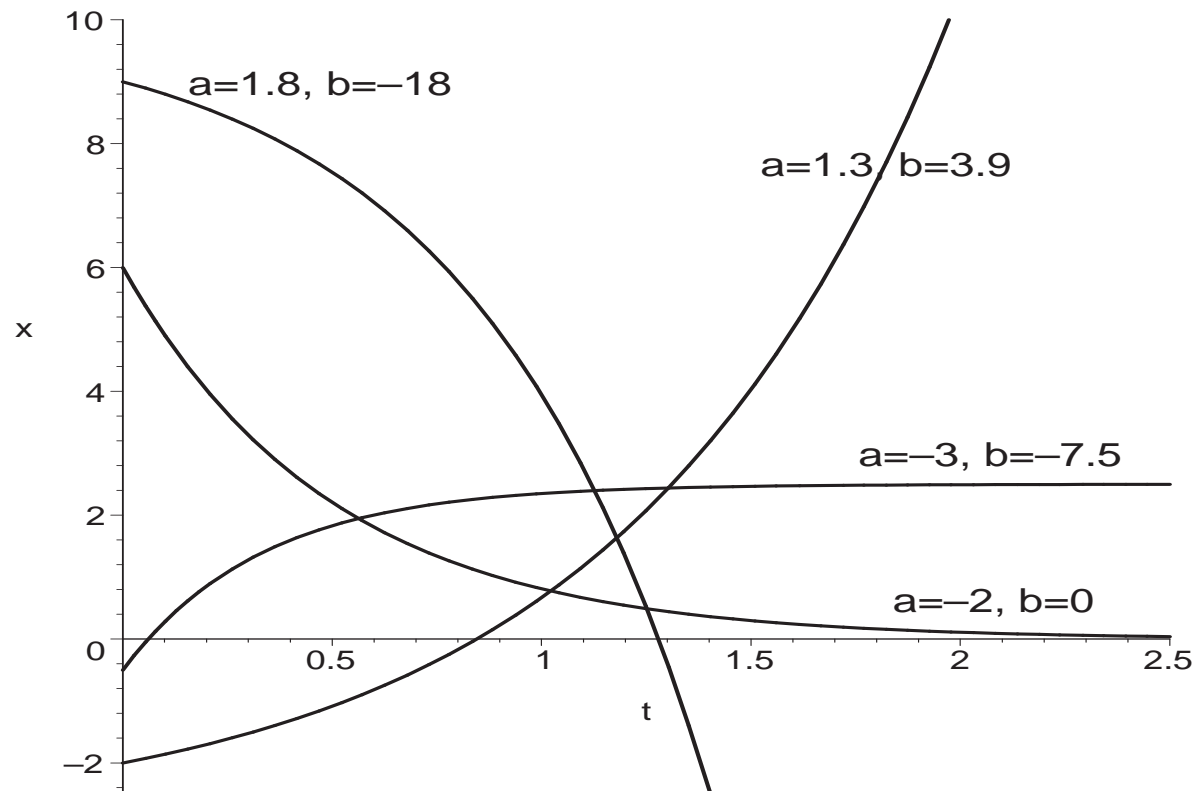
Combination of the Methods



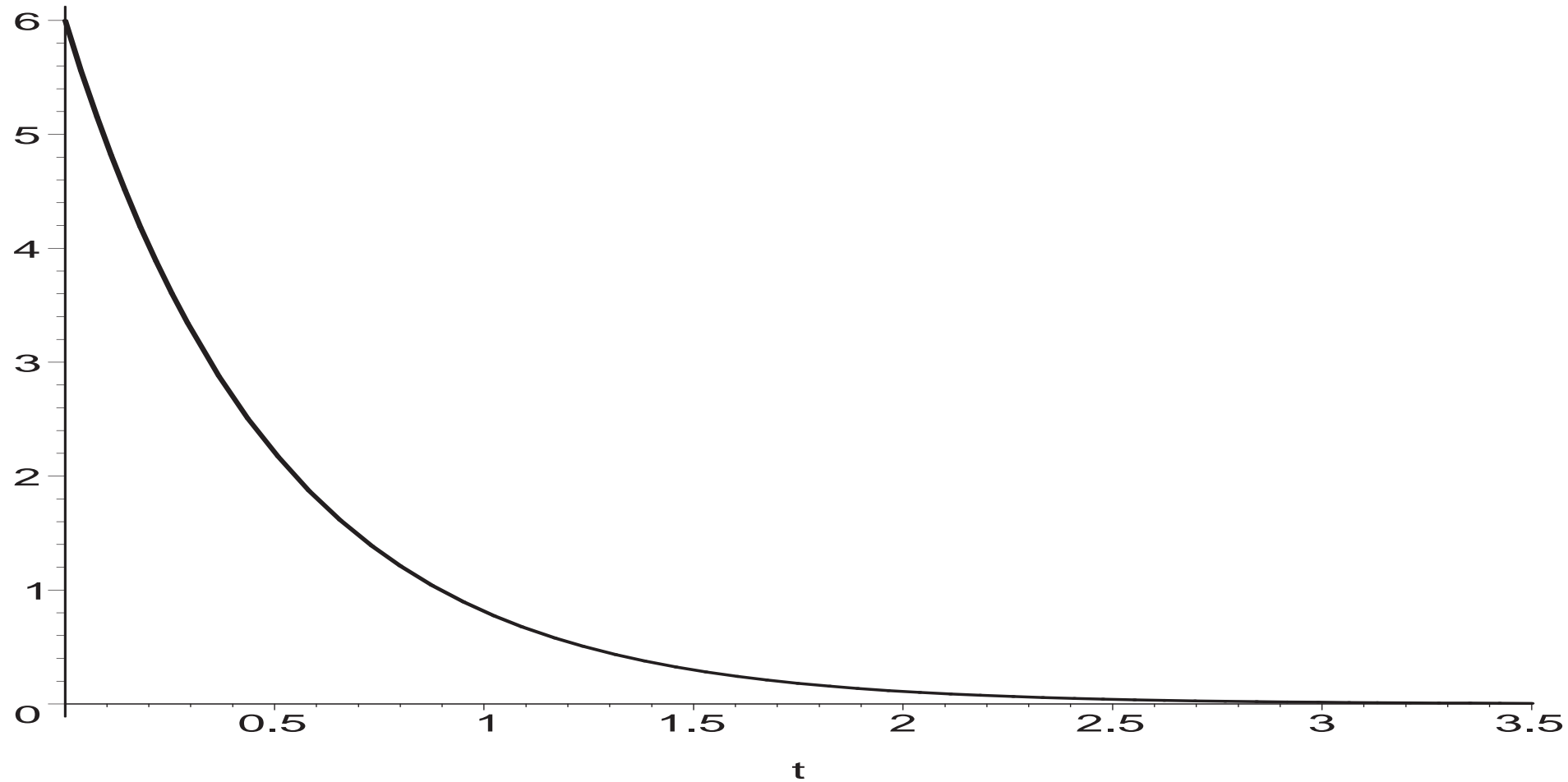
- A_1 stable w.r.t. region φ
- A_2 stable w.r.t. equilibrium x_0
- A asymptotically stable
- Asymptotic stability for larger set of initial states
- Can speed up region stability proof

Region Stability for Linear Dynamics

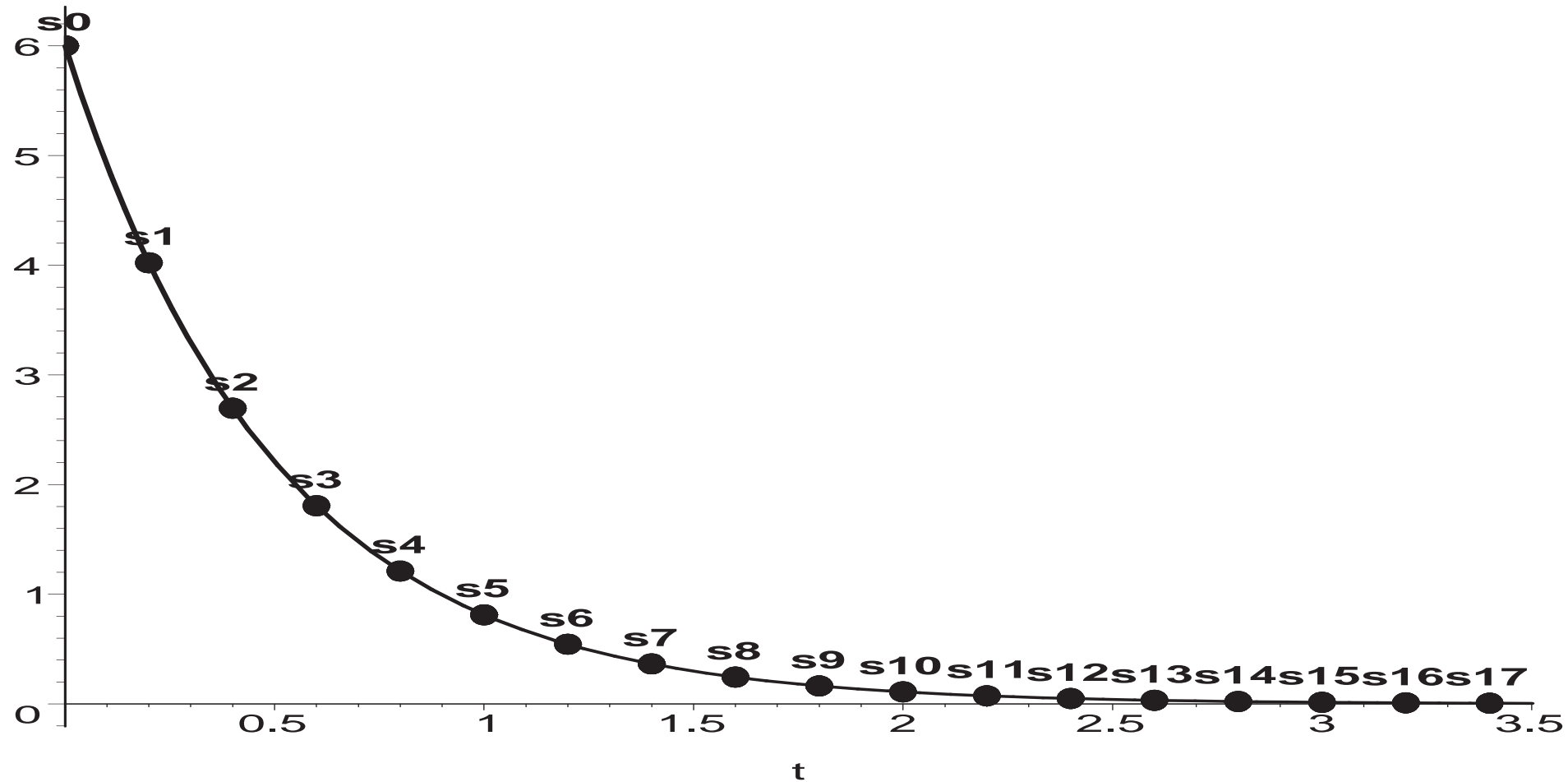
$$\dot{x} = ax + b, \quad a, b \in \mathbb{R}$$



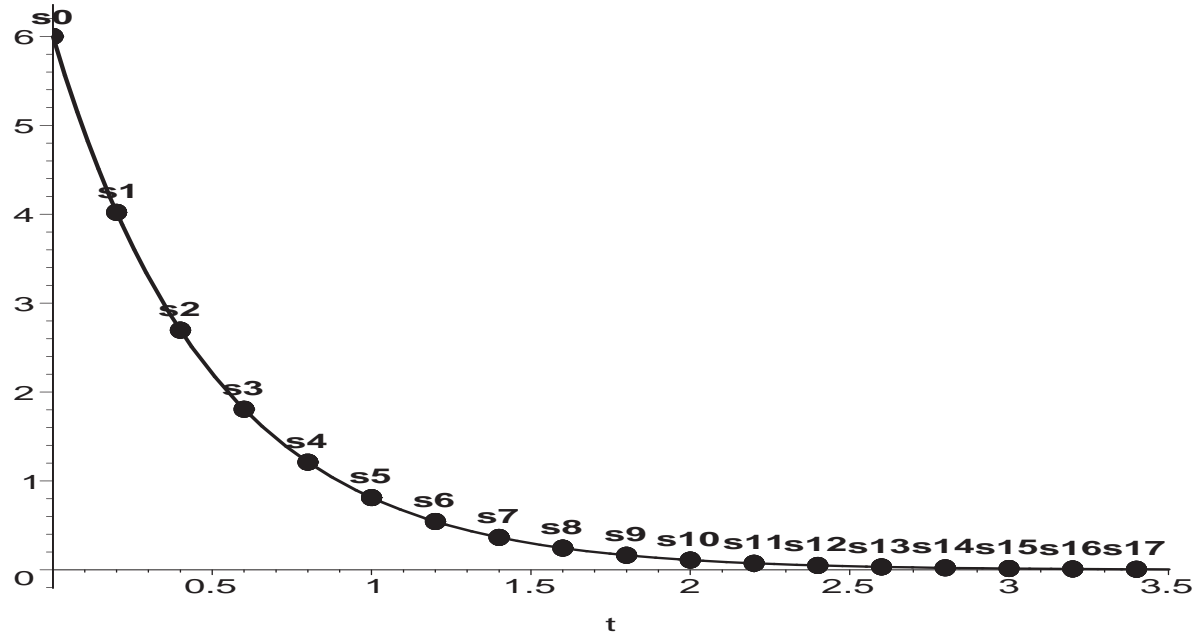
Region Stability for Linear Dynamics



Region Stability for Linear Dynamics



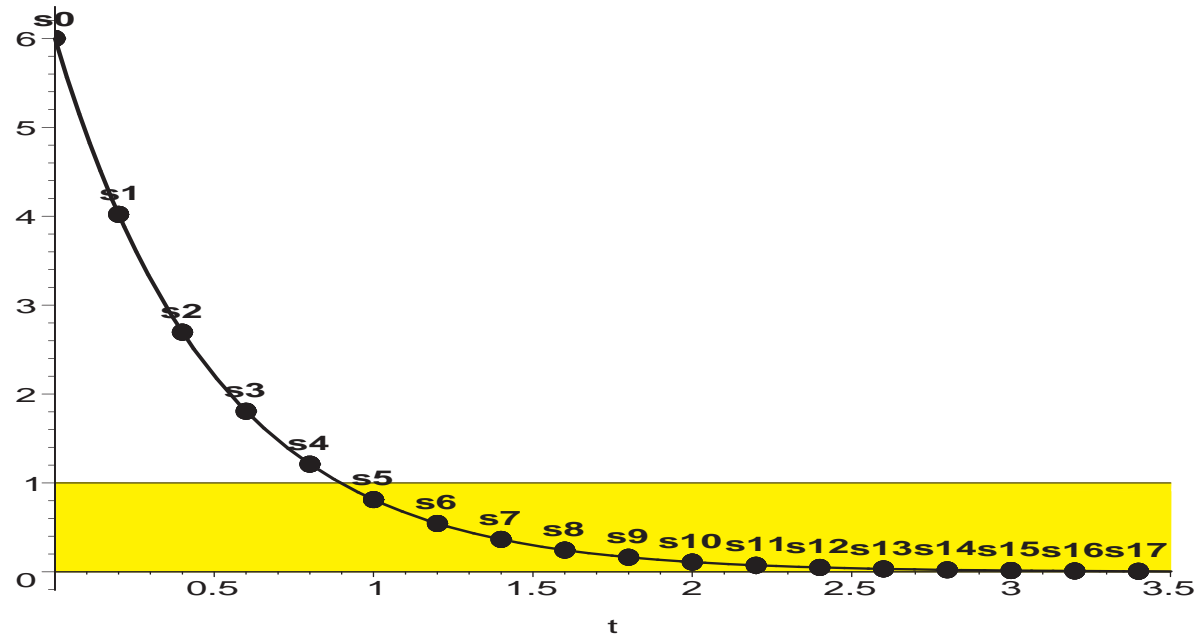
Region Stability for Linear Dynamics



(a) $\forall i : s_i \in \tau$

(b) $\forall i : t_{i+1} - t_i > \delta > 0$

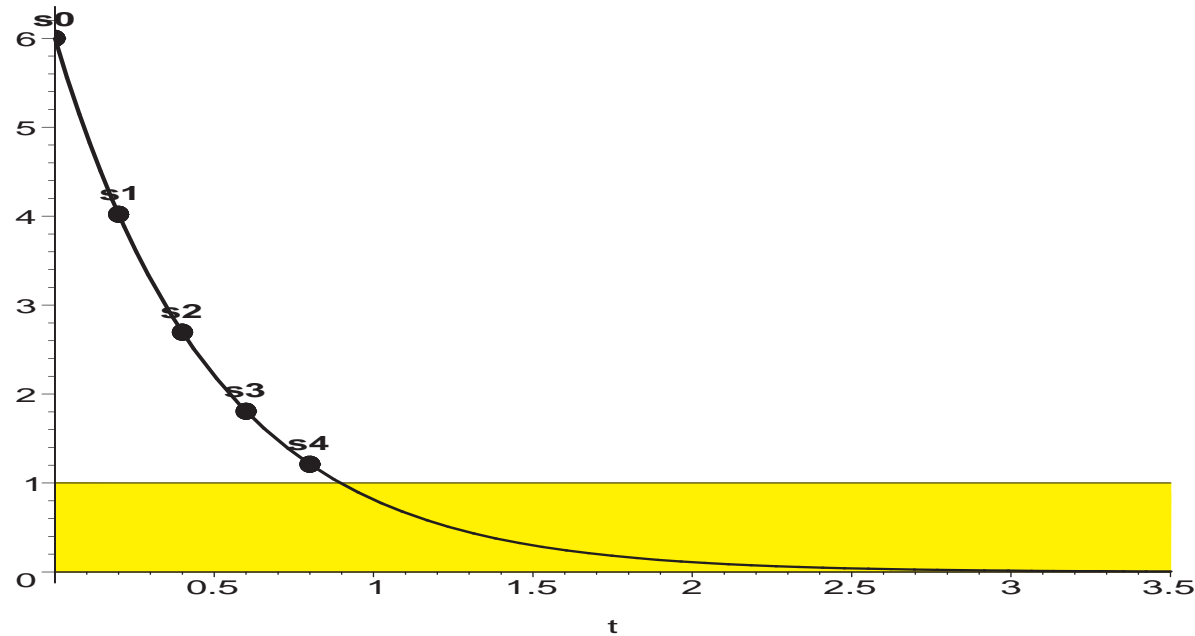
Region Stability for Linear Dynamics



(a) $\forall i : s_i \in \tau$

(b) $\forall i : t_{i+1} - t_i > \delta > 0$

Region Stability for Linear Dynamics



(a) $\forall i : s_i \in \tau$

(b) $\forall i : t_{i+1} - t_i > \delta > 0$

(c) $\forall i : s_i \notin \varphi$



Region Stability for Linear Dynamics

SNAPSHOT SEQUENCE:

(a) $\forall i : s_i \in \tau$

(b) $\forall i : t_{i+1} - t_i > \delta > 0$

(c) $\forall i : s_i \notin \varphi$



Region Stability for Linear Dynamics

CONDITION:

All snapshot sequences must be finite.

\Rightarrow Region stability for linear dynamical systems.



Implementation

Two issues:

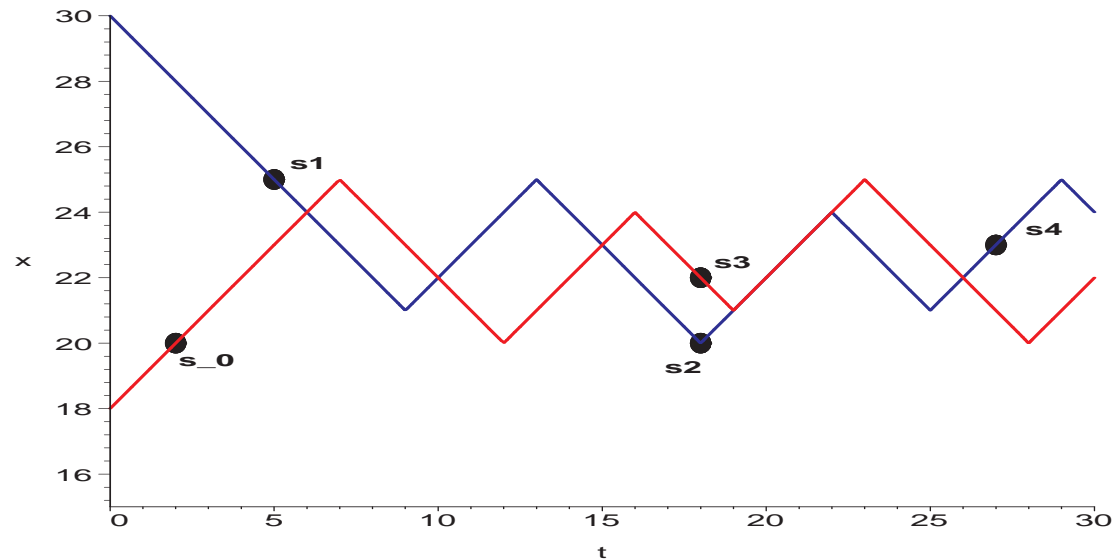
1. Computation of all snapshot sequences
2. Finiteness test.



Computation of Snapshot Sequences

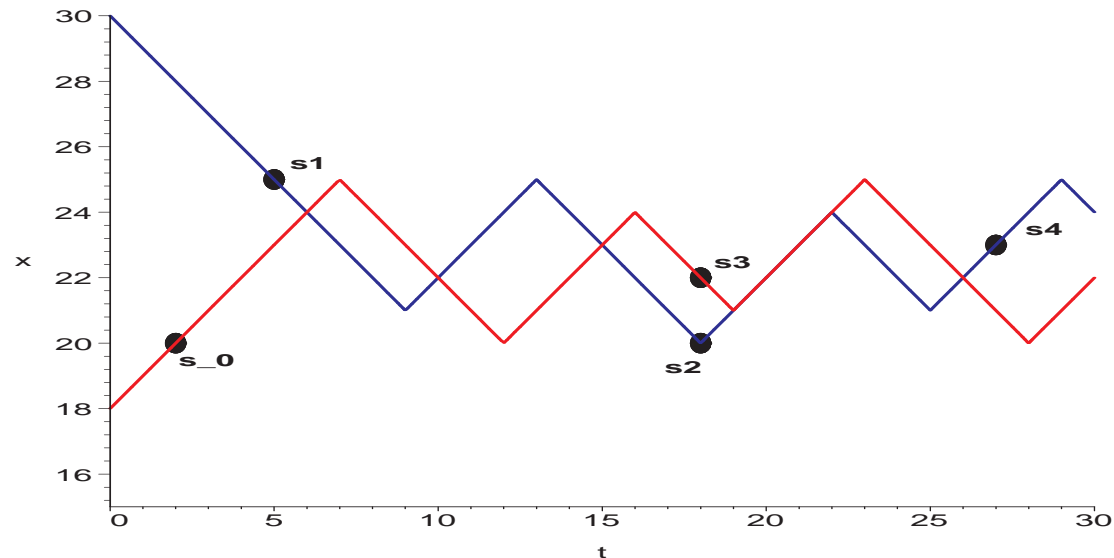
- Via reachability analysis:
- Compute (an overapproximation of) the set of all states on all trajectories of the system

Computation of Snapshot Sequences



$$\text{Reach} = \{s_0, s_1, s_2, s_3, s_4, \dots\}$$

Computation of Snapshot Sequences



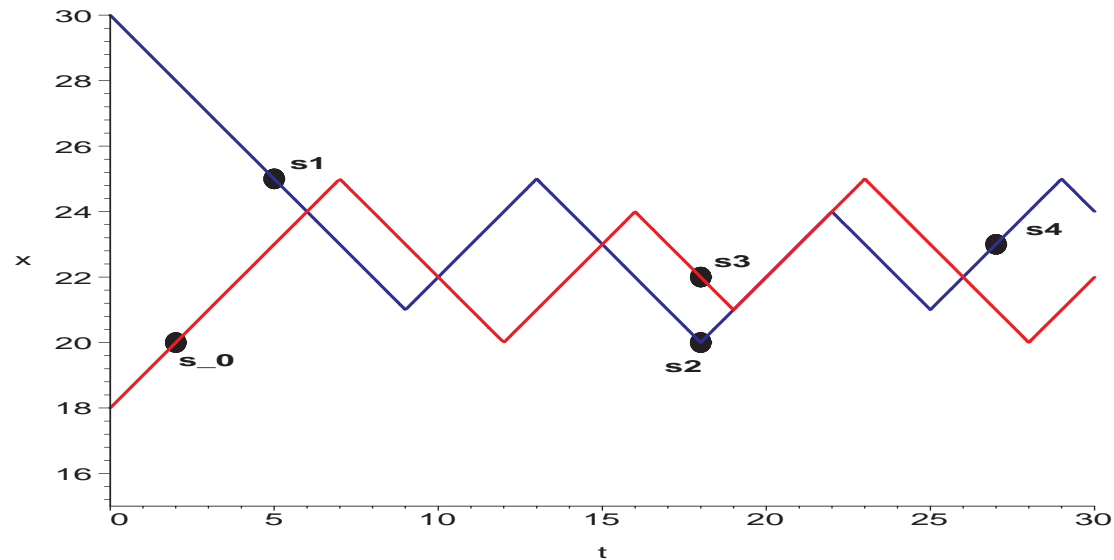
$$\text{Reach} = \{s_0, s_1, s_2, s_3, s_4, \dots\}$$

How to identify states on the same trajectory???

Unary reachability is not good enough!

Computation of Snapshot Sequences

Binary reachability analysis:



$$\text{BinReach} = \{(s_1, s_2), (s_2, s_4), (s_0, s_3), \dots\}$$



Representation of Snapshot Sequences

REPRESENTATION of all snapshot sequences:

- implicit
- by constraints that denote **binary relations**

$$\left. \begin{array}{ll} r : & \text{binary relation} \\ s_0, s_1, \dots : & \text{snapshot seq.} \end{array} \right\} \forall i : (s_i, s_{i+1}) \in r$$



Representation of Snapshot Sequences

The binary relation

$$x > 0, \quad x - x' = 1$$

would e.g. represent snapshot sequences of the system

$$\longrightarrow \boxed{\dot{x} = -1} \quad \varphi \equiv x \leq 0.$$



Computation of Binary Relations

STEP 1: System transformation

$$A \rightarrow A^T$$

such that

the **unary** reachability relations of A^T are

the **binary** reachability relations of A

that together represent all snapshot sequences.

STEP 2: Unary **reachability analysis** for A^T .



System Transformation

$$\begin{array}{l} x > 0 \\ \rightarrow \end{array} \boxed{\begin{array}{l} \ell_1 \\ \dot{x} = -1 \end{array}}, \quad \varphi \equiv x \leq 1$$

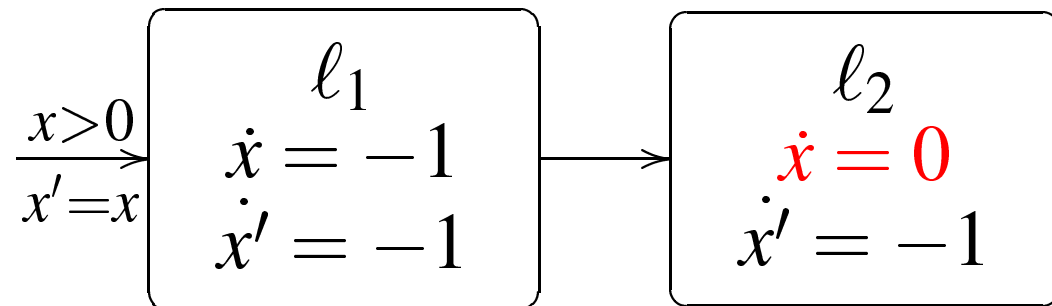


System Transformation

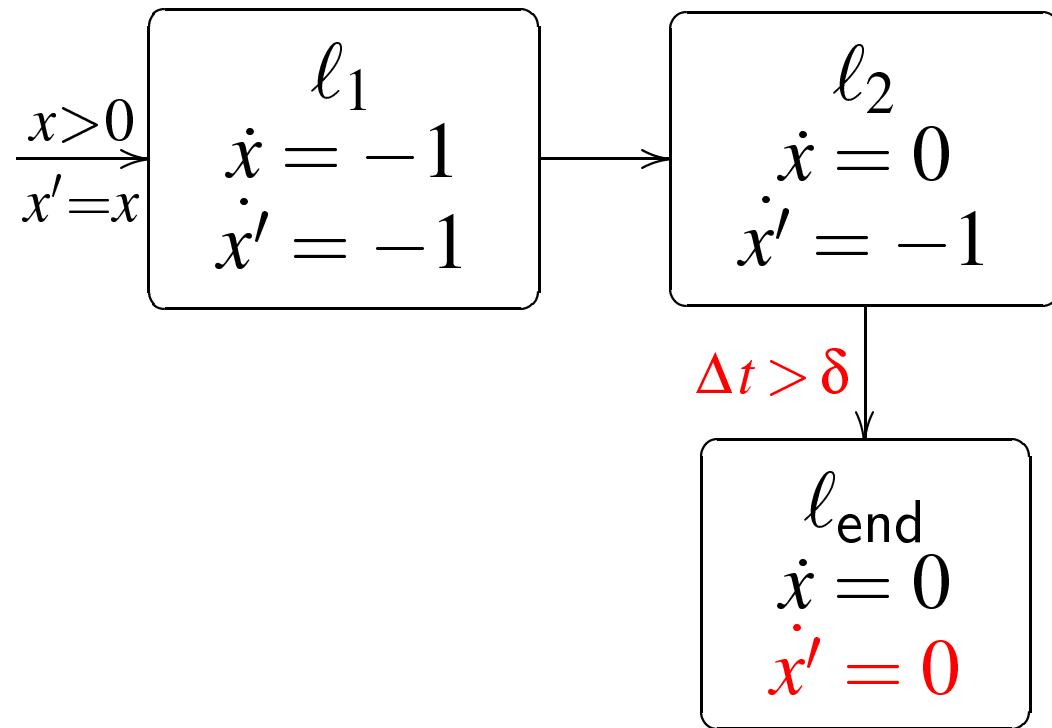
$$\begin{array}{l} \xrightarrow{x > 0} \\ x' = x \end{array} \boxed{\begin{array}{l} \ell_1 \\ \dot{x} = -1 \\ \dot{x}' = -1 \end{array}}$$



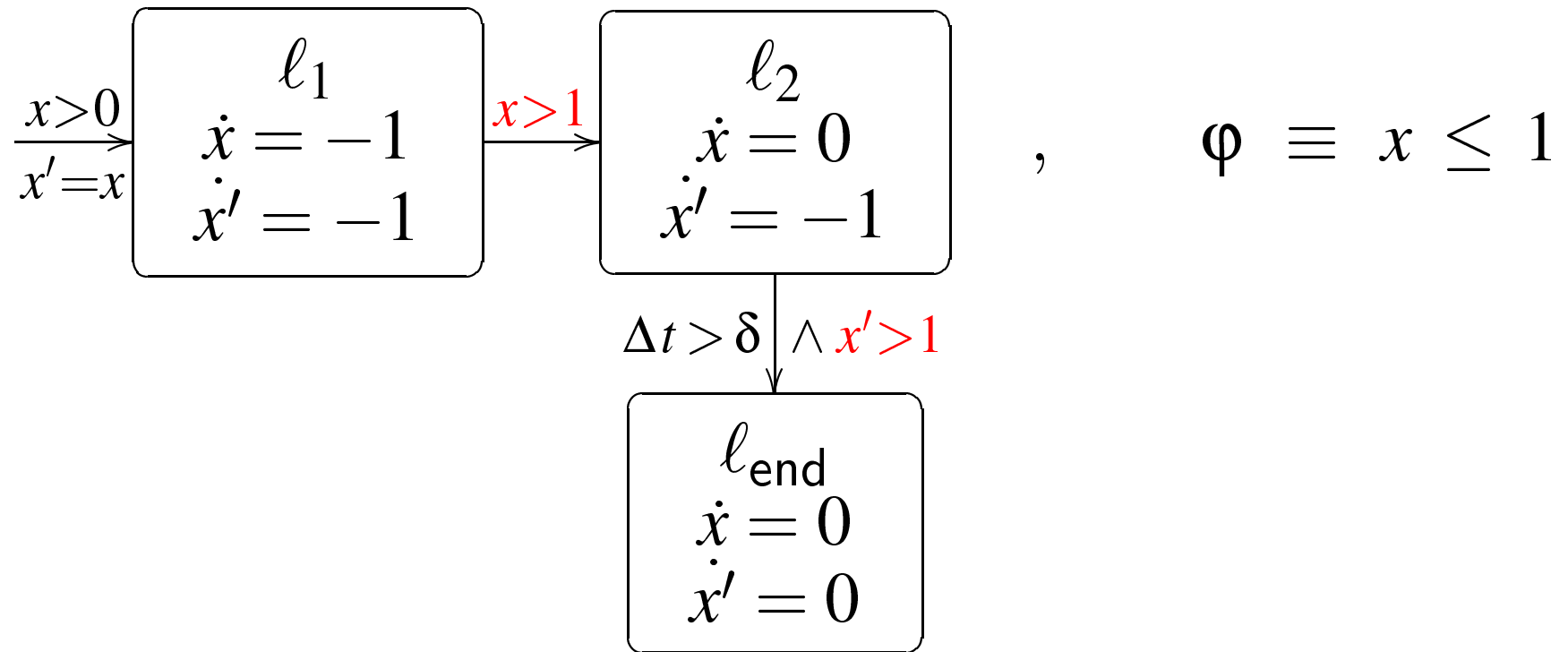
System Transformation



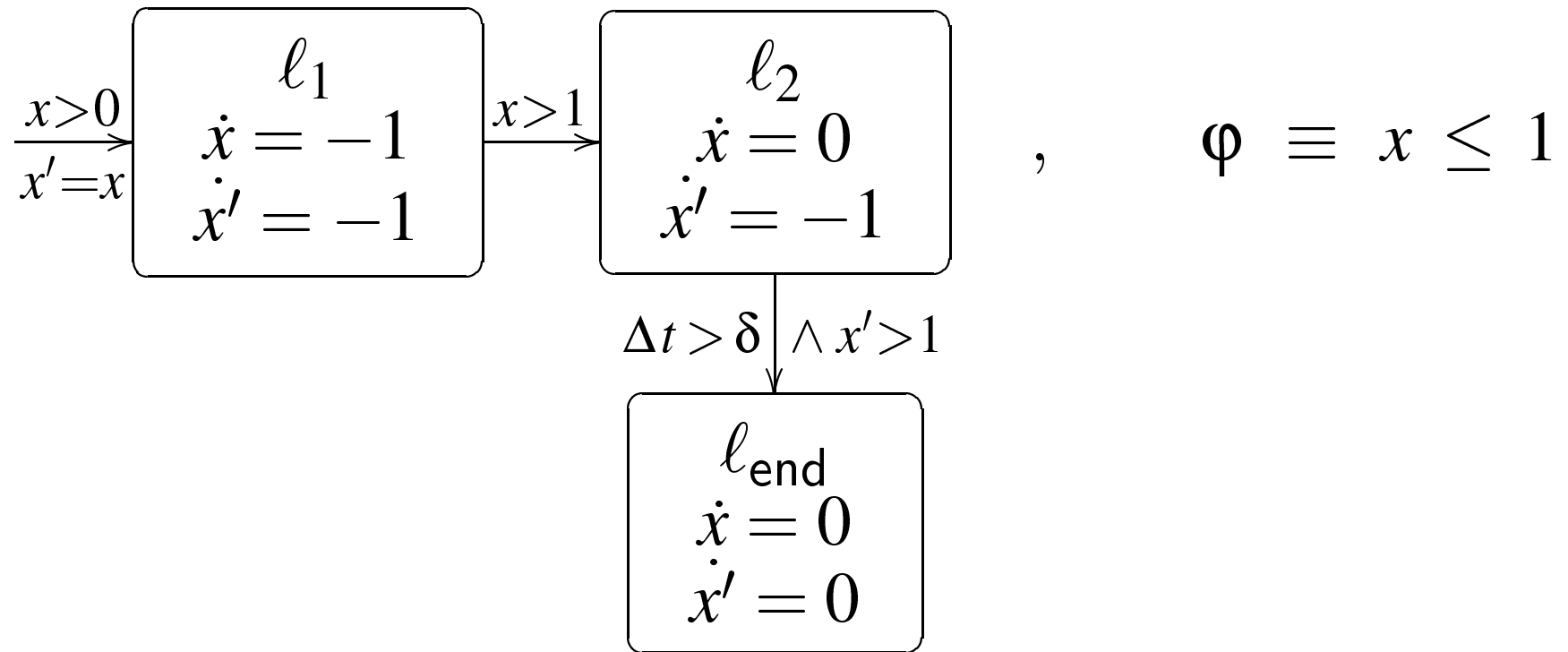
System Transformation



System Transformation



System Transformation



Output: $x > 1 \wedge x - x' \geq \delta$



Finiteness Test

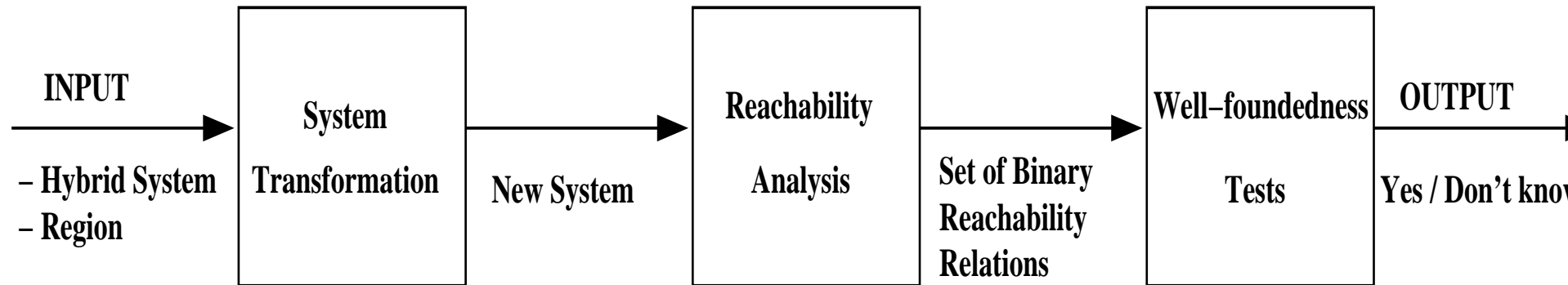
Check whether all binary relations r are **well-founded**.

Well-foundedness:

There is no infinite sequence of states s_0, s_1, s_2, \dots
such that each pair of consecutive states (s_i, s_{i+1})
satisfies the relation r .



Algorithm



- Reachability analysis: PHAVer
- Well-foundedness test: RankFinder



Correctness of the algorithm

- (1) Region stability \rightarrow finiteness test for snapshot sequences
- (2) Correctness of system transformation
- (3) Well-foundedness of each relation in the output of the reachability tool
 \Rightarrow finiteness of all snapshot sequences.



Correctness of the *l* orithm

Each relation in the output of the reachability tool
is well-founded
 \Rightarrow finiteness of all snapshot sequences.



Correctness of the algorithm

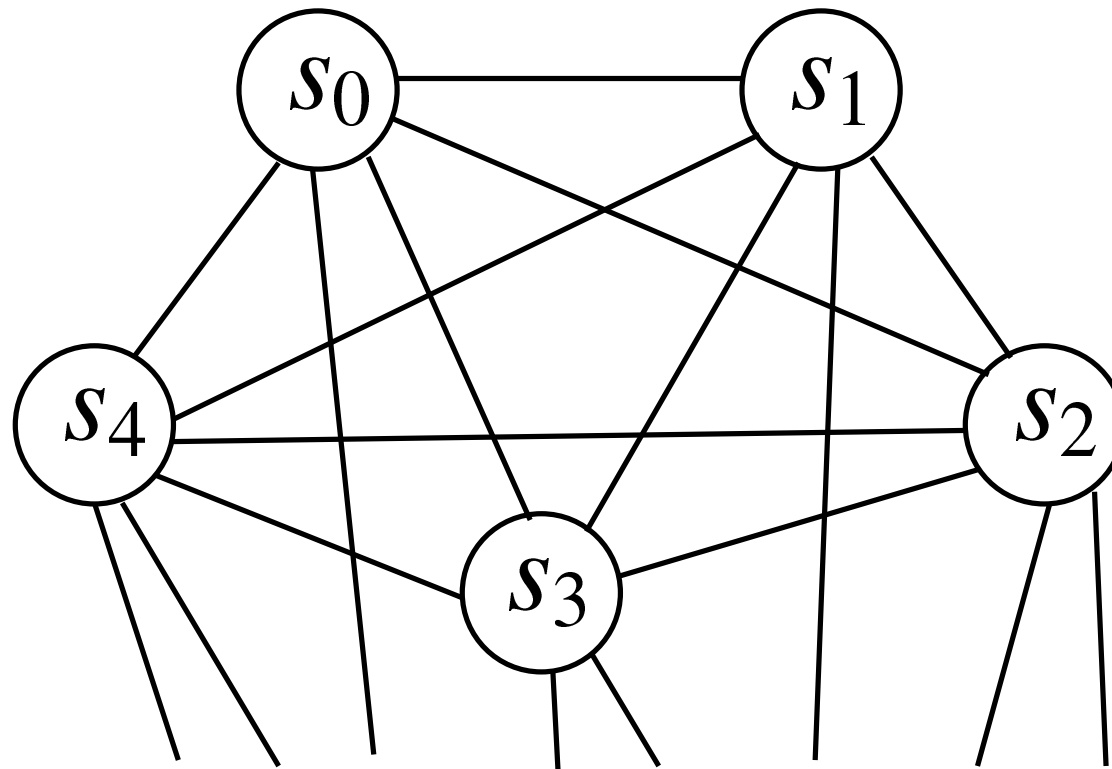
Each relation in the output of the reachability tool
is well-founded

\Rightarrow finiteness of all snapshot sequences.

- Assume $R = \{r_1, r_2, \dots, r_n\}$, all r_i well-founded
- But s_0, s_1, s_2, \dots infinite snapshot sequence

Correctness of the *1* orithm

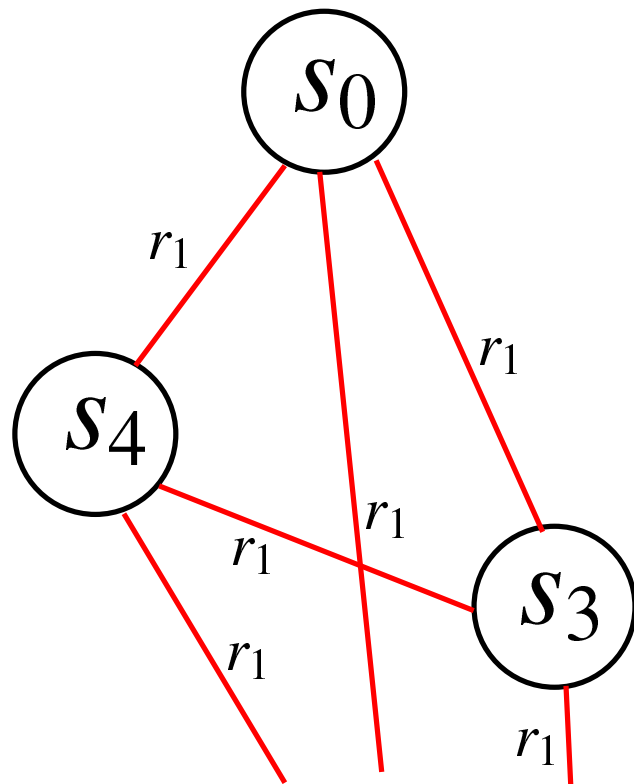
- Assume $R = \{r_1, r_2, \dots, r_n\}$, all r_i well-founded
- But s_0, s_1, s_2, \dots infinite snapshot sequence





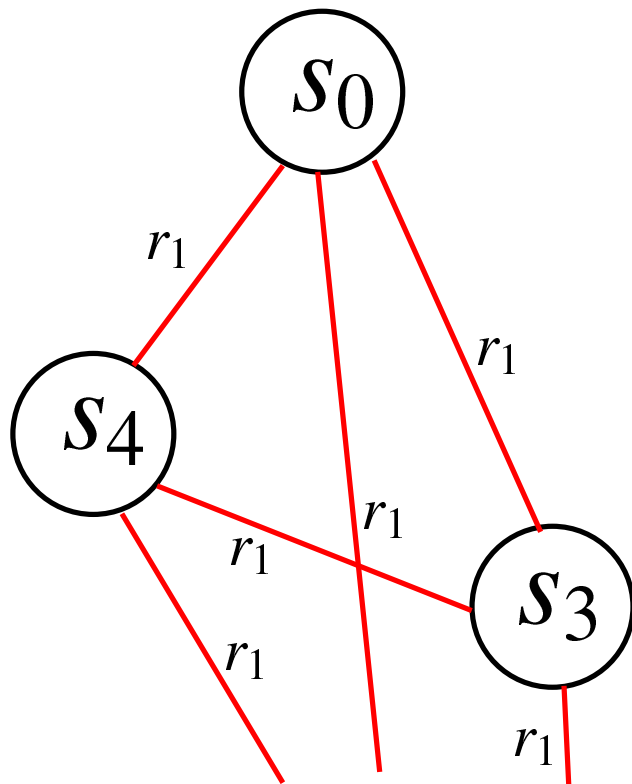
Correctness of the ϵ algorithm

RAMSEY'S THEOREM: exists infinite complete subgraph s.t. all edges have the same color.



Correctness of the 1 orithm

RAMSEY'S THEOREM: exists infinite complete subgraph s.t. all edges have the same color.



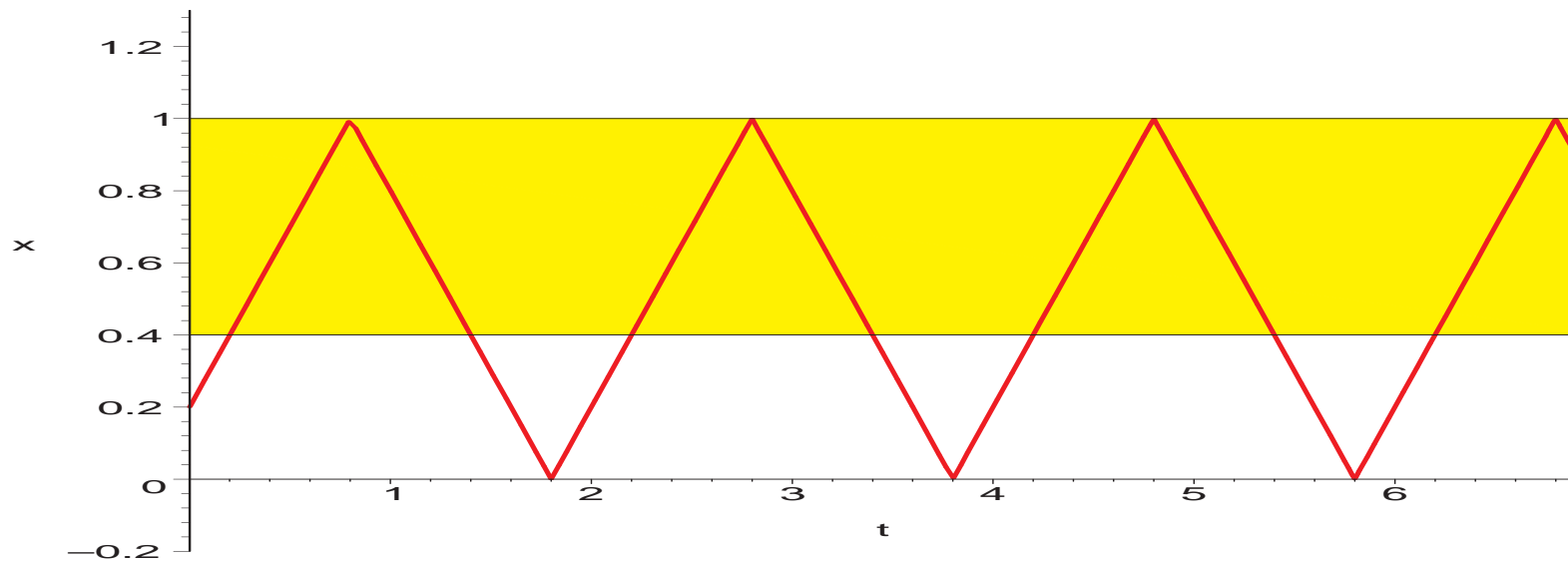
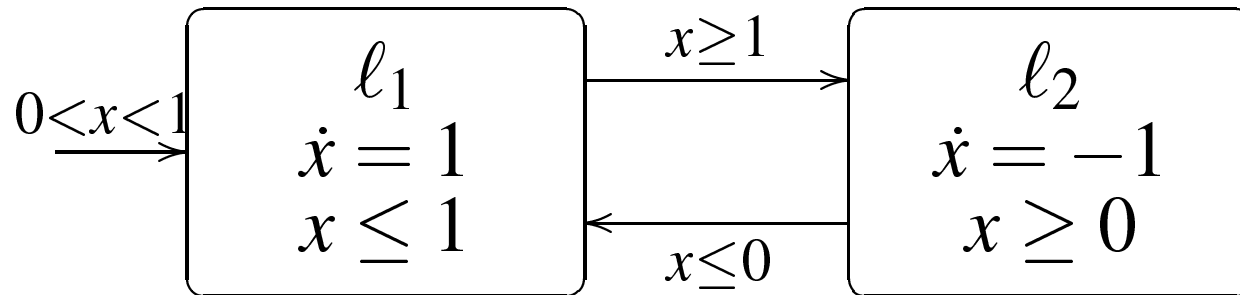
- subsequence s_0, s_3, s_4, \dots infinite
- relation r_1 not well-founded
→ CONTRADICTION



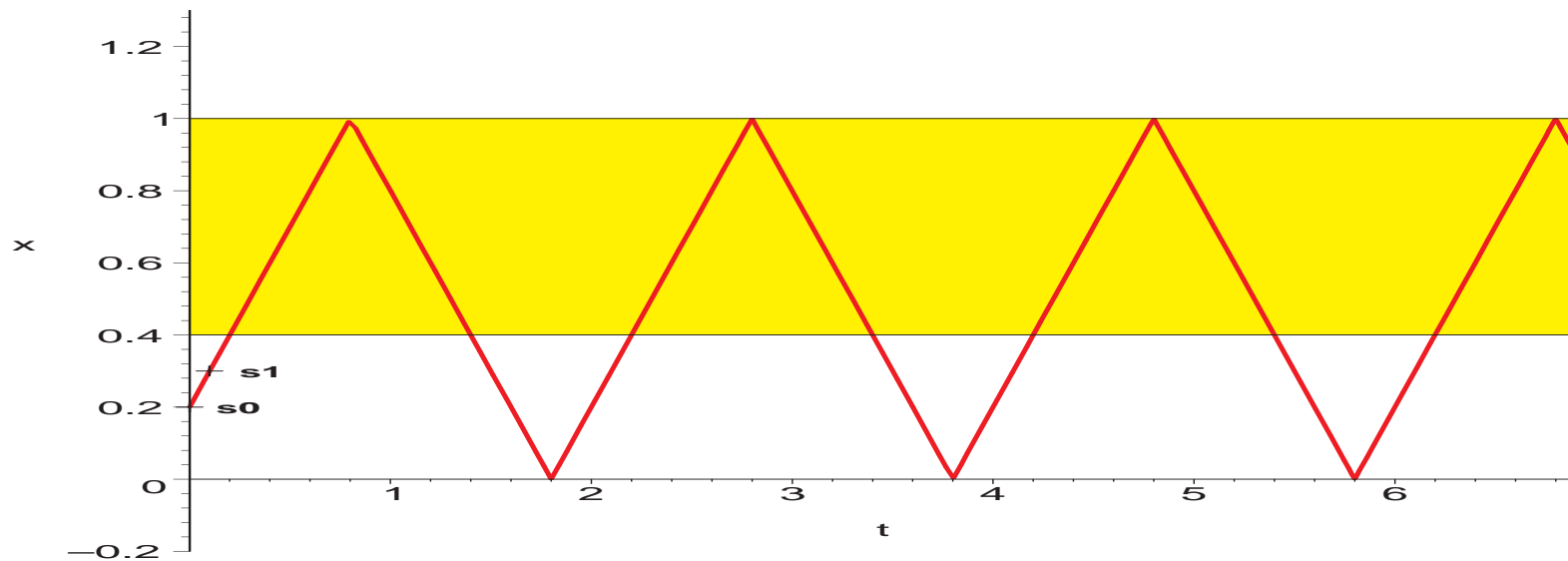
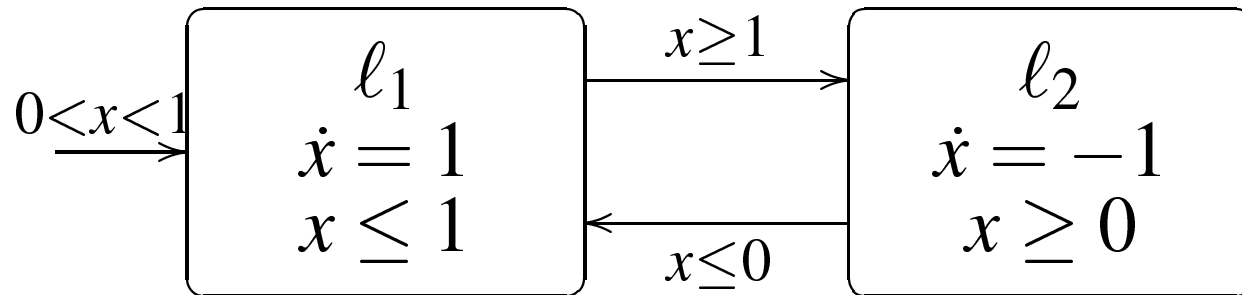
Proof Rule for Hybrid Systems

- So far: $\dot{x} = ax + b$, $a, b \in \mathbb{R}$
- Condition 1: all snapshot sequences on the **monotonic flow** must be finite.
→ Region stability for linear dynamical systems.
- Now: hybrid systems with more than one location.

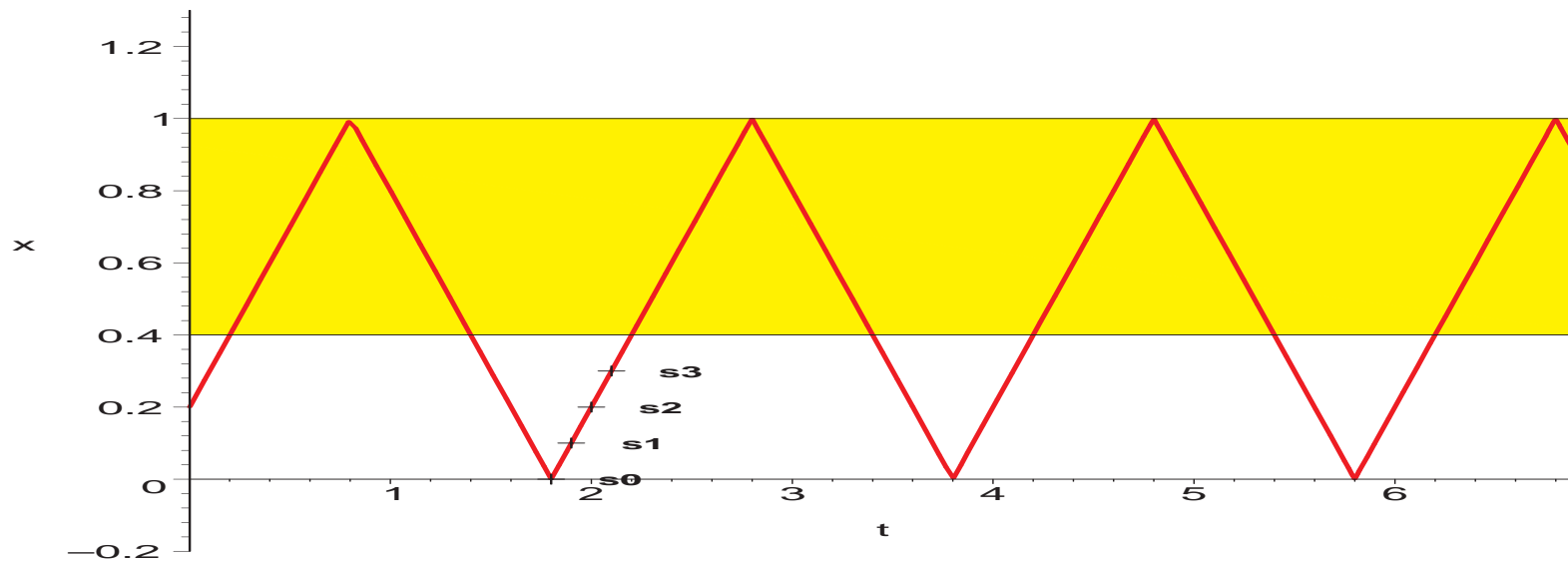
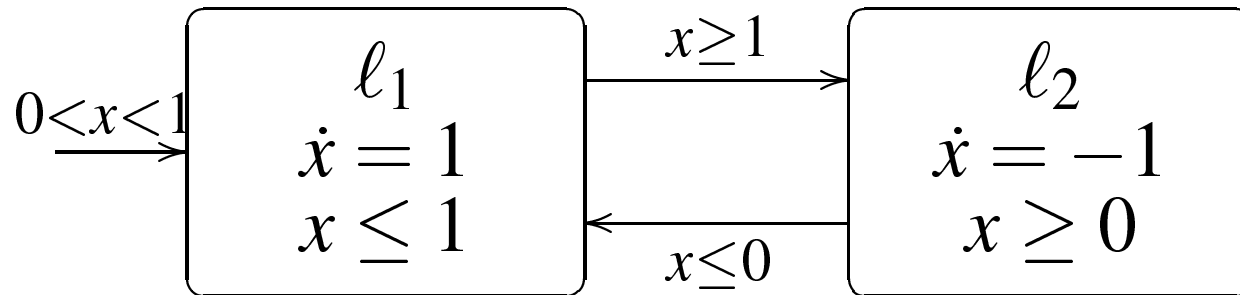
Systems with Several Locations



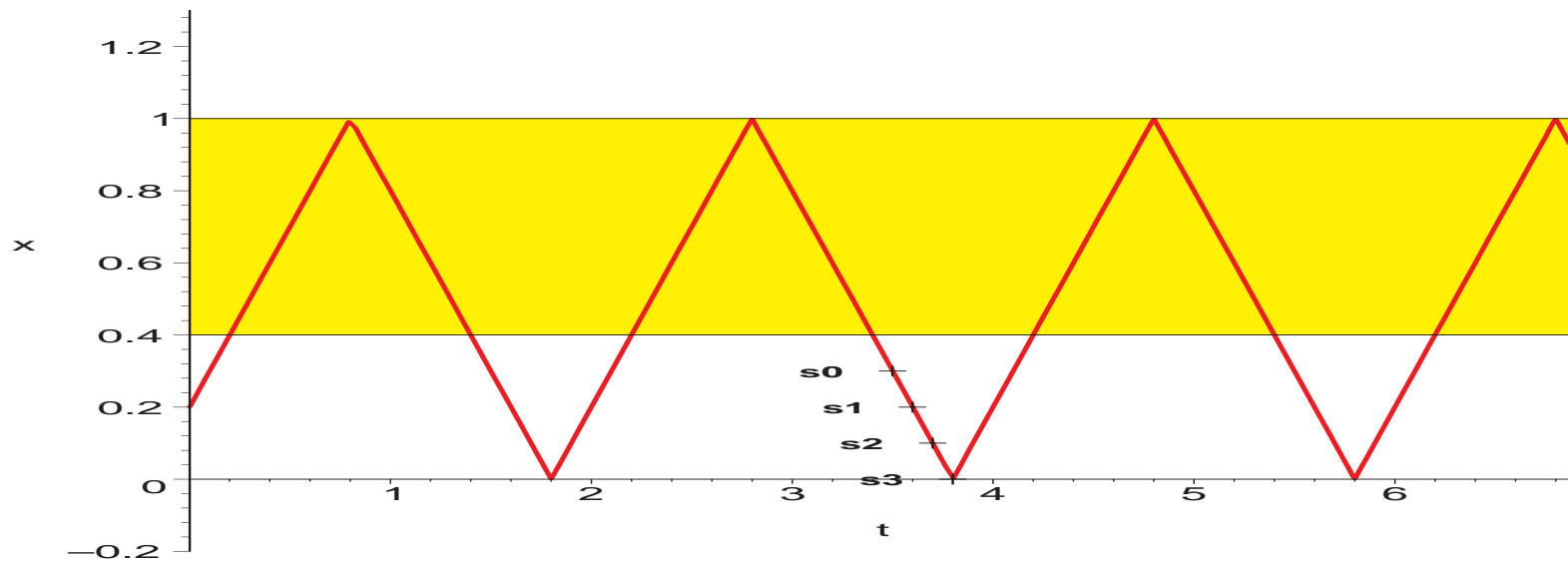
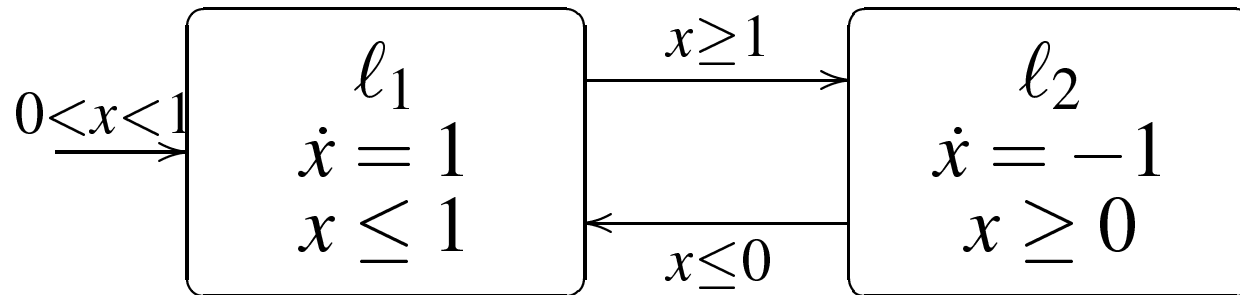
Systems with Several Locations



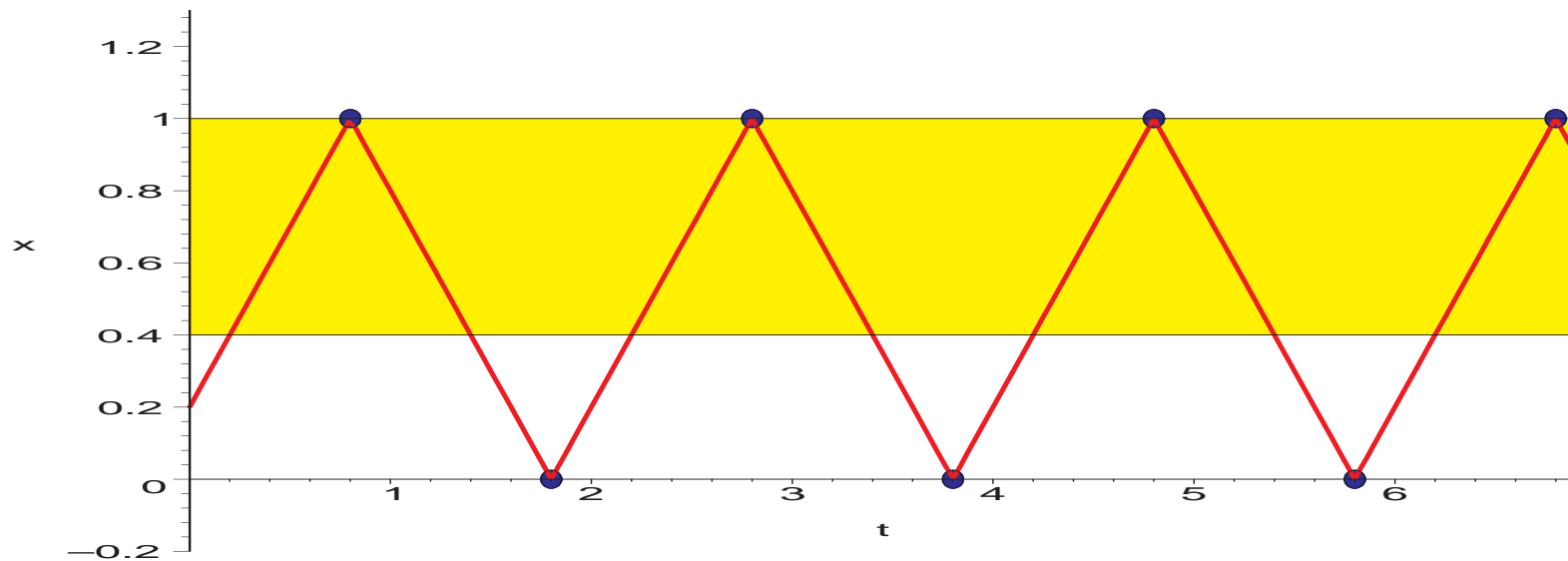
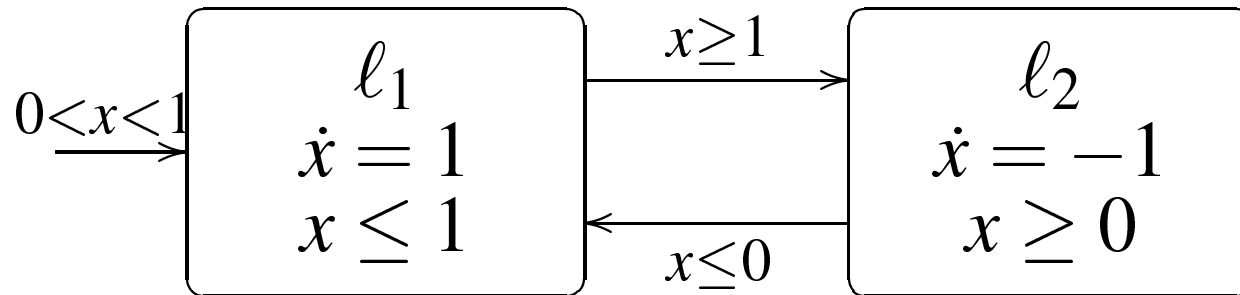
Systems with Several Locations



Systems with Several Locations



Systems with Several Locations



entry-points



Condition 2

Condition 1: all snapshot sequences on all monotonic flows must be finite.

Condition 2: all snapshot sequences of **entry-points** must be finite.

→ Region stability for hybrid systems with more than one location and linear flows.



Systems with non-monotonic flows

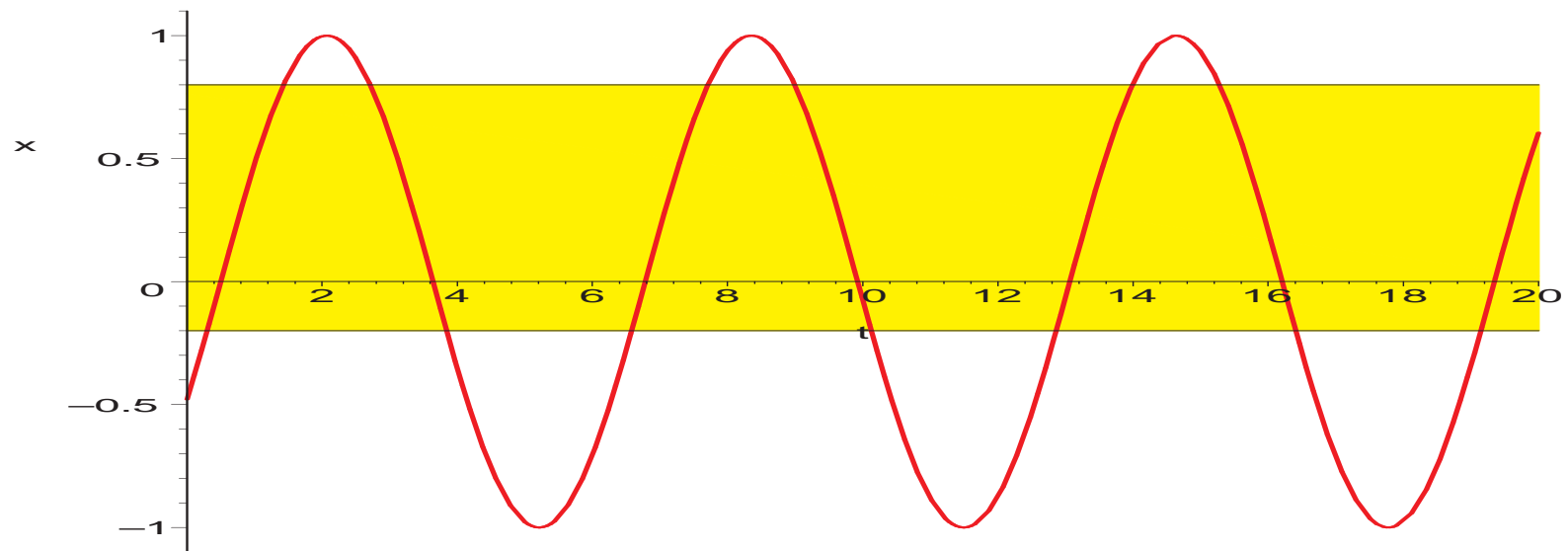
Non-monotonic flows, e.g.

$$\dot{x} = \cos(t) \quad , \quad \varphi \equiv x \in [-0.2, 0.8]$$

Systems with non-monotonic flows

Non-monotonic flows, e.g.

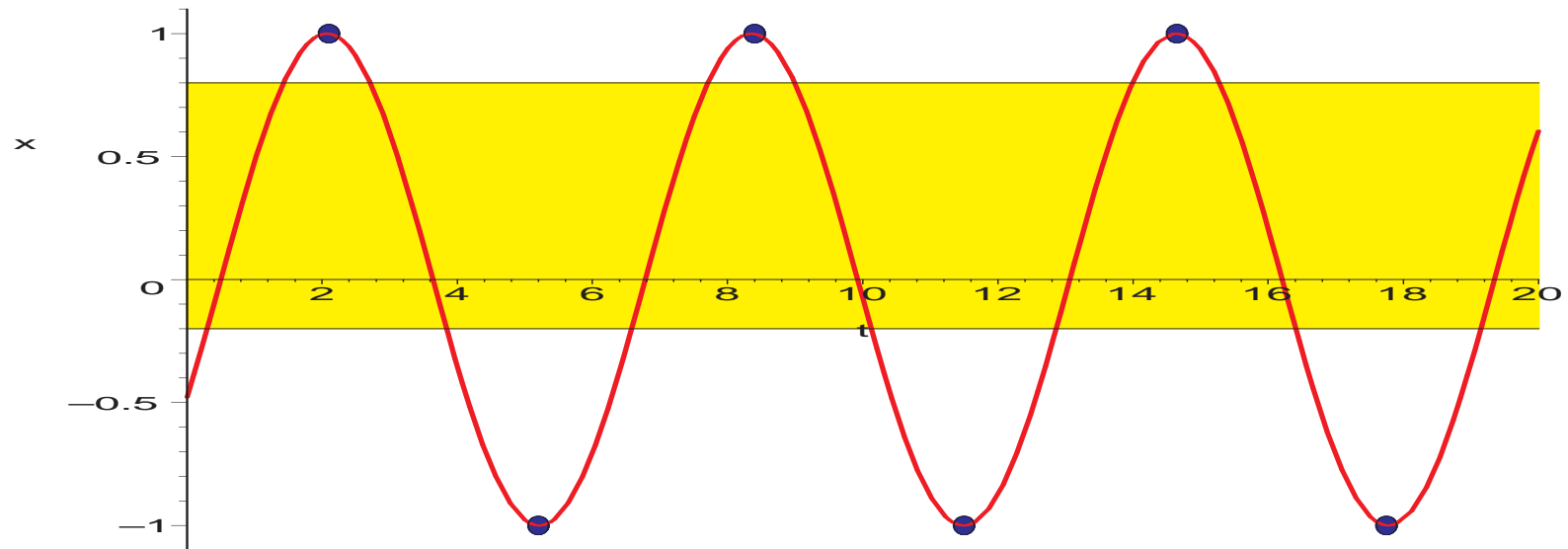
$$\dot{x} = \cos(t) \quad , \quad \varphi \equiv x \in [-0.2, 0.8]$$



Systems with non-monotonic flows

Non-monotonic flows, e.g.

$$\dot{x} = \cos(t) \quad , \quad \varphi \equiv x \in [-0.2, 0.8]$$



extremal-points



Condition 3

Condition 1: all snapshot sequences on the monotonic flow must be finite.

Condition 2: all snapshot sequences of entry-points must be finite.

Condition 3: all snapshot sequences of **extremal-points** must be finite.

→ Region stability for arbitrary hybrid systems.



Implementation

- very similar to implementation for linear dynamical systems
- compute **three kinds** of snapshot sequences
→ three system transformations
- finiteness test for each kind separately



Extensions

1. Region stability for n-dimensional systems

$$x = (x_1, \dots, x_n)^T, \quad \varphi \equiv x_i \in [x_i^{\min}, x_i^{\max}]$$

→ monotonicity / extremal-points in respect of x_i



Extensions

1. Region stability for n-dimensional systems

$$x = (x_1, \dots, x_n)^T, \quad \varphi \equiv x_i \in [x_i^{\min}, x_i^{\max}]$$

→ monotonicity / extremal-points in respect of x_i

2. Region stability w.r.t. **box regions**:

$$\varphi \equiv (x_1, \dots, x_n) \in [x_1^{\min}, x_1^{\max}] \times \dots \times [x_n^{\min}, x_n^{\max}]$$

$$\varphi_i \equiv x_i \in [x_i^{\min}, x_i^{\max}], \quad i = 1 \dots n$$

→ stability w.r.t. each interval region φ_i separately



Benchmarks

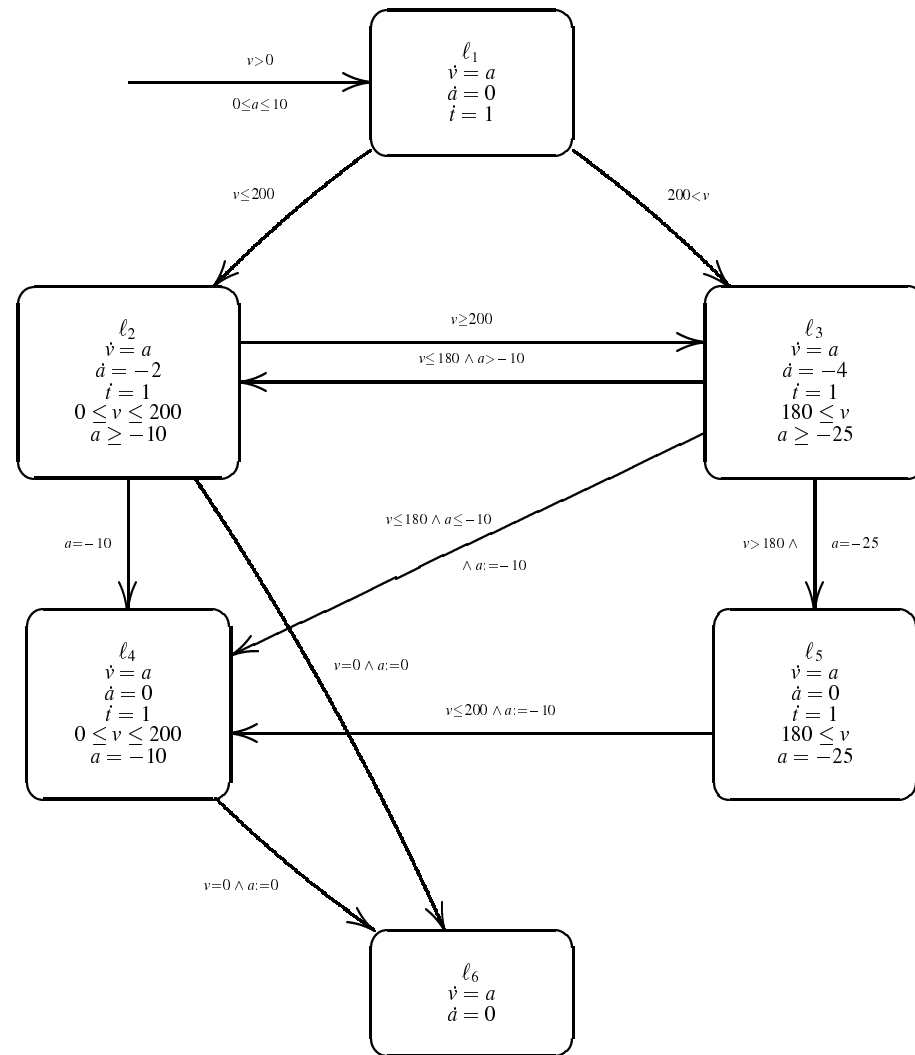
System	# Var's	# Loc's	Run Time
Freezer	2	1	0.191s
Heater	3	2	0.490s
Fragile heater	3	2	1.920s
Bouncing ball	3	2	4.209s
Pendulum	2	2	0.264s
Exothermic reaction	2	3	0.428s
One tank system	3	3	1.813s
Two tank system	3	4	16.545s
Distance controller	3	4	1.186s
Train brake	3	6	2.589s



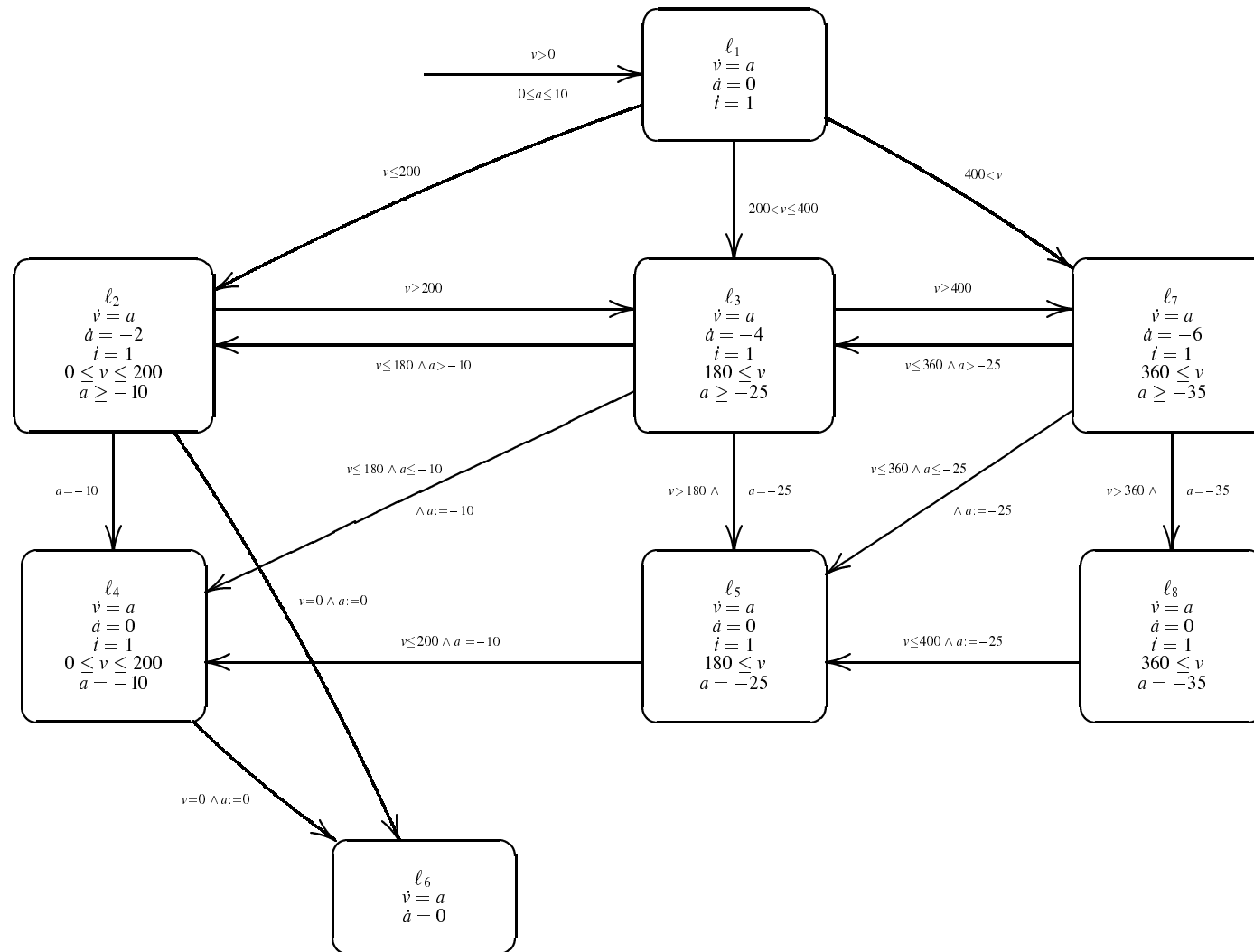
Scalability

- How does the method scale with the number of locations?
- How does it scale with the number of variables?

Scalability – # Locations



Scalability – # Locations





Scalability – # Locations

# Var's	# Loc's	Run Time
3	6	2.589s
3	8	4.652s
3	10	6.646s
3	12	9.783s
3	14	12.739s

→ linear in # locations



Scalability – # Variables

# Var's	# Loc's	Run Time
3	6	5.307s
4	8	33.065s
5	10	203.646s
6	12	out of time

→ exponential in # variables



Conclusion

- characterisation of region stability as finiteness of snapshot sequences
- implementation: reachability analysis + well-foundedness test
- fully automated

FUTURE WORK

- **scaling up**
- more general regions