

Modeling and Analysis of Hybrid Systems

Algorithmic analysis for linear hybrid systems

Prof. Dr. Erika Ábrahám

Informatik 2 - Theory of Hybrid Systems
RWTH Aachen University

SS 2015



Alur et al.: The algorithmic analysis of hybrid systems

Theoretical Computer Science, 138(1):3–34, 1995

Linear terms, constraints and formulas

- A **linear term** e over a set Var of variables is of the form

$$e ::= c \mid c \cdot x \mid e + e$$

where $x \in Var$ is a variable and c stands for an integer (rational) constant.

Example: $x_1 + 2x_2 + 5x_3$ is a linear term over $Var = \{x_1, x_2, x_3\}$.

- A **linear constraint** t over Var is an (in)equality

$$t ::= e \sim 0$$

with $\sim \in \{>, \geq, =, \leq, <\}$ and e a linear term over Var .

Example: $x_1 + 2x_2 + (-2) \geq 0$ is a linear constraint over $Var = \{x_1, x_2\}$. We sometimes deviate from this normal form and write, e.g., $x_1 + 2x_2 \geq 2$.

Linear terms, constraints and formulas

- A **conjunctive linear formula** φ over Var is defined by the following grammar:

$$\varphi ::= t \mid \varphi \wedge \varphi \mid \exists x. \varphi$$

with t a linear constraint over Var and $x \in Var$ a variable. Let Φ_X be the set of all conjunctive linear formulas with free (non-quantified) variables from $X \subseteq Var$.

Example: $\exists t. \exists x_1. x_1 \geq 0 \wedge x'_1 = x_1 + 2t \wedge t \geq 0$ is a conjunctive linear formula over $\{x_1, x'_1, t\}$, with **free** (non-quantified) variables $\{x'_1\}$.

- A **region** over Var is a pair $(l, \varphi) \in Loc \times \Phi_{Var}$ of a location and a conjunctive linear formula with free variables from Var .

Example: $(l, \exists t. \exists x_1. x_1 \geq 0 \wedge x'_1 = x_1 + 2t \wedge t \geq 0)$ is a region over $\{x'_1\}$.

- We will use the **intersection** of two sets $P^i, P^j \subseteq Loc \times \Phi_{Var}$ of regions, defined as

$$P^1 \hat{\cap} P^2 = \{(l, \varphi_1 \wedge \varphi_2) \mid (l, \varphi_1) \in P^1, (l, \varphi_2) \in P^2\} .$$

- Assume a set Var of variables, a linear formula $\varphi \in \Phi_{Var}$ over Var , a variable $x \in Var$, and a linear term e over Var . The **substitution** $\varphi[e/x]$ replaces each **free** occurrence of x in φ by e .

Example: $(x + 2y \leq 0)[5/y] = x + 2 \cdot 5 \leq 0$

Example: $(\exists y. x + 2y \leq 0)[5/y] = x + 2y \leq 0$

- We also write $\varphi[e_1, \dots, e_n/x_1, \dots, x_n]$ instead of $\varphi[e_1/x_1] \dots [e_n/x_n]$.
- For $Var = \{x_1, \dots, x_n\}$ we will also use a primed variable set $Var' = \{x'_1, \dots, x'_n\}$ and write short $\varphi[Var'/Var]$ for $\varphi[x'_1/x_1] \dots [x'_n/x_n]$.

The **semantics** of linear terms, constraints and formulas over $Var = \{x_1, \dots, x_n\}$ in the context of a valuation $\nu \in V_{Var}$ (i.e., $\nu : Var \rightarrow \mathbb{R}$) is as usual (we use the same notation for the syntax and the semantics of constants and operators):

$$\begin{aligned}\nu(c) &\equiv c \\ \nu(c \cdot x) &\equiv c \cdot \nu(x) \\ \nu(e_1 + e_2) &\equiv \nu(e_1) + \nu(e_2) \\ \nu(e \sim 0) &\equiv \nu(e) \sim 0 \\ \nu(\varphi_1 \wedge \varphi_2) &\equiv \nu(\varphi_1) \text{ and } \nu(\varphi_2) \\ \nu(\exists x. \varphi) &\equiv \text{exists } v \in \mathbb{R} \text{ such that } \nu(\varphi[v/x]) \text{ holds}\end{aligned}$$

Linear terms, constraints and formulas

The **solution set** $Sat(\varphi)$ of a linear formula φ over Var is the set of all valuations $\nu \in V_{Var}$ that make φ true:

$$Sat(\varphi) = \{\nu \in V_{Var} \mid \nu(\varphi) \text{ holds}\}$$

The **solution set** $Sat((l, \varphi))$ of a region $(l, \varphi) \in Loc \times \Phi_{Var}$ over Loc and Var is

$$Sat((l, \varphi)) = \{(l, \nu) \in Loc \times V_{Var} \mid \nu(\varphi) \text{ holds}\}$$

Two region sets $P_1, P_2 \subseteq Loc \times \Phi_{Var}$ are **equivalent**, written $P_1 \hat{=} P_2$, iff

$$\bigcup_{R \in P_1} Sat(R) = \bigcup_{R \in P_2} Sat(R) .$$

We define similarly the inclusion $P_1 \hat{\subseteq} P_2$ iff

$$\bigcup_{R \in P_1} Sat(R) \subseteq \bigcup_{R \in P_2} Sat(R) .$$

Linear hybrid automata I

A **linear hybrid automaton** is a hybrid automaton

$\mathcal{H} = (Loc, Var, Lab, Edge, Act, Inv, Init)$ which can be represented by a tuple

$H = (Loc, Var, Lab, Edge, Act, Inv, Init)$ satisfying the following:

- The finite set **Edge** $\subseteq Loc \times \Phi_{Var} \times \Phi_{Var \cup Var'} \times Loc$ defines the set of edges $Edge = \{(l, \mu_e, l') \mid e = (l, Guard_e, Reset_e, l') \in Edge\}$, where the transition relation μ_e is given as

$$\mu_e = \{(\nu, \nu') \in V_{Var} \times V_{Var} \mid \nu \in Sat(Guard_e) \text{ and } (\nu \oplus \nu') \in Sat(Reset_e)\}$$

with $\nu \oplus \nu' \in V_{Var \cup Var'}$ such that $(\nu \oplus \nu')(x) = \nu(x)$ for $x \in Var$ and $(\nu \oplus \nu')(x') = \nu'(x')$ for $x' \in Var'$.

- The set **Act** $\subseteq Loc \times \Phi_{Var}$ contains for each location $l \in Loc$ exactly one region $(l, Act_l) \in Act$ whose second component is of the form $Act_l = \bigwedge_{i=1}^n x_i + k_{l,i}^{lower} t \leq x'_i \wedge x'_i \leq x_i + k_{l,i}^{upper} t$, defining the activities

$$Act(l) = \{f : \mathbb{R}_{\geq 0} \rightarrow V_{Var} \mid \dot{f}_{x_i} \in [k_{l,i}^{lower}, k_{l,i}^{upper}] \text{ for all } i \in \{1, \dots, n\}\},$$

where $f_{x_i} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ with $f_{x_i}(t) = f(t)(x_i)$ for all $t \in \mathbb{R}_{\geq 0}$.

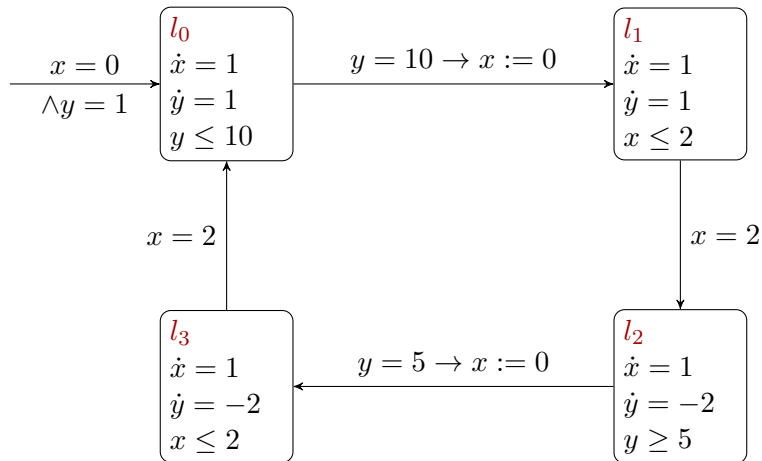
- The set $\mathbf{Inv} \subseteq \mathit{Loc} \times \Phi_{\mathit{Var}}$ contains for each location $l \in \mathit{Loc}$ exactly one region $(l, \mathbf{Inv}_l) \in \mathbf{Inv}$ such that

$$\mathit{Inv}(l) = \mathit{Sat}(\mathbf{Inv}_l) .$$

- $\mathbf{Init} \subseteq \mathit{Loc} \times \Phi_{\mathit{Var}}$ is a finite set of regions specifying the initial states by

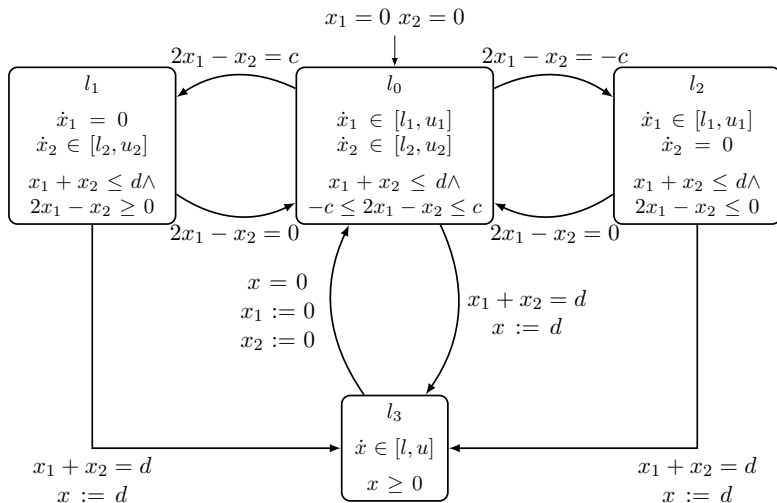
$$\mathbf{Init} = \bigcup_{(l, \varphi) \in \mathbf{Init}} \{(l, \nu) \mid \nu \in \mathit{Sat}(\varphi)\} .$$

Water-level monitor



Mixer of fluids

$$0 < l_1 < u_1, 0 < l_2 < u_2, l \leq u \leq 0, d > 0, c > 0$$



Reminder: Semantics of hybrid automata

$$(l, a, \mu, l') \in Edge \quad (\nu, \nu') \in \mu \quad \nu' \in Inv(l')$$

Rule Discrete

$$(l, \nu) \xrightarrow{a} (l', \nu')$$

$$f \in Act(l) \quad f(0) = \nu \quad f(t) = \nu'$$
$$t \geq 0 \quad \forall 0 \leq t' \leq t. f(t') \in Inv(l)$$

Rule Time

$$(l, \nu) \xrightarrow{t} (l, \nu')$$

Forward analysis

- Given a set of initial states $Init \subseteq \Sigma$, we want to compute the set of all states which are reachable from $Init$:

$$Reach^+(Init) = \{\sigma' \in \Sigma \mid \exists \sigma \in Init. \sigma \rightarrow^* \sigma'\} .$$

- Given a **set of initial states** $Init \subseteq \Sigma$, we want to compute the set of all states which are **reachable** from $Init$:

$$Reach^+(Init) = \{\sigma' \in \Sigma \mid \exists \sigma \in Init. \sigma \rightarrow^* \sigma'\} .$$

- More specifically, we want to check whether the reachable region intersects with a set of **bad (unsafe)** states.

Forward reachability analysis

```
method forward_reach(  
    hybrid_automaton_representation H = (Loc, Var, Lab, Edge, Act, Inv, Init),  
    region_set Pbad) {  
    //start from the time successors of initial regions  
    P0 := T+(Init ∧ Inv);  
    if (P0 ∧ Pbad ≠ ∅) return unsafe;  
    i := 0;  
    while Pi ≠ ∅ {  
        //compute Pi+1 := T+(D+(Pi))  
        Pi+1 := ∅;  
        for each (R = (l, φ) ∈ Pi) {  
            for each e = (l, ·, ·, l') ∈ Edge {  
                R' := Tl'+(De+(R));  
                if ({R'} ∧ Pbad ≠ ∅) return unsafe;  
                else if (not {R'} ⊆ ⋃j=0i Pj)  
                    Pi+1 := Pi+1 ∪ {R'};  
            }  
        }  
        i := i+1;  
    }  
    return safe;  
}
```


- We define the **forward time closure** $\mathcal{T}_l^+(\varphi)$ of a **formula** $\varphi \in \Phi_{Var}$ at $l \in Loc$ as

$$\mathcal{T}_l^+(\varphi) = \exists x_{pre}. \exists t. t \geq 0 \wedge \varphi[x_{pre}/x] \wedge \text{Act}_l[x_{pre}, x/x, x'] \wedge \text{Inv}_l$$

- **Region** $R = (l, \varphi) \in Loc \times \Phi_{Var}$:

$$\mathcal{T}_l^+(R) = (l, \mathcal{T}_l^+(\varphi))$$

- **Set of regions** $P \subseteq Loc \times \Phi_{Var}$:

$$\mathcal{T}^+(P) = \{\mathcal{T}_l^+(R) \mid R = (l, \varphi) \in P\}$$

One-step reachability under discrete steps

- We define the **postcondition** $\mathcal{D}_e^+(\varphi)$ of a **formula** $\varphi \in \Phi_{Var}$ with respect to an edge $e = (l, \text{Guard}_e, \text{Reset}_e, l')$ as

$$\mathcal{D}_e^+(\varphi) = \exists x_{pre}. \varphi[x_{pre}/x] \wedge \text{Guard}_e[x_{pre}/x] \wedge \text{Reset}_e[x_{pre}, x/x, x'] \wedge \text{Inv}_{l'}$$

- **Region** $R = (l, \varphi) \in Loc \times \Phi_{Var}$:

$$\mathcal{D}_e^+(R) = (l', \mathcal{D}_e^+(\varphi))$$

- **Set of regions** $P \subseteq Loc \times \Phi_{Var}$:

$$\mathcal{D}^+(P) = \{\mathcal{D}_e^+(R) \mid R = (l, \varphi) \in P, e = (l, \text{Guard}_e, \text{Reset}_e, l') \in \text{Edge}\}$$

Backward analysis

- Given a set of target states $B \subseteq \Sigma$, we want to compute the set of all states from which a state in B is reachable:

$$Reach^-(B) = \{\sigma \in \Sigma \mid \exists \sigma' \in B. \sigma \rightarrow^* \sigma'\} .$$

- Given a **set of target states** $B \subseteq \Sigma$, we want to compute the set of all states from which a state in B is reachable:

$$Reach^-(B) = \{\sigma \in \Sigma \mid \exists \sigma' \in B. \sigma \rightarrow^* \sigma'\} .$$

- More specifically, we want to check whether the set of backward reachable states intersects with a set of **initial** states.

Backward reachability analysis

```
method backward_reach() {
  i := 0;
   $P^0 := \{T_l^-(R) \mid R = (l, \varphi) \in P^{bad} \hat{\cap} \text{Inv}\};$  //time predec. of bad reg
  if ( $P^0 \hat{\cap} \text{Init} \neq \emptyset$ ) return unsafe;
  while  $P^i \neq \emptyset$  {
    //compute time predecessors of
    //discrete predecessors of all regions from  $P^i$ 
    for each  $(R = (l, \phi) \in P^i)$  {
      for each  $e = (l', \cdot, \cdot, l) \in \text{Edge}$  {
         $R' := T_{l'}^-(\mathcal{D}_e^-(R));$ 
        if ( $\{R'\} \hat{\cap} \text{Init} \neq \emptyset$ ) return unsafe;
        else if (not  $\{R'\} \hat{\subseteq} \bigcup_{j=0}^i P^j$ )
           $P^{i+1} := P^{i+1} \cup \{R'\};$ 
      }
    }
    i := i+1;
  }
  return safe;
}
```

- We define the **backward time closure** $\mathcal{T}_l^-(\varphi)$ of a **formula** $\varphi \in \Phi_{Var}$ at $l \in Loc$ as

$$\mathcal{T}_l^-(\varphi) = \exists x_{post}. \exists t. t \geq 0 \wedge \varphi[x_{post}/x] \wedge \mathbf{Act}_l[x, x_{post}/x, x'] \wedge \mathbf{Inv}_l .$$

- **Region** $R = (l, \varphi) \in Loc \times \Phi_{Var}$:

$$\mathcal{T}_l^-(R) = (l, \mathcal{T}_l^-(\varphi))$$

- **Set of regions** $P \subseteq Loc \times \Phi_{Var}$:

$$\mathcal{T}^-(P) = \{\mathcal{T}_l^-(R) \mid R = (l, \varphi) \in P\}$$

One-step reachability under discrete steps

- We define the **precondition** $\mathcal{D}_e^-(\varphi)$ of a **formula** $\varphi \in \Phi_{Var}$ with respect to an edge $e = (l, \text{Guard}_e, \text{Reset}_e, l')$ as

$$\mathcal{D}_e^-(\varphi) = \exists x_{post}. \varphi[x_{post}/x] \wedge \text{Guard}_e \wedge \text{Reset}_e[x, x_{post}/x, x'] \wedge \text{Inv}_l.$$

- **Region** $R = (l', \varphi) \in \text{Loc} \times \Phi_{Var}$:

$$\mathcal{D}_e^-(R) = (l, \mathcal{D}_e^-(\varphi))$$

- **Set of regions** $P \subseteq \text{Loc} \times \Phi_{Var}$:

$$\mathcal{D}^-(P) = \{\mathcal{D}_e^-(R) \mid R = (l', \varphi) \in P, e = (l, \text{Guard}_e, \text{Reset}_e, l') \in \text{Edge}\}$$