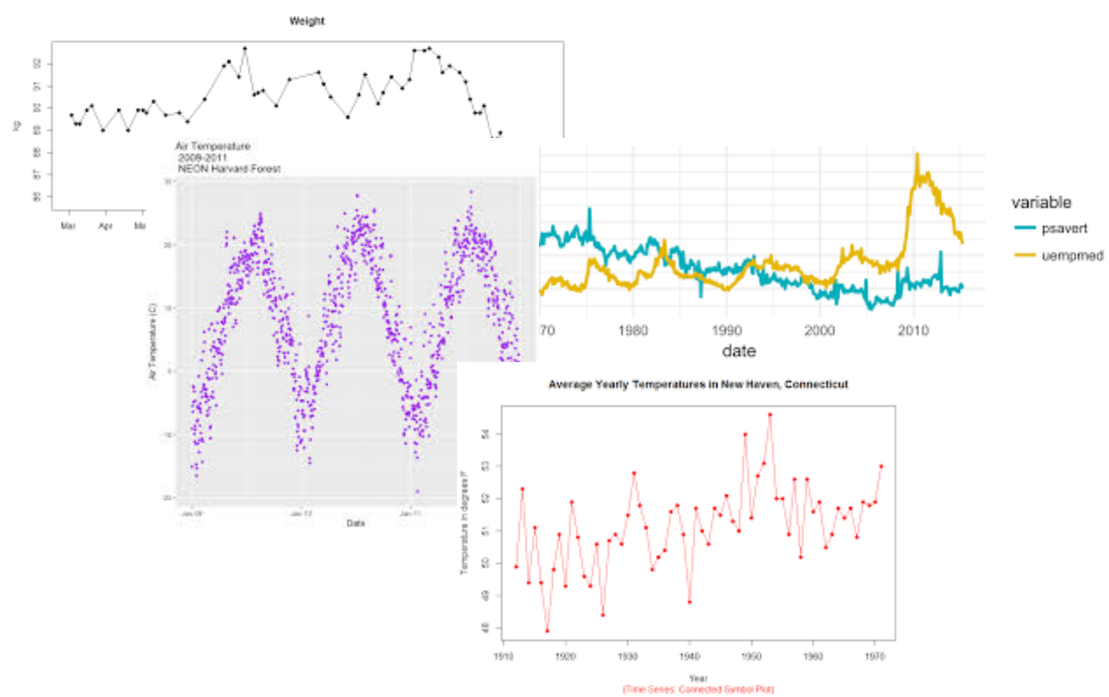
# Synthesis of Linear Hybrid Automata from Experimental Data

**Miriam García Soto**
Co-authored work with Thomas A. Henzinger, Christian Schilling and Luka Zeleznik

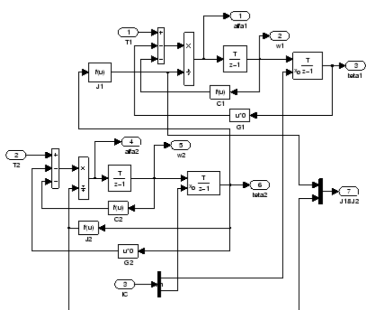
IST Austria - 2019

**Main goal of many sciences is to create a model from a real system**




**Experimental data**

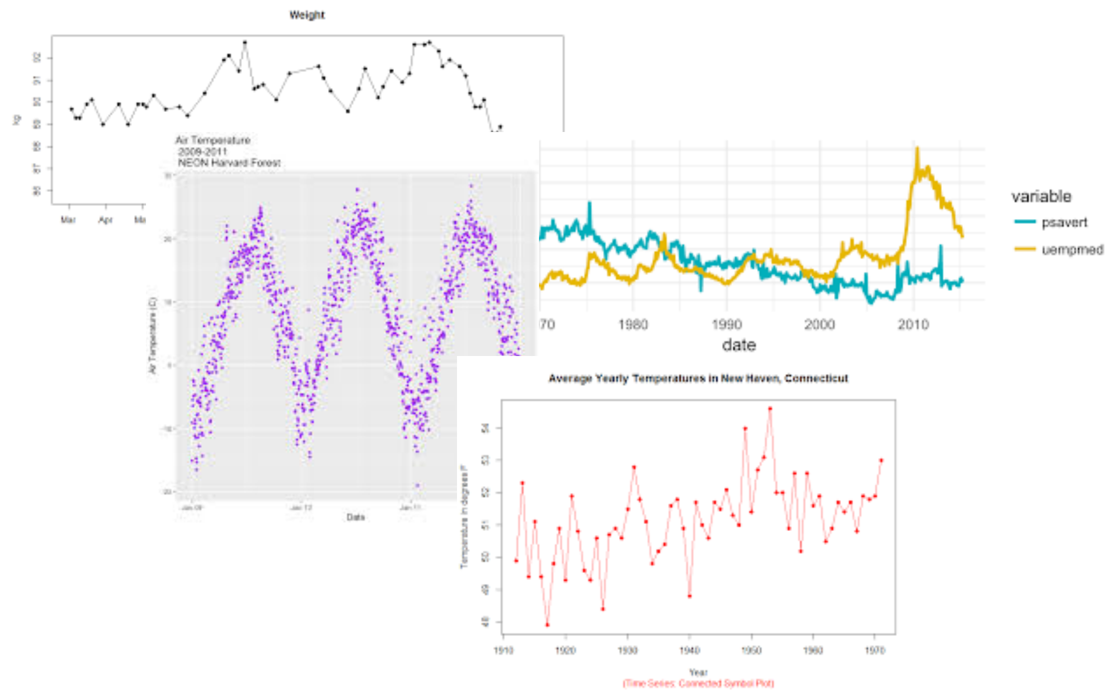
**Analysis**



**Model**

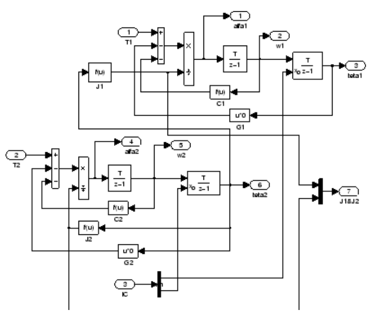
# Motivation

**Main goal of many sciences is to create a model from a real system**




**Experimental data**

**Analysis**



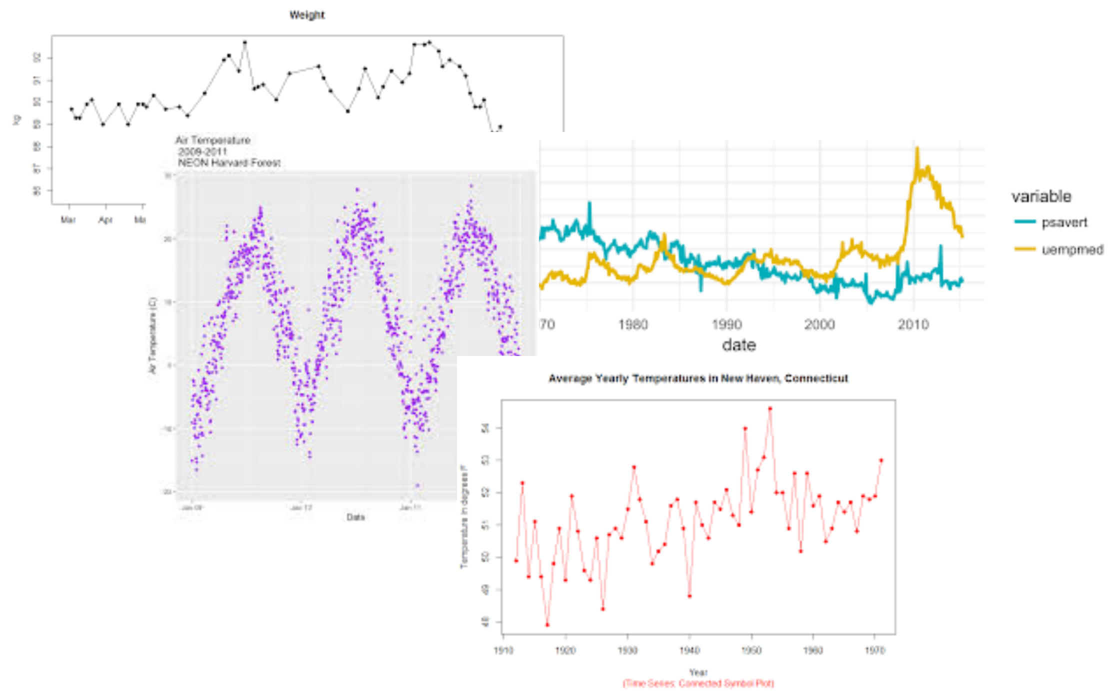
**Model**

**Challenge**

How to automatically create a model?

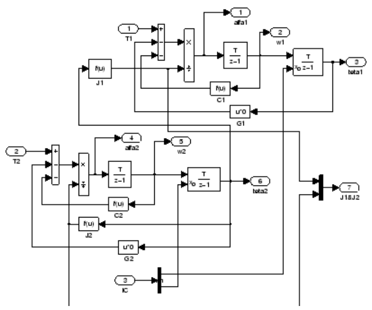
# Motivation

**Main goal of many sciences is to create a model from a real system**




**Experimental data**

**Synthesis**



**Model**

**Challenge**

How to automatically create a model?

# Model: Linear Hybrid Automaton

# Hybrid behavior


- Finite behavioral states
- Behavioral states depending on concentration of food, oxygen, temperature, etc.
- Mixed discrete and continuous behavior

# Hybrid behavior

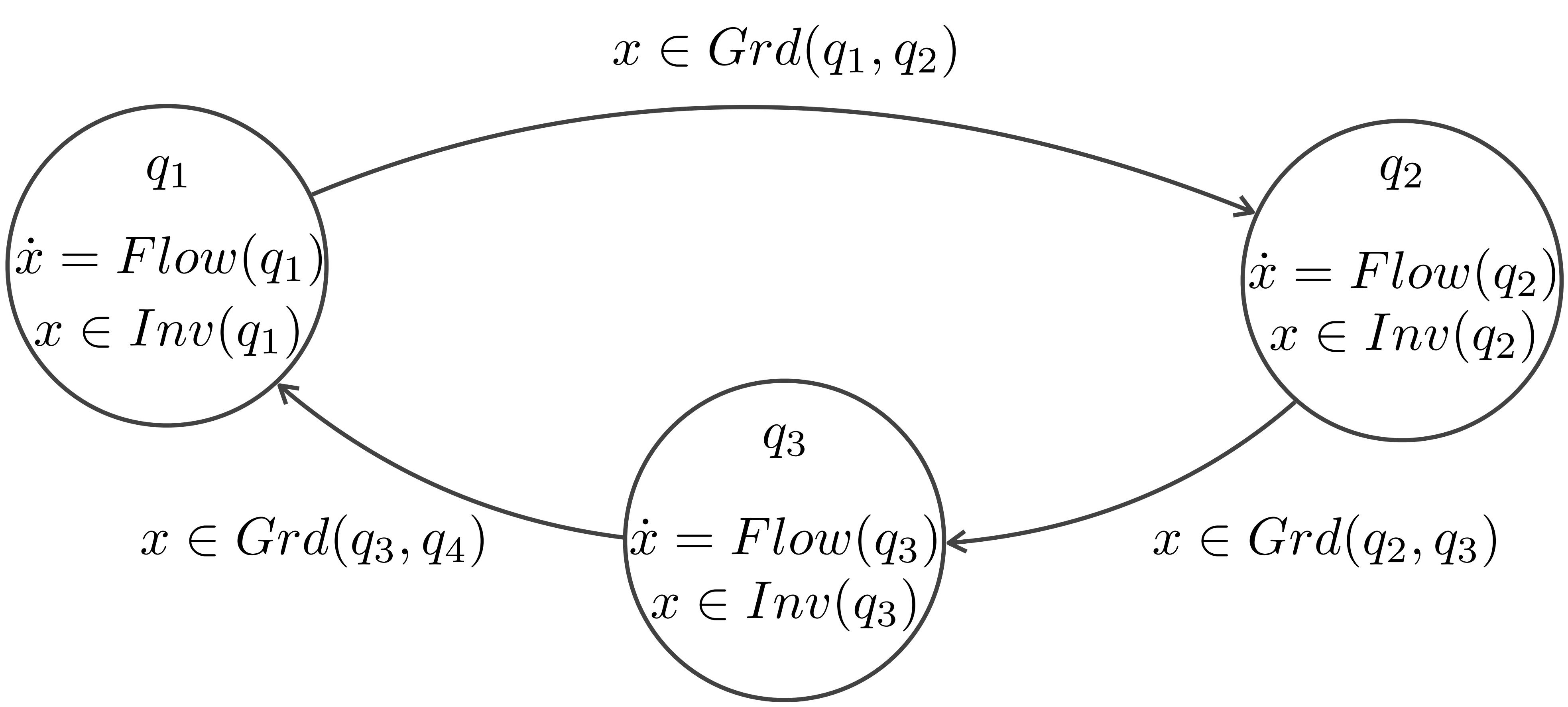

- Finite behavioral states
- Behavioral states depending on concentration of food, oxygen, temperature, etc.
- Mixed discrete and continuous behavior

**Hybrid Systems** capture the mixed **continuous** and **discrete** behaviour.

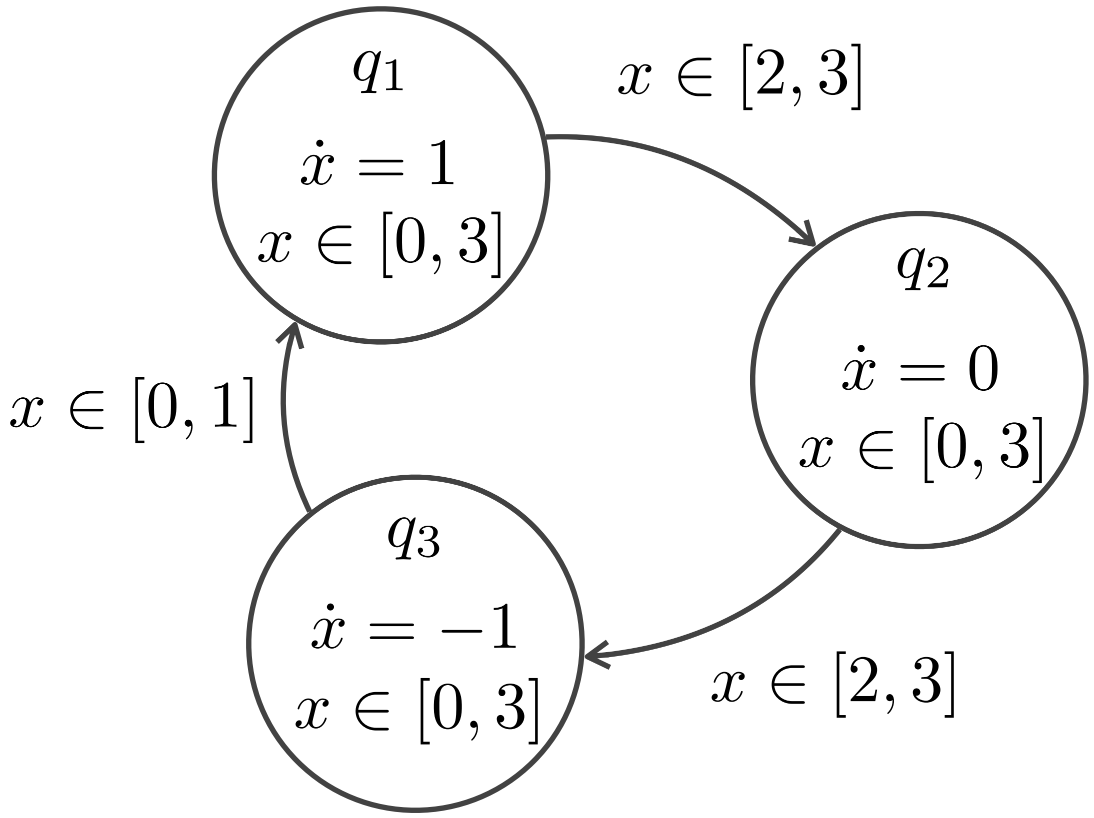
# Linear Hybrid Automaton (LHA)

Definition. A **linear hybrid automaton** (LHA) $H$ is a tuple (Q, E, $\mathbb{R}^n$, Flow, Inv, Grd)

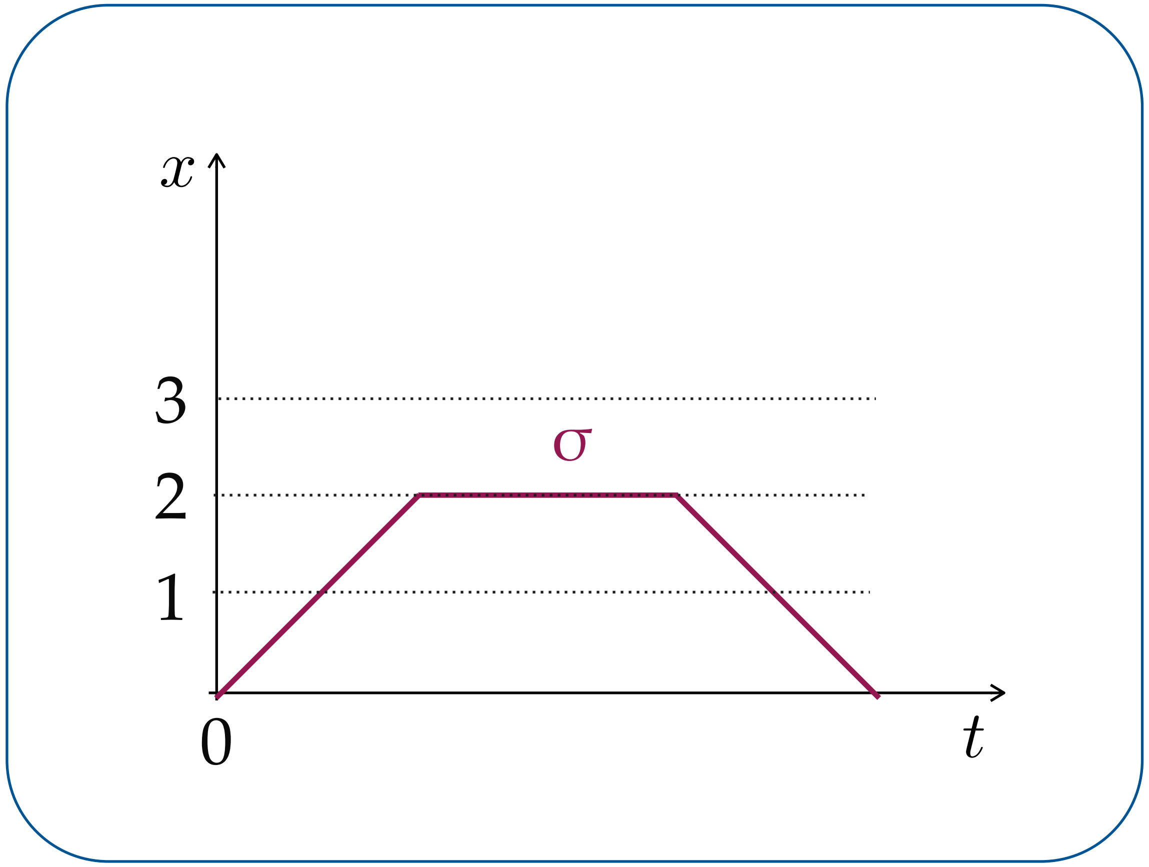
- $Q$ is a finite set of modes,
- E $\in Q$ x $Q$ is a transition relation,
- Flow: $Q \longrightarrow \mathbb{R}^n$ is the flow function,
- Inv: $Q \longrightarrow$ cpoly(n) is the invariant function, and
- Grd: $E \longrightarrow$cpoly(n) is the guard function, where cpoly(n) is the set of compact and convex polyhedral sets over $\mathbb{R}^n$.



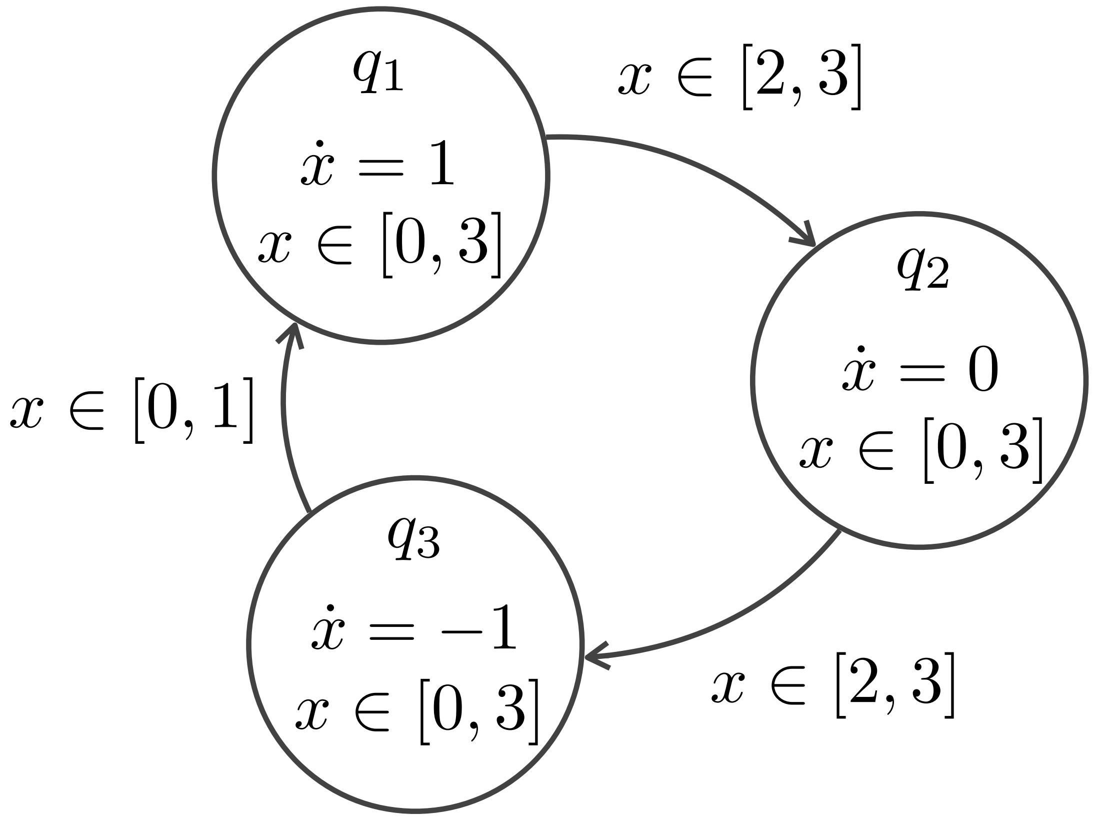
# LHA sample



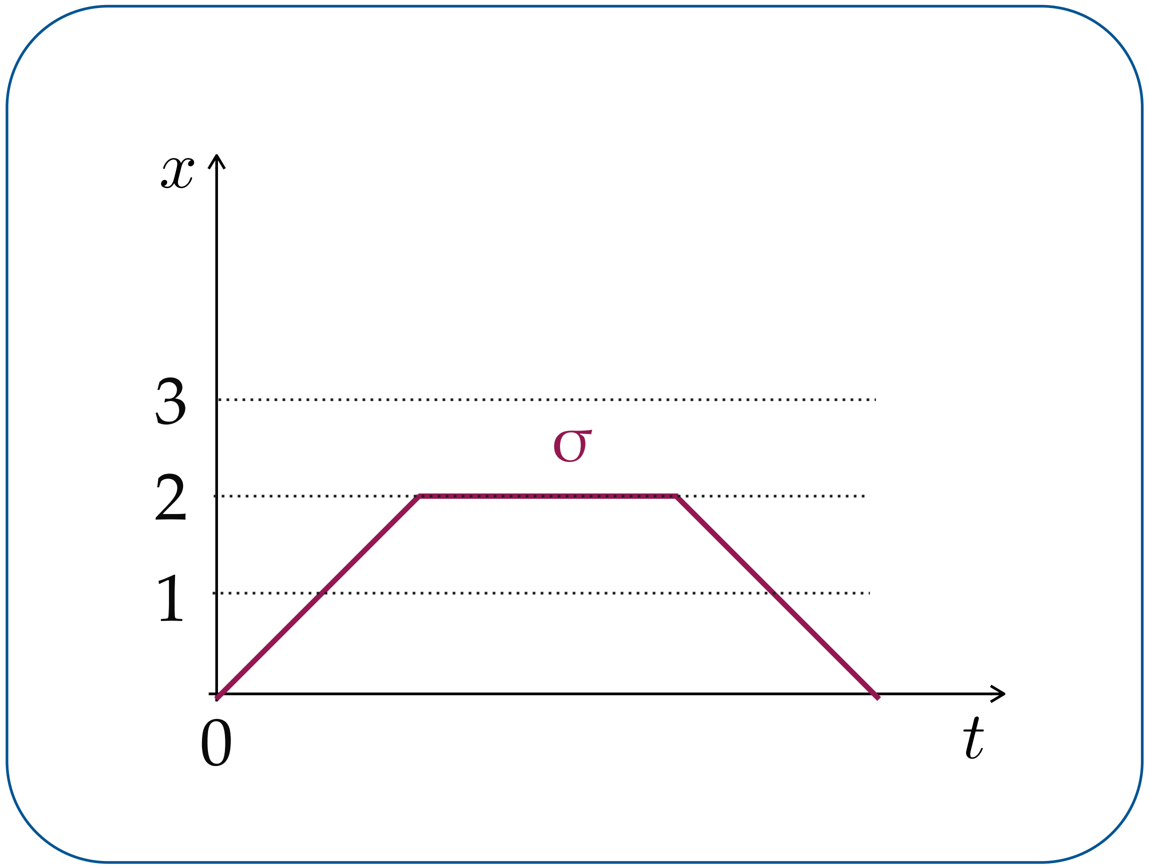

Execution



# LHA sample

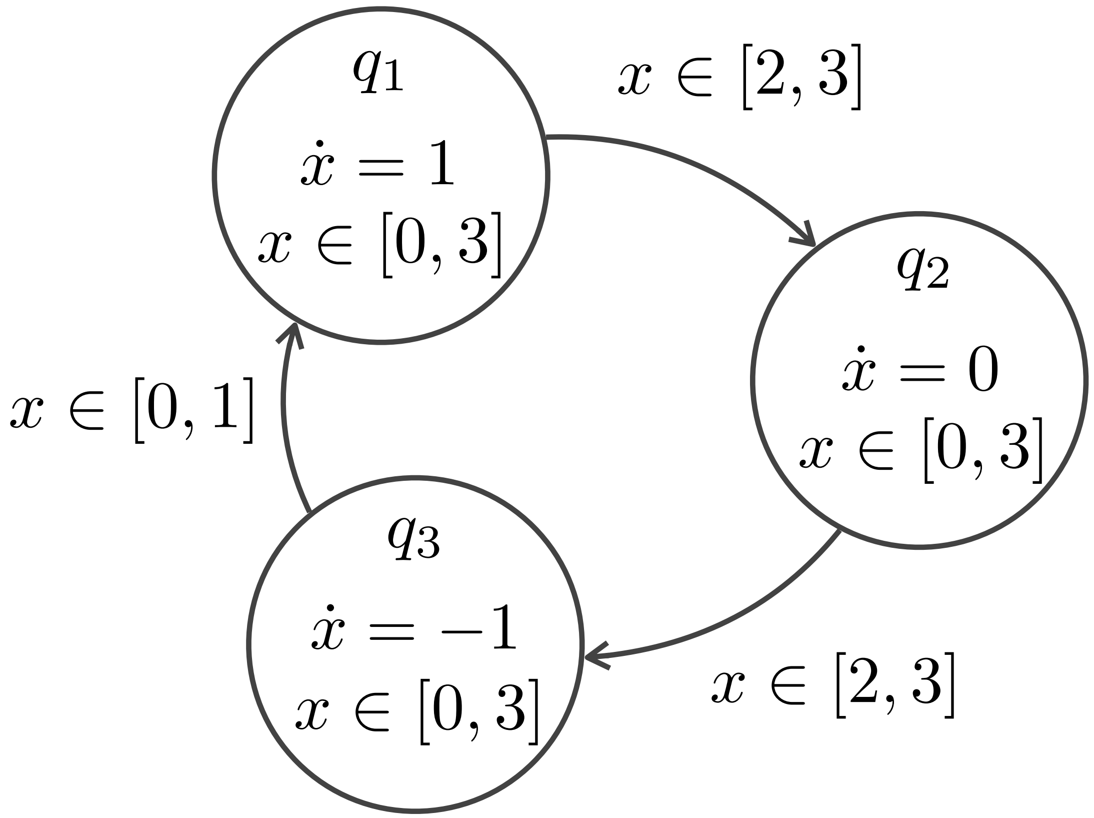



Execution

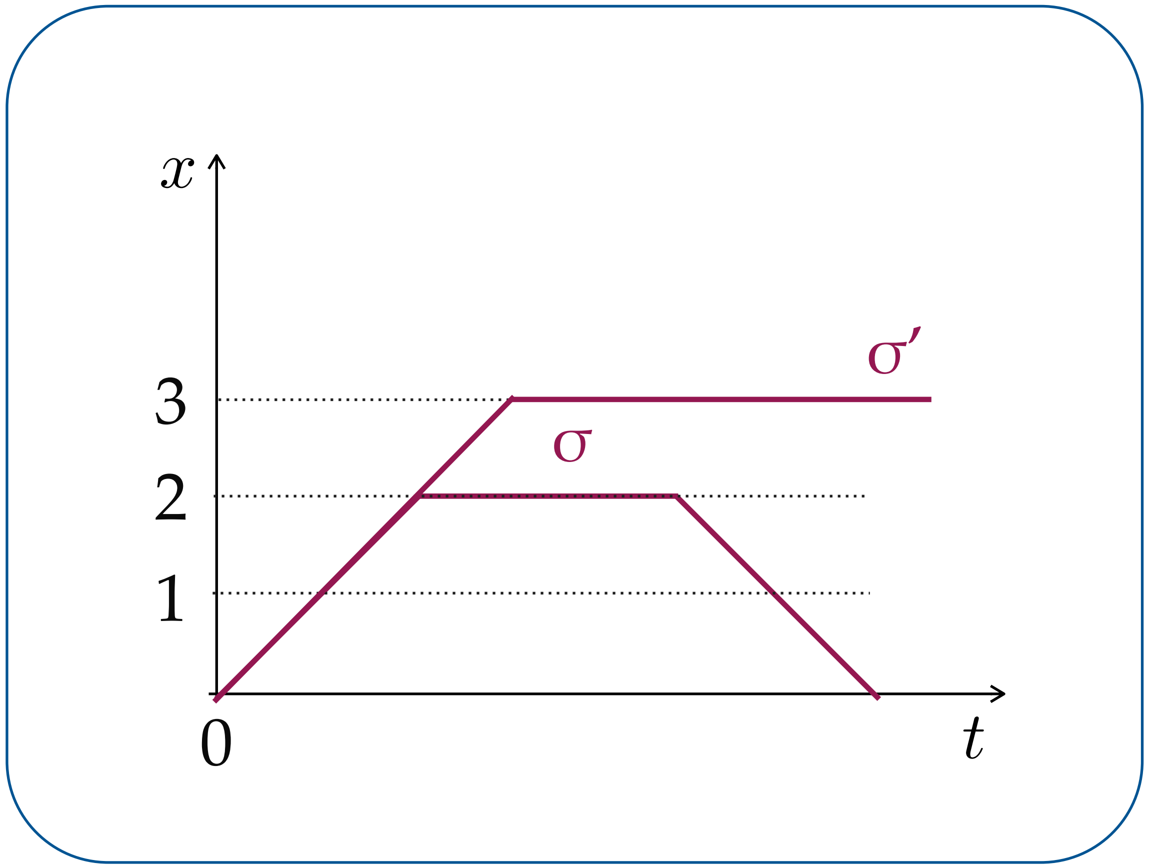



- We consider non deterministic linear hybrid automata
- The LHA features piecewise-linear executions

# LHA sample




Execution



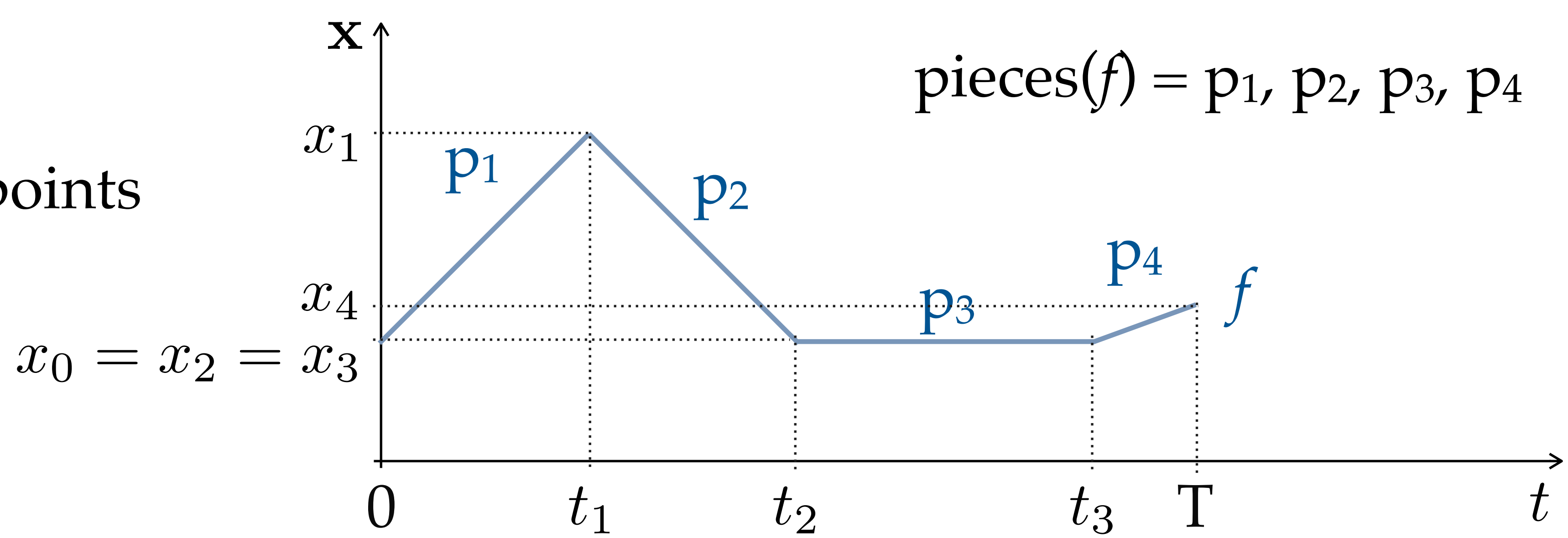

- We consider non deterministic linear hybrid automata
- The LHA features piecewise-linear executions

# Piecewise-linear function

Definition. $f : [0,T] \longrightarrow \mathbb{R}^n$ is an **m-piecewise-linear** (m-PWL) **function** if

- $f \equiv \mathbf{p_1, p_2, \ldots, p_m}$ sequence of m affine pieces of the form $p_i(t) = \mathbf{a_i}t + \mathbf{b_i}$, where:
  - $\mathbf{a_i}$ is the slope($p_i$) and **b** is the initial value
  - $f(t) = p_i(t)$ for $t \in \mathrm{dom}(p_i)$
  - $f$ is continuous

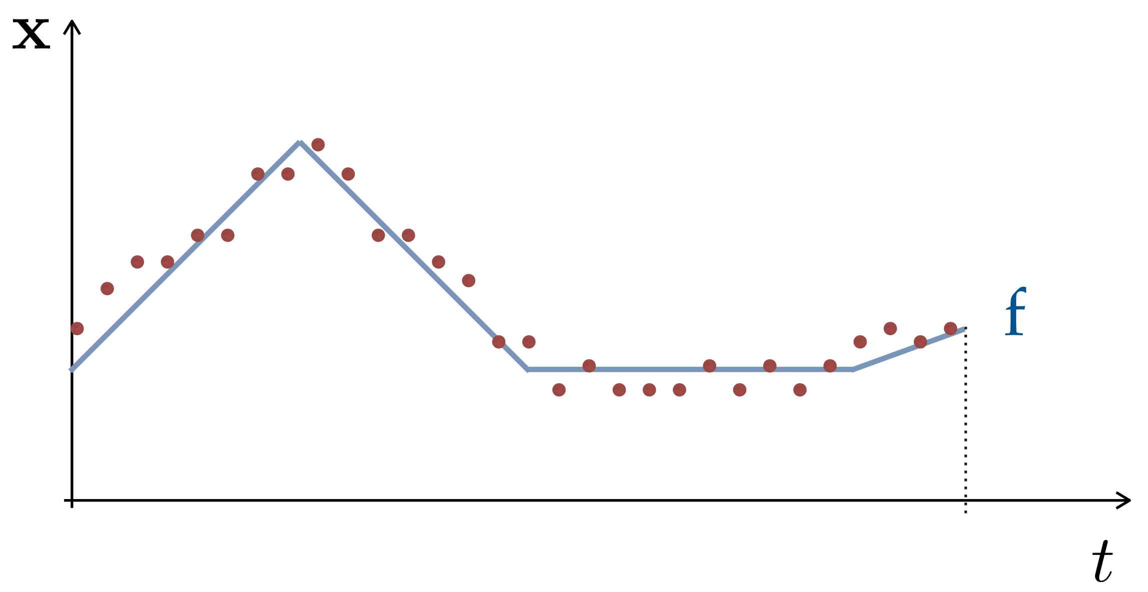
switching points




switching times

# Data: Piecewise-linear approximation

# Time series over-approximation



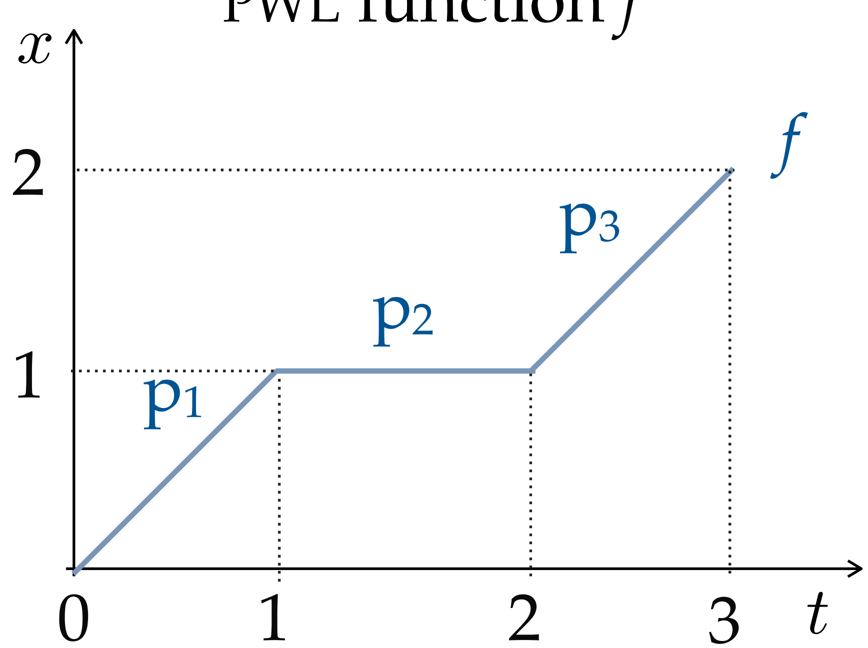

- Douglas-Peucker line simplification algorithm
- Linear regression
- Hakimi-Schmeichel algorithm

# Canonical linear hybrid automaton

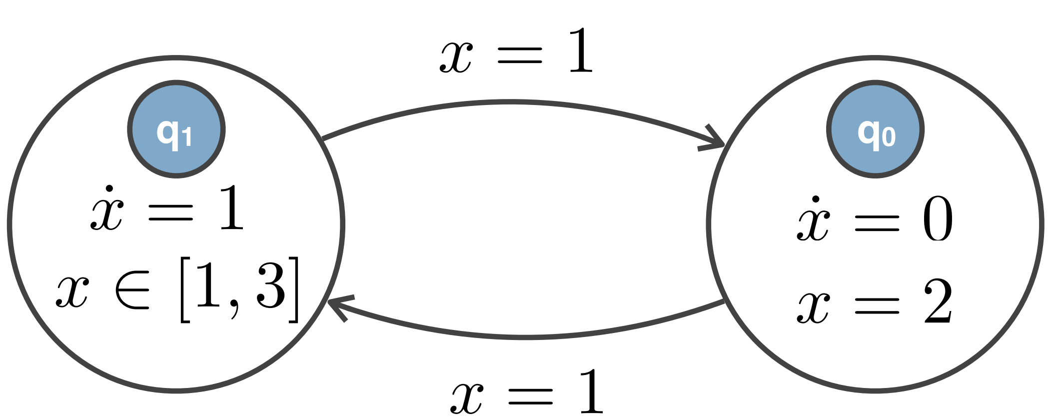
Definition. Let $f$ be an n-dimensional PWL function. Then, the **canonical automaton of $f$** is defined as $H_f = (Q, E, \mathbb{R}^n, \text{Flow}, \text{Inv}, \text{Grd})$ with

- $Q = \{q_a : a \in \mathbb{R}^n, \exists\, p \in \text{pieces}(f) \text{ with slope}(p) = a\}$,
- $E = \{ (q_a, q_{a'}) \in Q \times Q : \exists\, p, p' \in \text{pieces}(f) \text{ adjacent, slope}(p) = a, \text{slope}(p') = a'\}$
- $\text{Flow}(q_a) = a$,
- $\text{Inv}(q_a) = \text{convex\_hull}\,(\{\text{img}(p) : p \in \text{pieces}(f), \text{slope}(p) = a\})$, and
- $\text{Grd}(q_a,q_{a'}) = \text{convex\_hull}\,(\{\text{end\_point}(p) : \exists\, p, p' \in \text{pieces}(f) \text{ adjacent, slope}(p) = a, \text{slope}(p') = a'\})$

PWL function $f$




Canonical automaton $H_f$



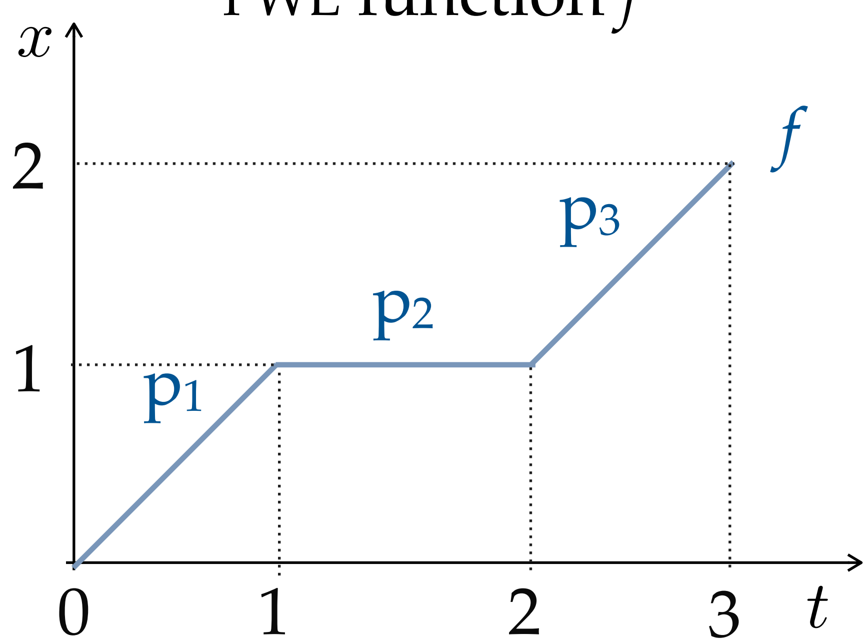

$\text{slope}(p_1) = 1$, $\text{slope}(p_2) = 0$, $\text{slope}(p_1) = 1$

# Canonical linear hybrid automaton

**Definition.** Let $f$ be an n-dimensional PWL function. Then, the **canonical automaton of $f$** is defined as $H_f = (Q, E, \mathbb{R}^n, \text{Flow}, \text{Inv}, \text{Grd})$ with

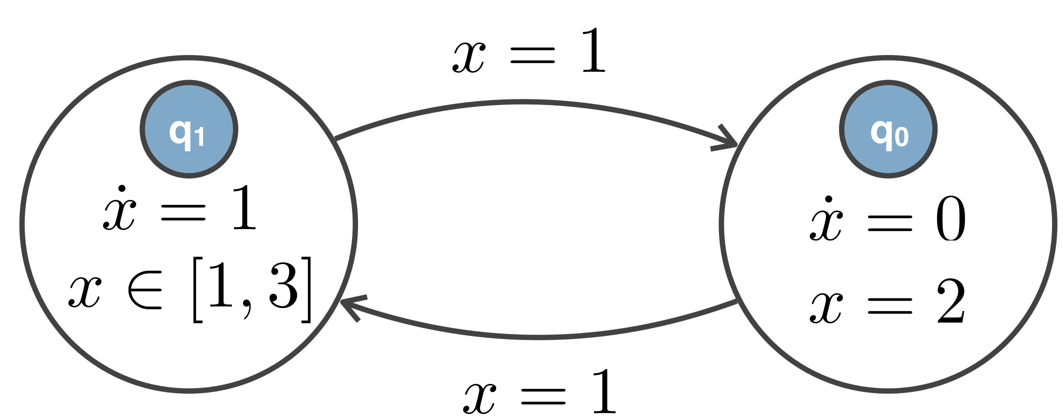
- $Q = \{q_a : a \in \mathbb{R}^n, \exists\, p \in \text{pieces}(f) \text{ with slope}(p) = a\}$,
- $E = \{ (q_a, q_{a'}) \in Q \times Q : \exists\, p, p' \in \text{pieces}(f) \text{ adjacent, slope}(p) = a, \text{slope}(p') = a'\}$
- $\text{Flow}(q_a) = a$,
- $\text{Inv}(q_a) = \text{convex\_hull}(\{\text{img}(p) : p \in \text{pieces}(f), \text{slope}(p) = a\})$, and
- $\text{Grd}(q_a, q_{a'}) = \text{convex\_hull}(\{\text{end\_point}(p) : \exists\, p, p' \in \text{pieces}(f) \text{ adjacent, slope}(p) = a, \text{slope}(p') = a'\})$

PWL function $f$




$\text{slope}(p_1) = 1, \text{slope}(p_2) = 0, \text{slope}(p_1) = 1$

Canonical automaton $H_f$



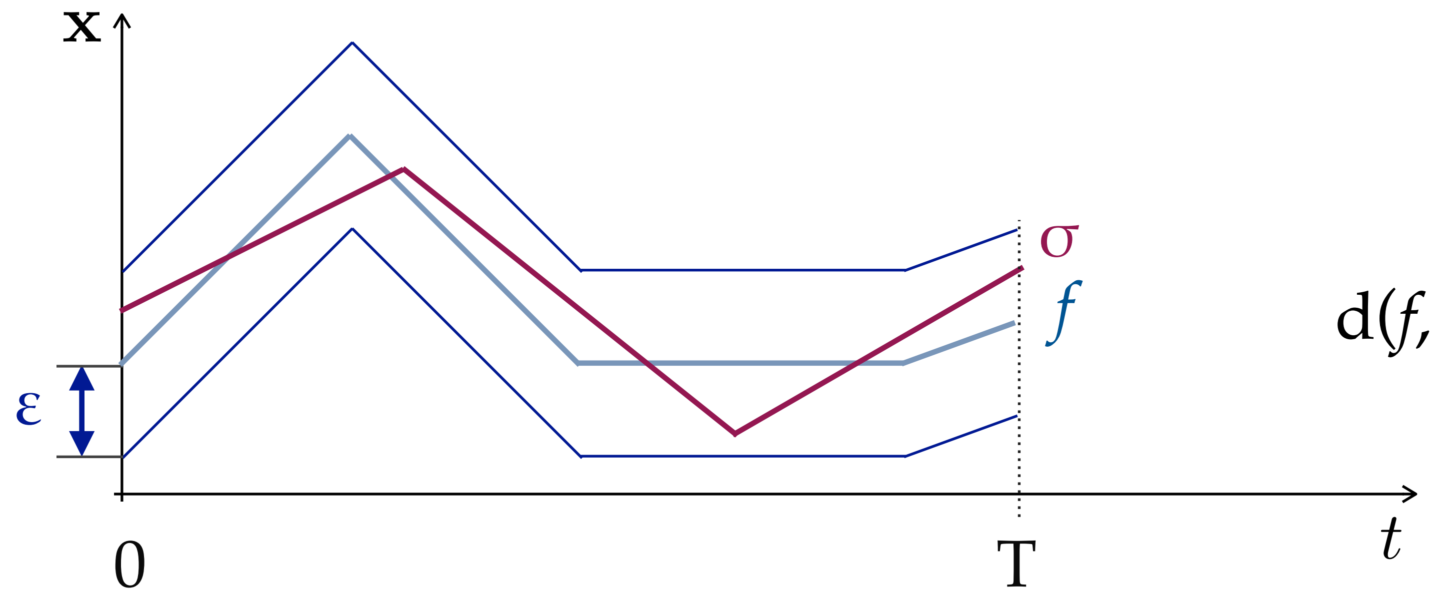

*Given a PWL function $f$, $f$ is an execution of $H_f$*

# Synthesis Problem

# Synthesis problem

Given a finite set of PWL functions $F$ and a value $\varepsilon \in \mathbb{R}_{\geq 0}$, construct an LHA $H$ that ε-captures every function $f \in F$.

Definition. Given a PWL function $f$ and a value $\varepsilon \in \mathbb{R}_{\geq 0}$, we say that an LHA ***H*** **ε-captures *f*** if there exists an execution σ in $H$ with $d(f, \sigma) \leq \varepsilon$.




$$d(f, \sigma) = \max_{t \in [0,T]} \| f(t) - \sigma(t) \|$$

*ε is a trade-off between the size and the precision of the model.*

# Specifications

We solve the synthesis problem for two different specifications, in addition to $H$ $\varepsilon$-capturing every function $f$ in $F$ :

Synchronous specification

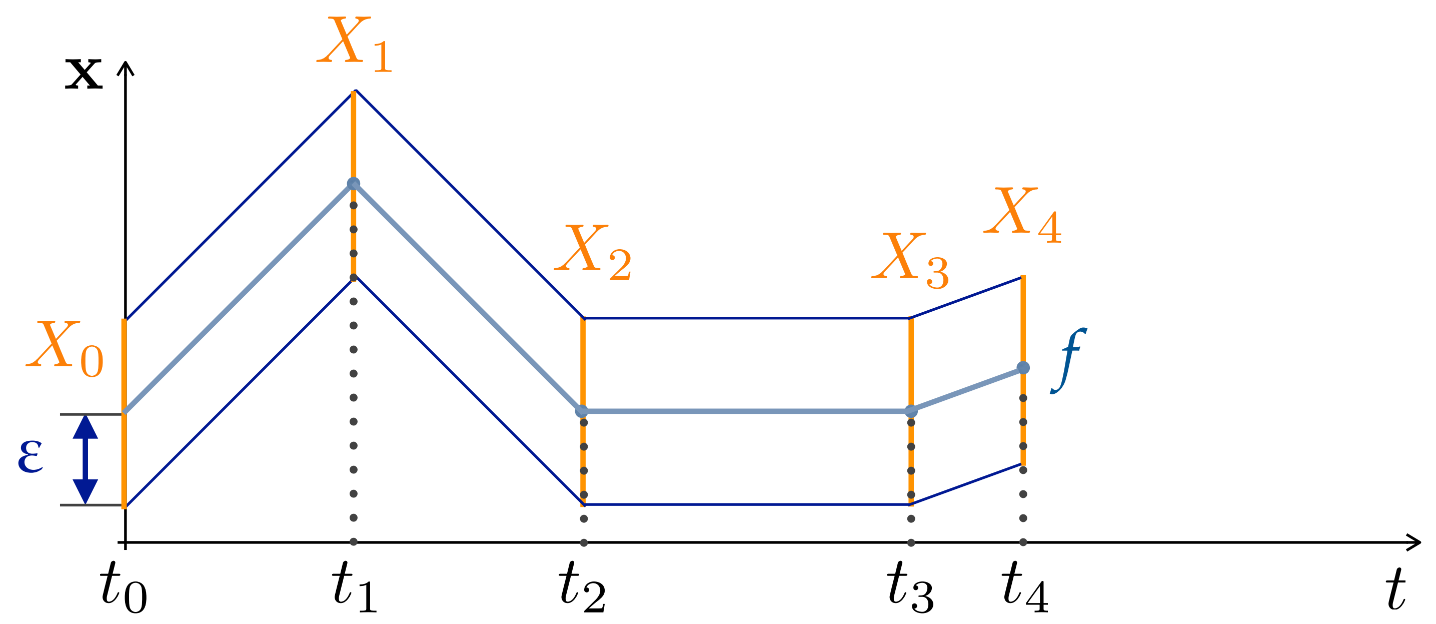
Asynchronous specification

# Synchronous specification

# Synchronous specification

**Specification**

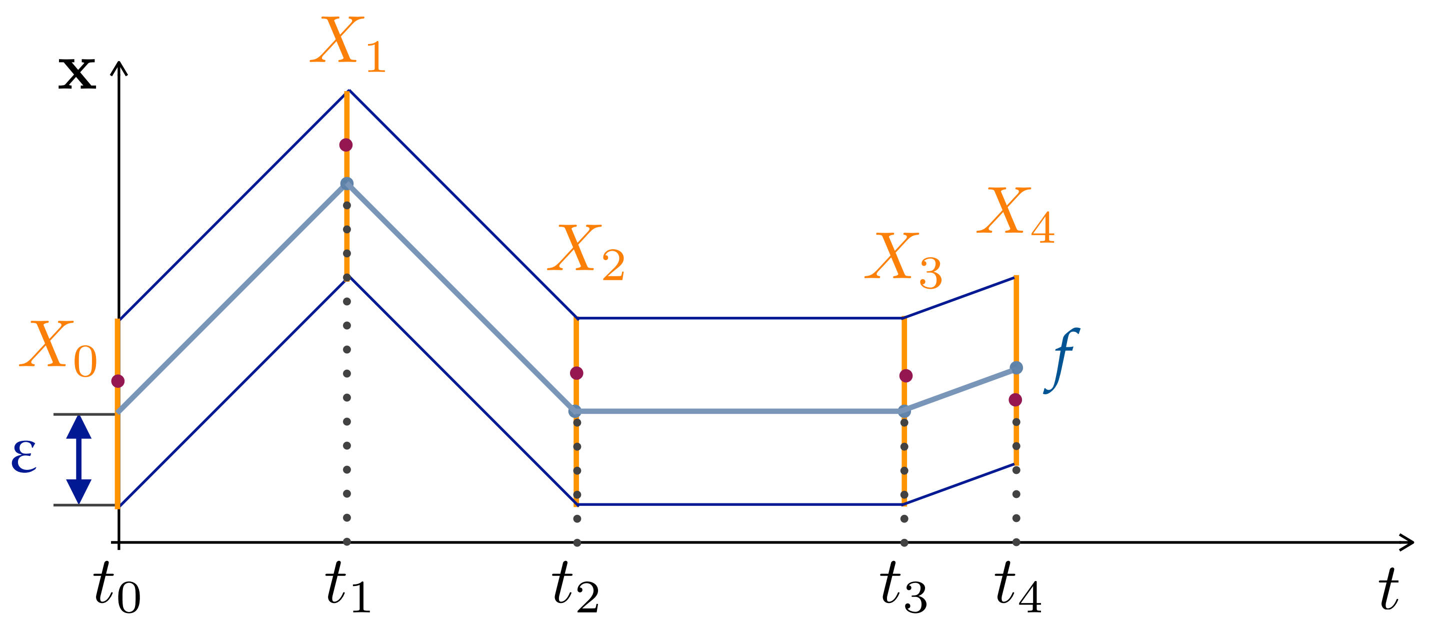
- $H$ ε-captures every function $f$ in $F$
- $H$ switches synchronously with the functions in $F$



# Synchronous specification

**Specification**

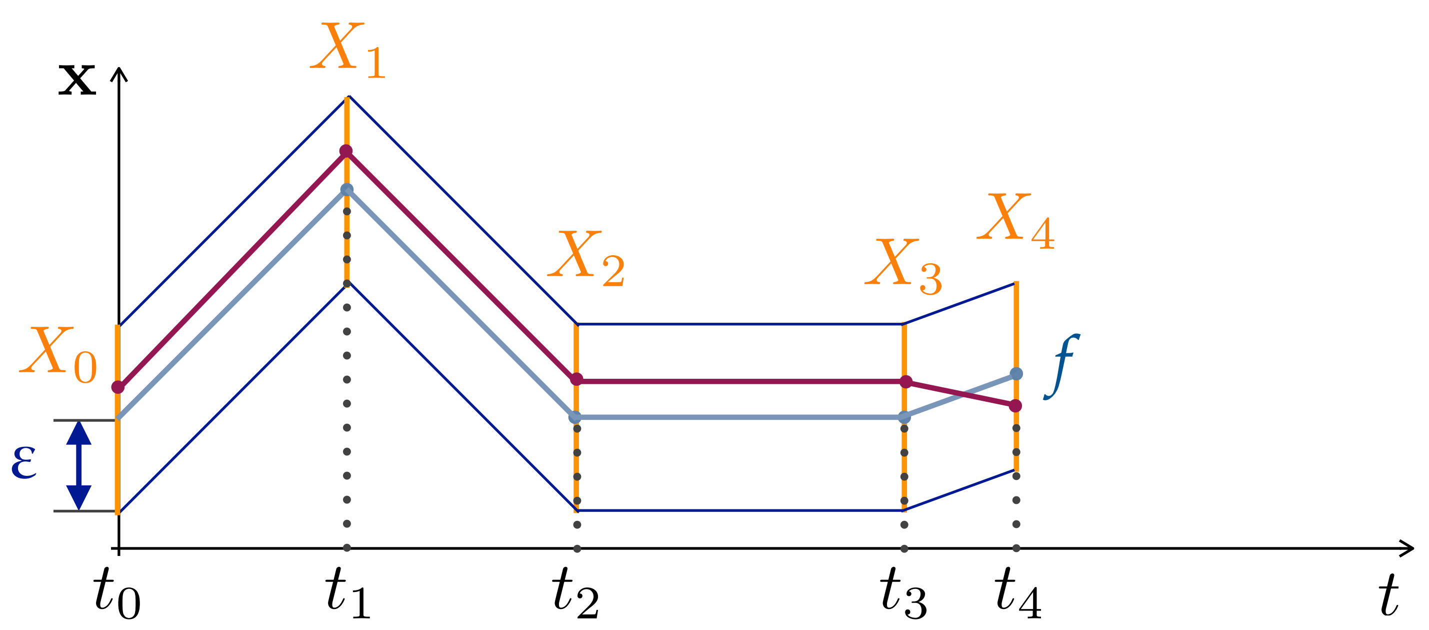
- $H$ ε-captures every function $f$ in $F$
- $H$ switches synchronously with the functions in $F$



# Synchronous specification

**Specification**

- $H$ ε-captures every function $f$ in $F$
- $H$ switches synchronously with the functions in $F$

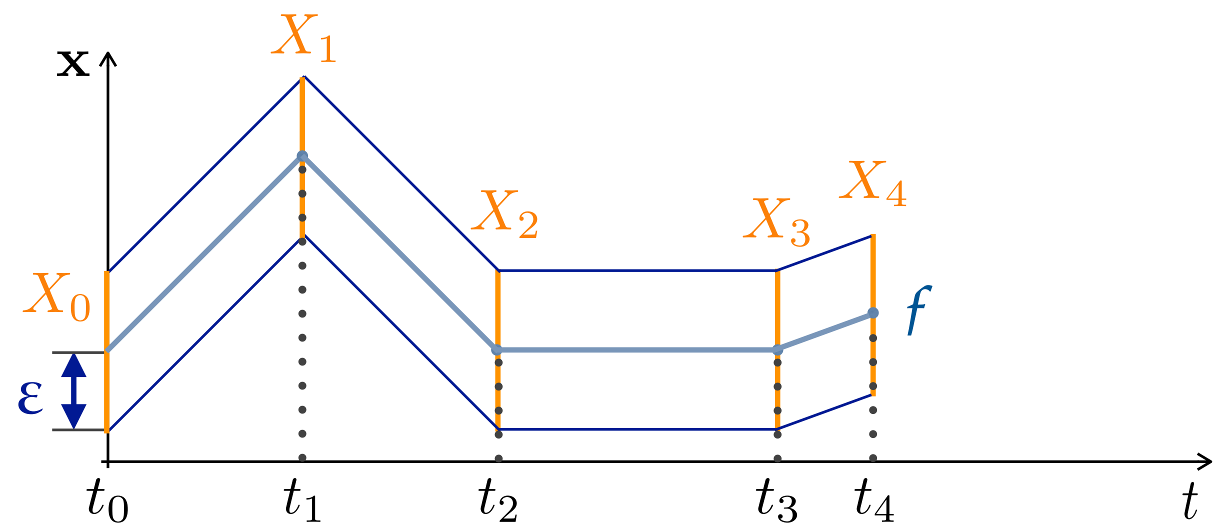


# SMT-based synthesis approach

**SMT-based synthesis approach**

- For every function $f$ in $F$, construct a PWL function $g$ satisfying the specification with $\ell$ different slopes:
  - $g \equiv q_1,\ldots,q_m$
  - Switching times of $g$ are the same as switching times of $f$ : $t_0,\ldots,t_m$
  - Switching points of $g$ are $\mathbf{y}_0,\ldots,\mathbf{y}_m$ and they are ε-close to switching points of $f$, $\mathbf{x}_0 = f(t_0),\ldots,\mathbf{x}_m = f(t_m) \longrightarrow$ Expressed as $\mathbf{y}_j \in X_j$
  - $\mathbf{b}_1 = \text{slope}(q_1)$, …, $\mathbf{b}_m = \text{slope}(q_m)$ with $\ell$ different values $(\mathbf{c}_1,\ldots,\mathbf{c}_\ell)$

$$\phi_{f,\varepsilon}(\ell) \equiv \bigwedge_{j=1}^{m} \mathbf{y}_j = \mathbf{y}_{j-1} + \mathbf{b}_j(t_j - t_{j-1}) \wedge \bigwedge_{j=0}^{m} \mathbf{y}_j \in X_j \wedge \bigwedge_{j=1}^{m} \bigvee_{k=1}^{\ell} \mathbf{b}_j = \mathbf{c}_k$$

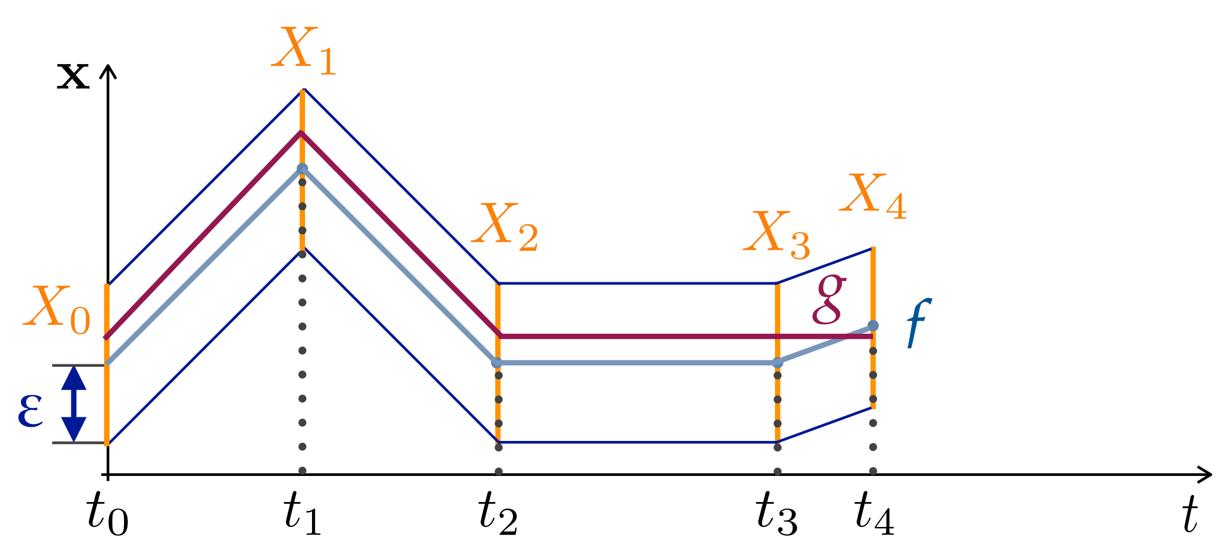


# SMT-based synthesis approach

## SMT-based synthesis approach

- For every function $f$ in $F$, construct a PWL function $g$ satisfying the specification with $\ell$ different slopes:
  - $g \equiv q_1,\ldots,q_m$
  - Switching times of $g$ are the same as switching times of $f$ : $t_0,\ldots,t_m$
  - Switching points of $g$ are $\mathbf{y}_0,\ldots,\mathbf{y}_m$ and they are ε-close to switching points of $f$, $\mathbf{x}_0 = f(t_0),\ldots,\mathbf{x}_m = f(t_m) \longrightarrow$ Expressed as $\mathbf{y}_j \in X_j$
  - $\mathbf{b}_1 = \text{slope}(q_1)$, …, $\mathbf{b}_m = \text{slope}(q_m)$ with $\ell$ different values $(\mathbf{c}_1,\ldots,\mathbf{c}_\ell)$

$$\phi_{f,\varepsilon}(\ell) \equiv \bigwedge_{j=1}^{m} \mathbf{y}_j = \mathbf{y}_{j-1} + \mathbf{b}_j(t_j - t_{j-1}) \wedge \bigwedge_{j=0}^{m} \mathbf{y}_j \in X_j \wedge \bigwedge_{j=1}^{m} \bigvee_{k=1}^{\ell} \mathbf{b}_j = \mathbf{c}_k$$




*g satisfies* $\phi_{f,\varepsilon}(3)$

**<u>SMT-based synthesis approach</u>**

- Global linear arithmetic formula for *F*

$$\phi_{F,\varepsilon}(\ell) \equiv \bigwedge_{f\in F} \phi_{f,\varepsilon}(\ell) \quad \Rightarrow \quad \textit{SMT solver}$$

- Minimization of the number of different slopes

$$\min_{\ell \leq M} \phi_{F,\varepsilon}(\ell) \text{ is satisfiable}$$

$$M = \#\ \text{different slopes in } F \quad \Rightarrow \quad G = \{g_f : f \in F\}$$

- Canonical automaton of *G* $\Rightarrow$ $H_G$

# SMT-based synthesis approach

**SMT-based synthesis approach**

- Global linear arithmetic formula for $F$

$$\phi_{F,\varepsilon}(\ell) \equiv \bigwedge_{f\in F} \phi_{f,\varepsilon}(\ell) \quad \Rightarrow \quad \textit{SMT solver}$$

- Minimization of the number of different slopes

$$\min_{\ell \leq M} \phi_{F,\varepsilon}(\ell) \text{ is satisfiable}$$

$$M = \# \text{ different slopes in } F \quad \Rightarrow \quad G = \{g_f : f \in F\}$$

- Canonical automaton of $G$ $\quad \Rightarrow \quad H_G$

**Theorem - SMT-based synthesis**

Given a finite set of PWL functions $F$ and a value $\varepsilon \geq 0$, the LHA $H_G$ solves the synthesis problem with **minimal number of modes**.

# SMT-based synthesis approach

- **SMT-based** synthesis **approach** provides an **optimal global solution**
- Works well with **short** and **low-dimensional** input PWL **functions**
- Does **not scale** to realistic problem sizes

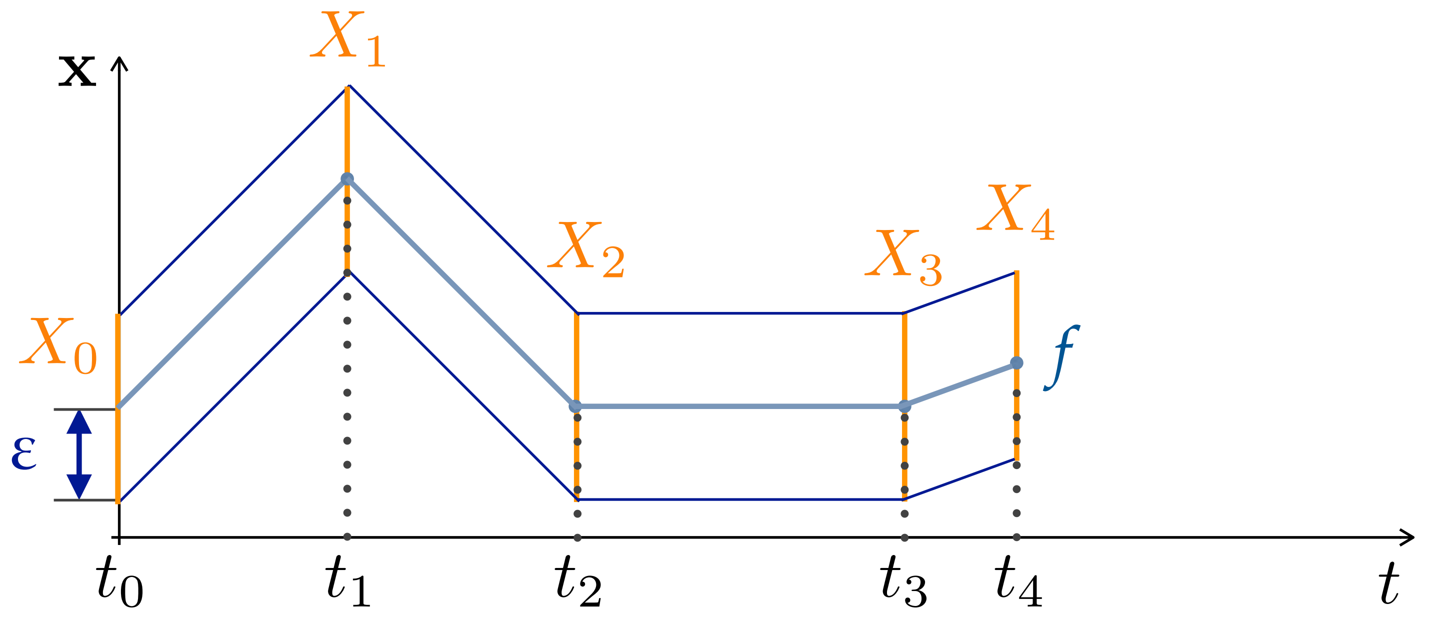
**Alternate approach**
Membership-based synthesis approach

# Asynchronous specification

# Asynchronous specification

**Specification**

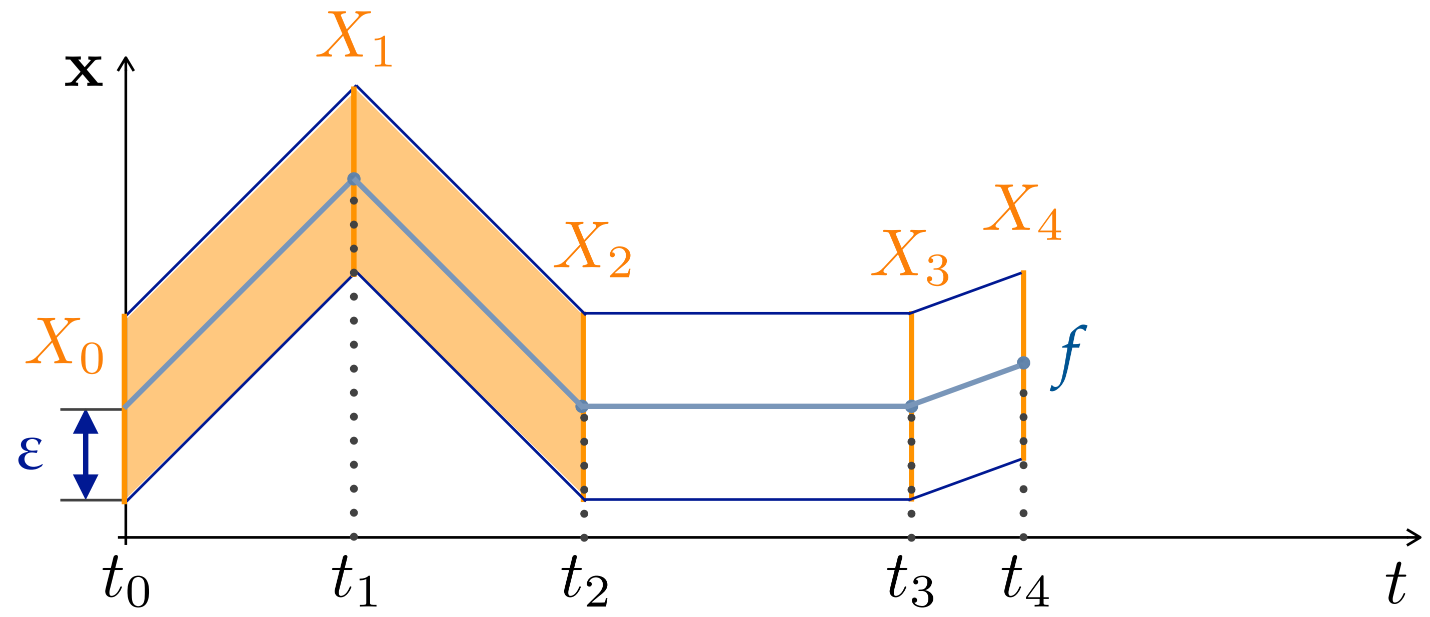
- $H$ ε-captures every function $f$ in $F$
- $H$ switches in the time interval determined by the previous and posterior time instants in the functions $f$ in $F$



# Asynchronous specification

**Specification**

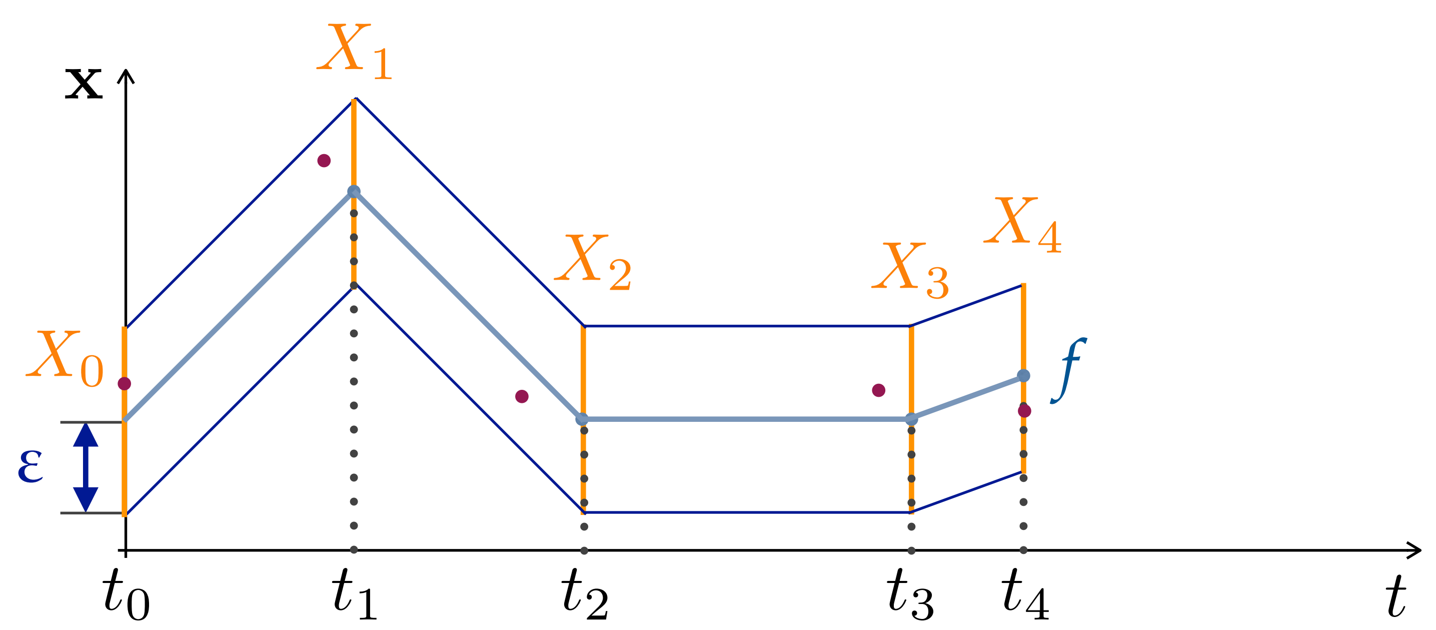
- $H$ ε-captures every function $f$ in $F$
- $H$ switches in the time interval determined by the previous and posterior time instants in the functions $f$ in $F$



# Asynchronous specification

**Specification**

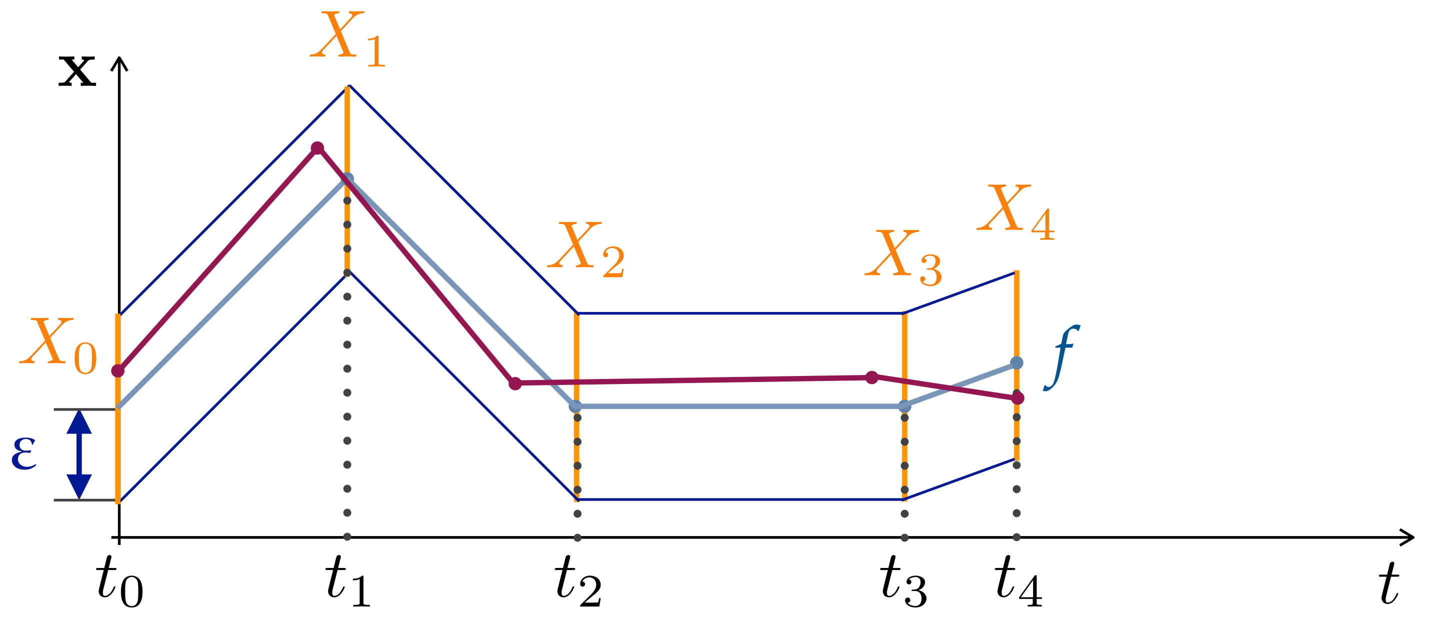
- $H$ ε-captures every function $f$ in $F$
- $H$ switches in the time interval determined by the previous and posterior time instants in the functions $f$ in $F$



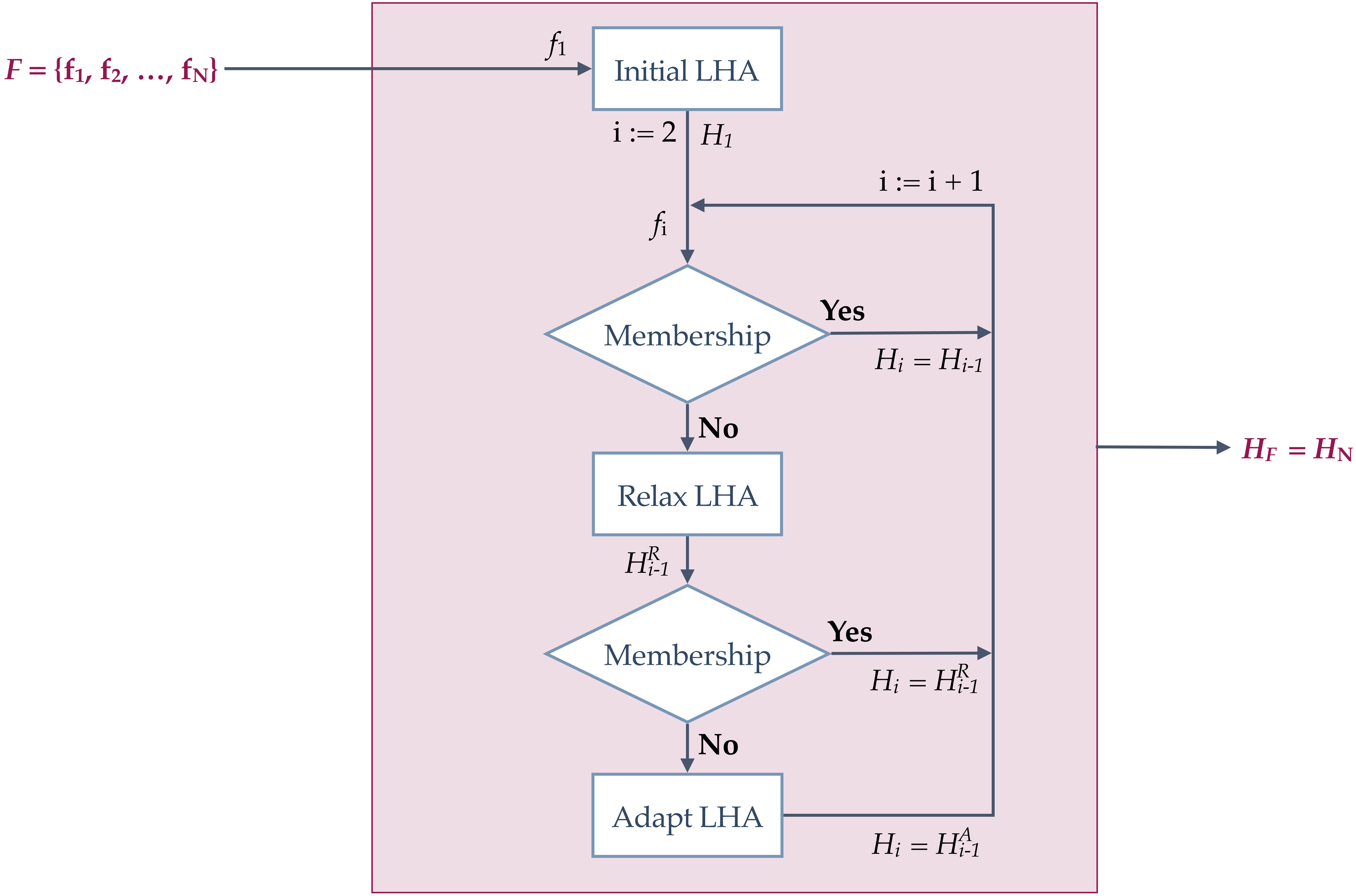
# Asynchronous specification

**Specification**

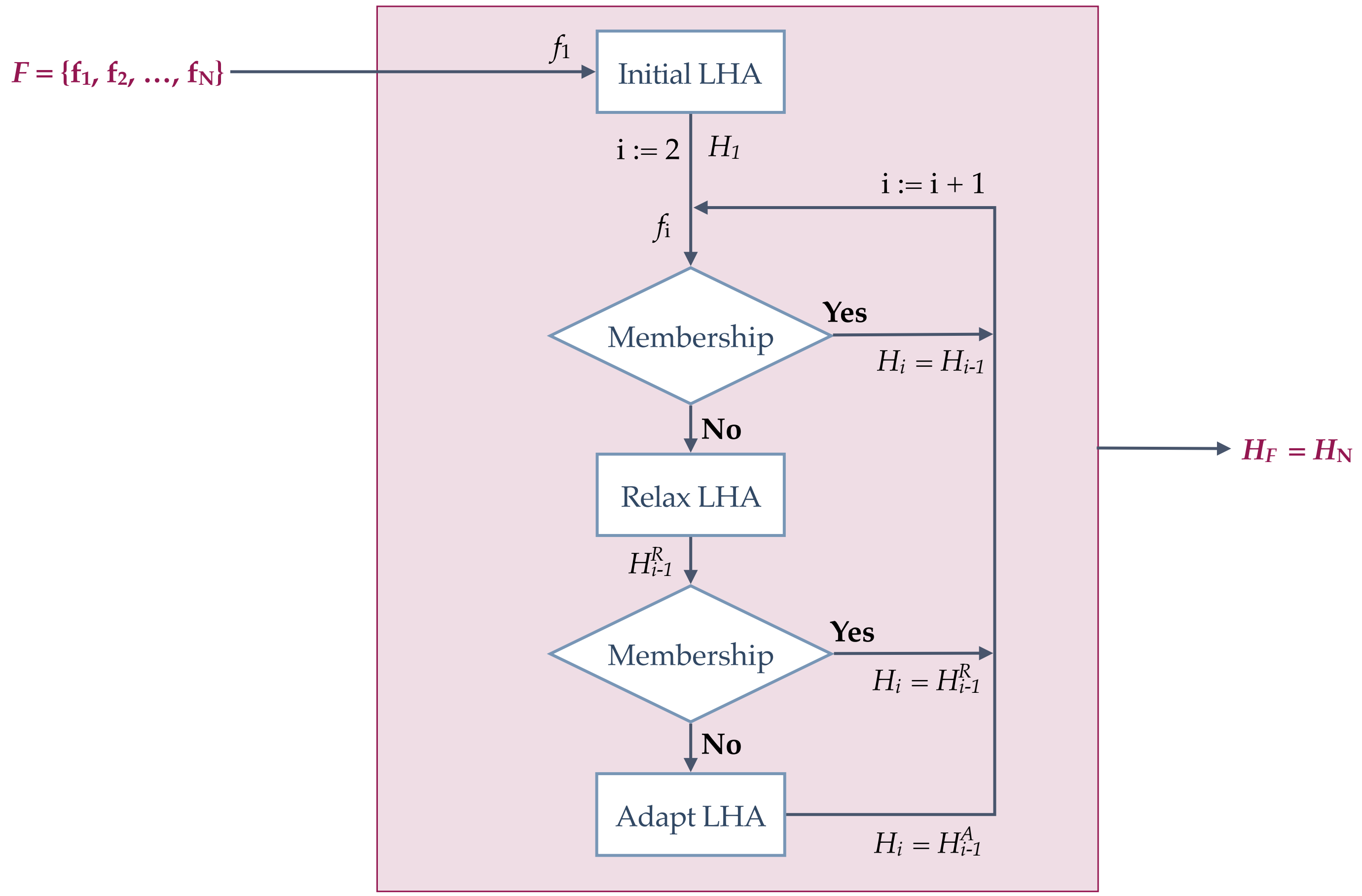
- $H$ ε-captures every function $f$ in $F$
- $H$ switches in the time interval determined by the previous and posterior time instants in the functions $f$ in $F$



# Membership-based synthesis approach



# Membership-based synthesis approach




- **Systematically iterates** over new data
- A **positive** initial **membership** result requires **no modification** of the LHA
- **Returns** a **counterexample** when the **membership** in the relaxed LHA **fails**
- The **counterexample** is used to not only increment the guards and invariants of the LHA but to **add** discrete modes

# Membership

# Membership problem

Given an LHA *H* and a PWL function *f*, decide if there exists an execution σ in *H* consistent with *f* and such that $d(f, \sigma) \leq \varepsilon$

Definition. An execution σ of ***H*** is **consistent with** an m-PWL function ***f*** $\equiv p_1,\ldots,p_m$ if σ has m affine pieces and its switching times $t_1,\ldots,t_{m-1}$ are such that $t_i \in \text{domain}(p_i \cup p_{i+1})$

# Membership problem

Given an LHA *H* and a PWL function *f*, decide if there exists an execution $\sigma$ in *H* consistent with *f* and such that $d(f, \sigma) \leq \varepsilon$

Definition. An execution $\sigma$ of ***H*** is **consistent with** an m-PWL function ***f*** $\equiv p_1,\ldots,p_m$ if $\sigma$ has m affine pieces and its switching times $t_1,\ldots,t_{m-1}$ are such that $t_i \in domain(p_i \cup p_{i+1})$

**Membership procedure**

- Construct the **$\varepsilon$-tube of *f***
- Search for all mode **paths** $\pi$ **of length m in *H***
- Search for **executions** determined by $\pi$ **satisfying** the problem **constraints**

# Membership problem

Given an LHA *H* and a PWL function *f*, decide if there exists an execution σ in *H* consistent with *f* and such that $d(f, \sigma) \leq \varepsilon$

Definition. An execution σ of ***H*** is **consistent with** an m-PWL function ***f*** $\equiv p_1,\ldots,p_m$ if σ has m affine pieces and its switching times $t_1,\ldots,t_{m-1}$ are such that $t_i \in \text{domain}(p_i \cup p_{i+1})$

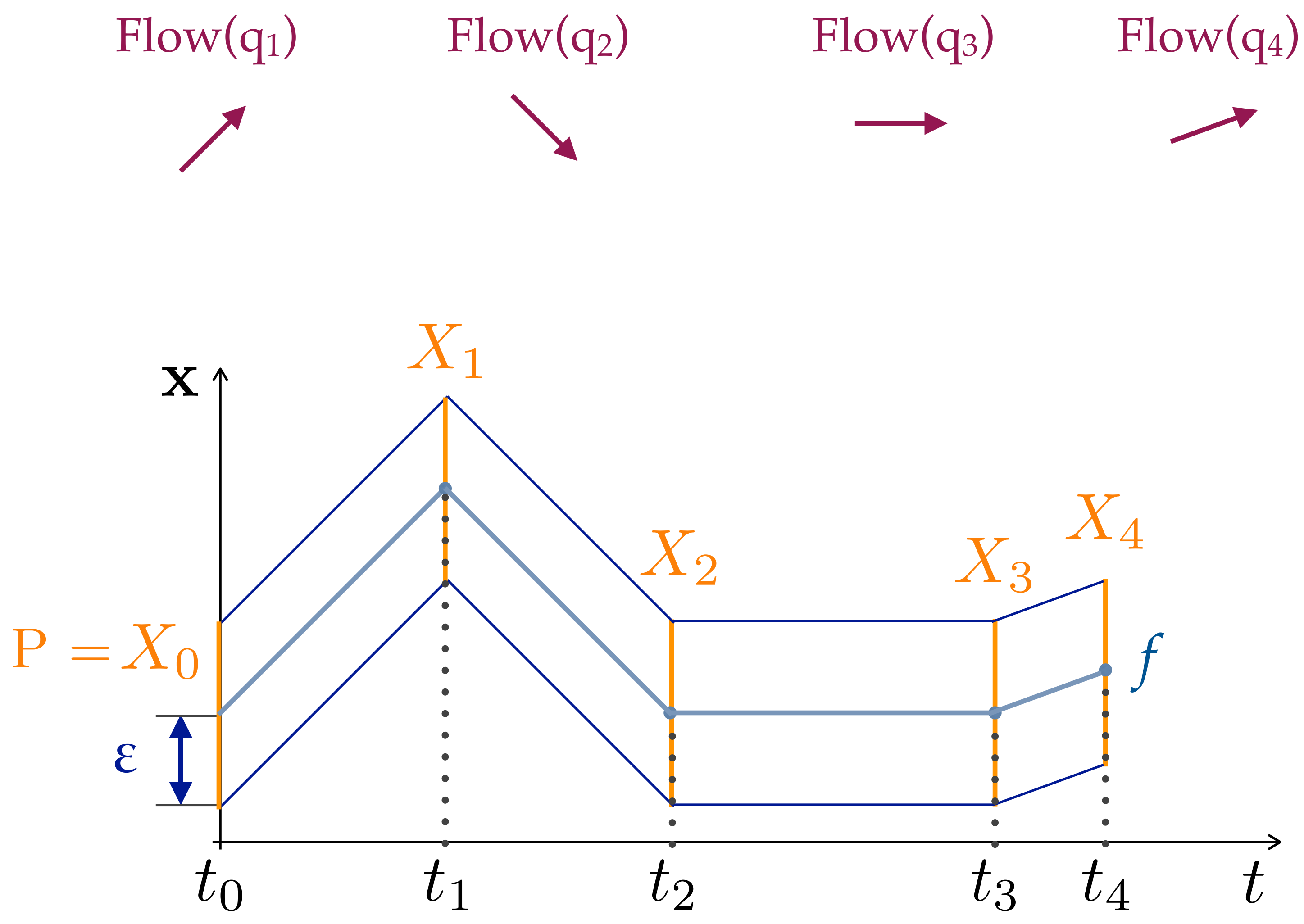
**Membership procedure**

- Construct the **ε-tube of *f***
- Search for all mode **paths** π **of length m in *H***
- Search for **executions** determined by π **satisfying** the problem **constraints**

**Analysis core:**
reachability computation

# Reachability computation

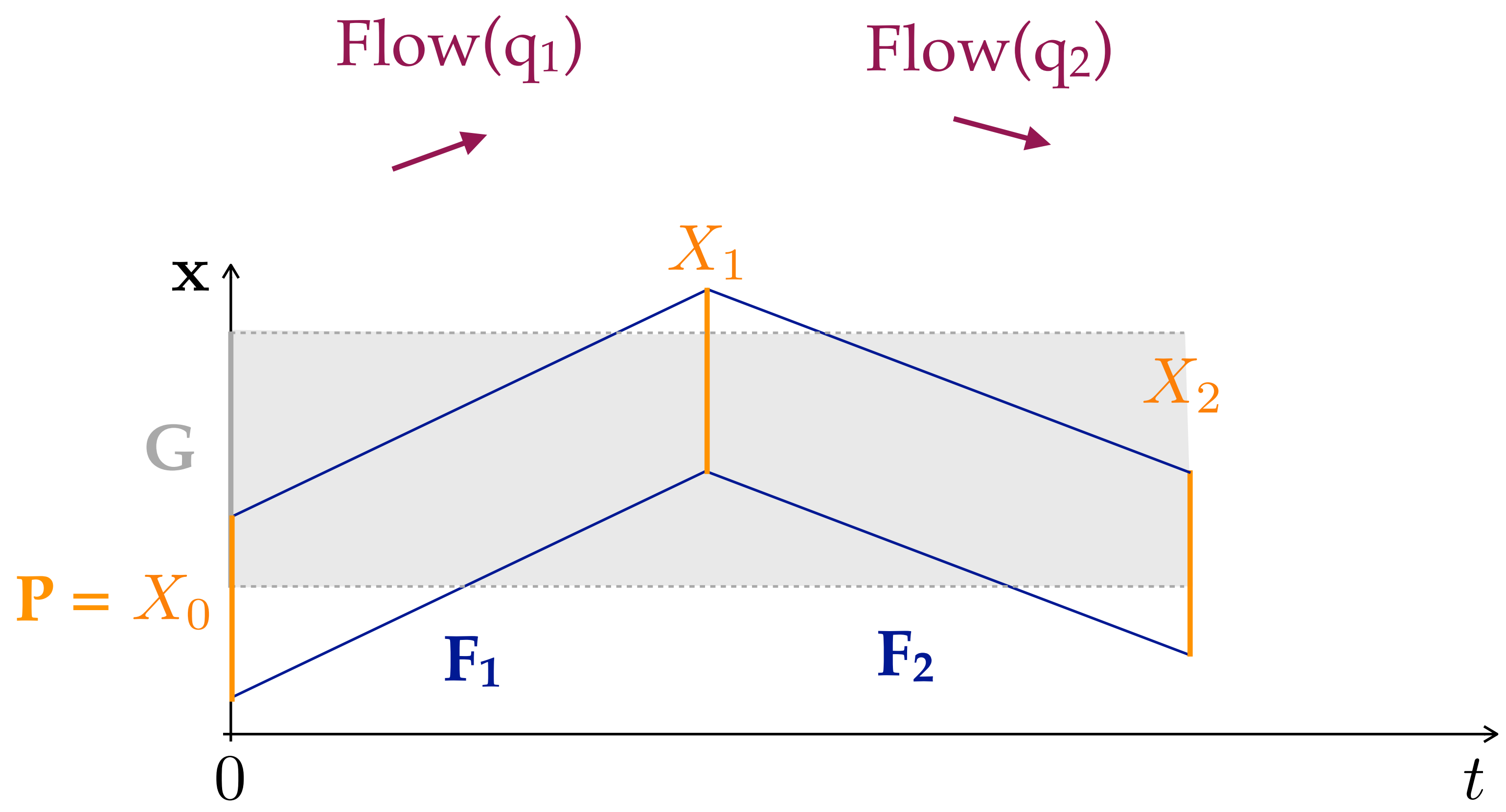
- Linear hybrid automaton $H$ and PWL function $f$
- $\pi \equiv q_1, q_2, q_3, q_4$ mode path in $H$



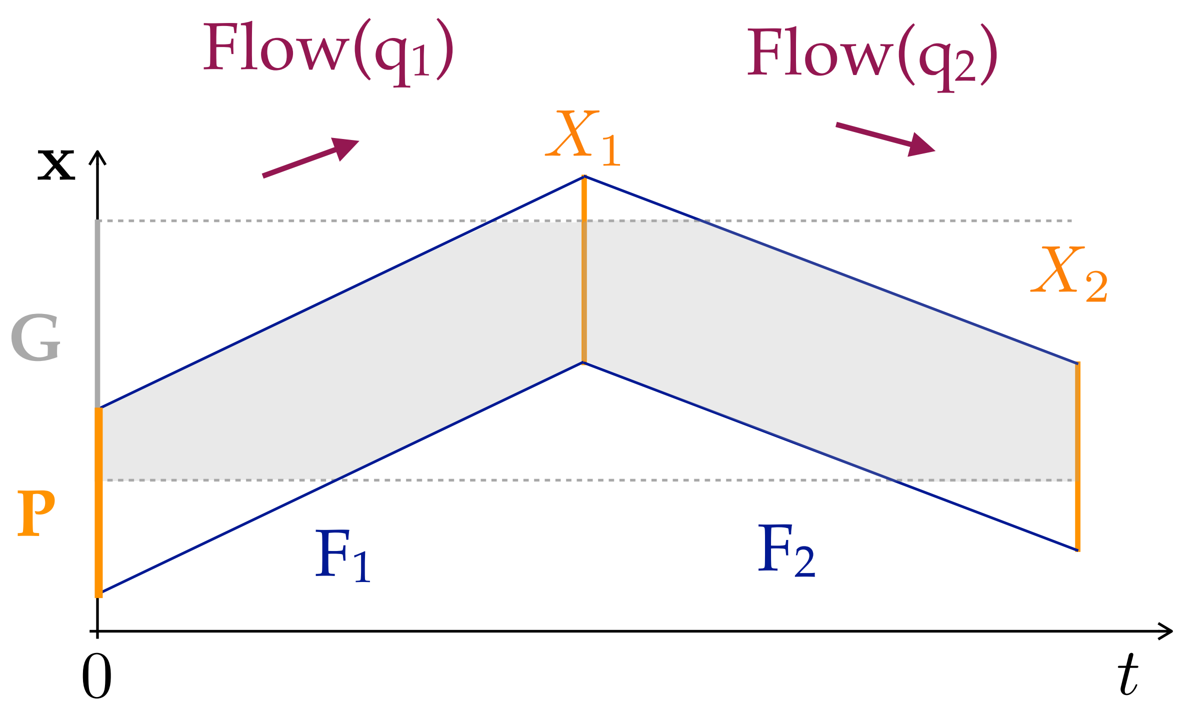
# Reachability computation

**Basic computation: reachable switching set**

- Consider two ε-tube pieces $\mathbf{F_1}$ and $\mathbf{F_2}$
- Consider two modes, $q_1$ and $q_2$, their flows, **Flow($q_1$)** and **Flow($q_2$)**, and the guard from $q_1$ to $q_2$, **G**.
- Consider an initial polyhedral set **P**



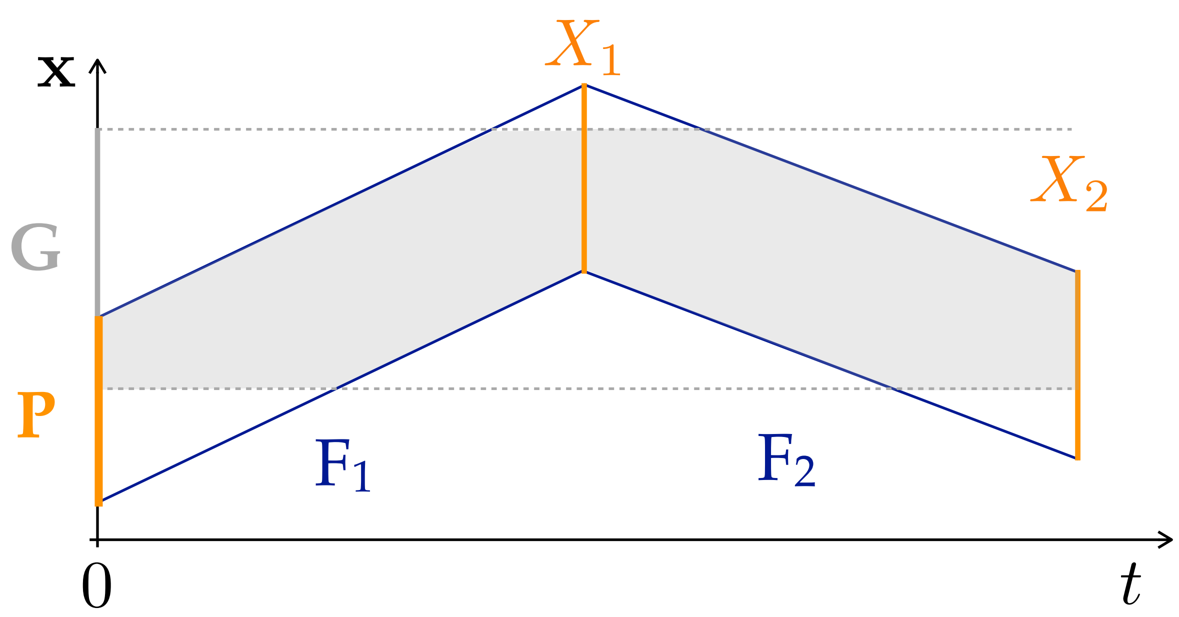
# Reachability computation




$$A^{aux} = POST\ (P, G_1, Flow(q_1))$$

$$A = PRE\ (X_1, A^{aux}, Flow(q_2))$$

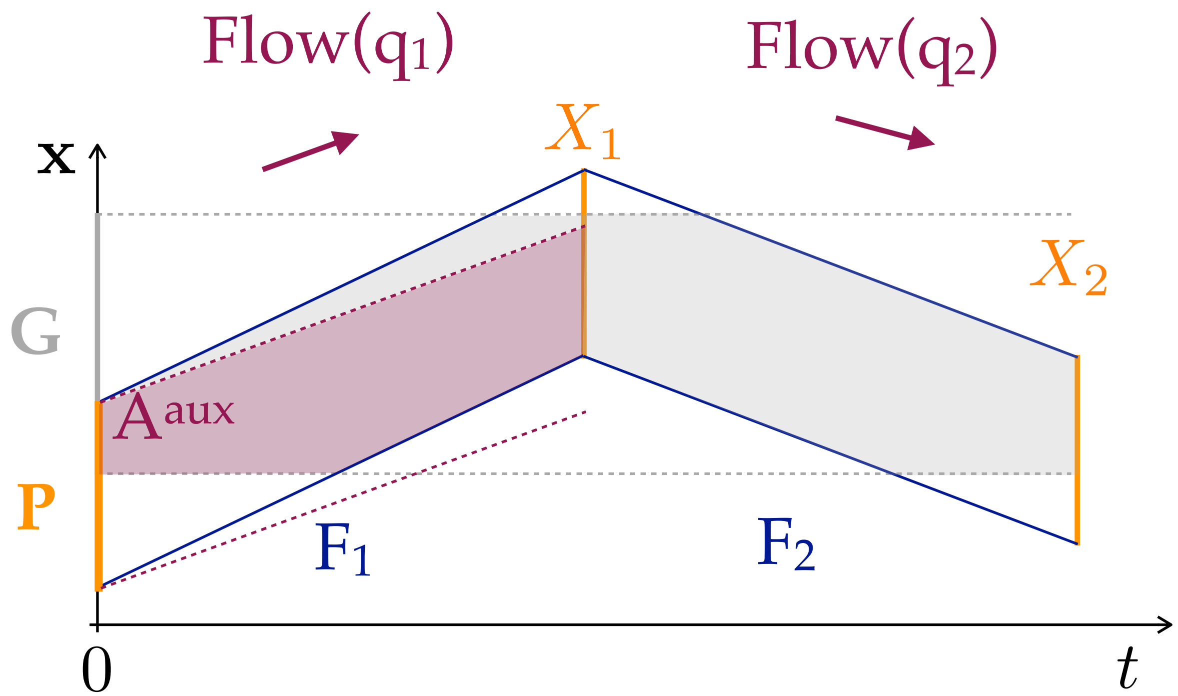
$G_1 = F_1 \cap G \qquad G_2 = F_2 \cap G$




$$B^{aux} = PRE\ (X_1, P, Flow(q_1))$$

$$B = POST\ (B^{aux}, G_2, Flow(q_1))$$

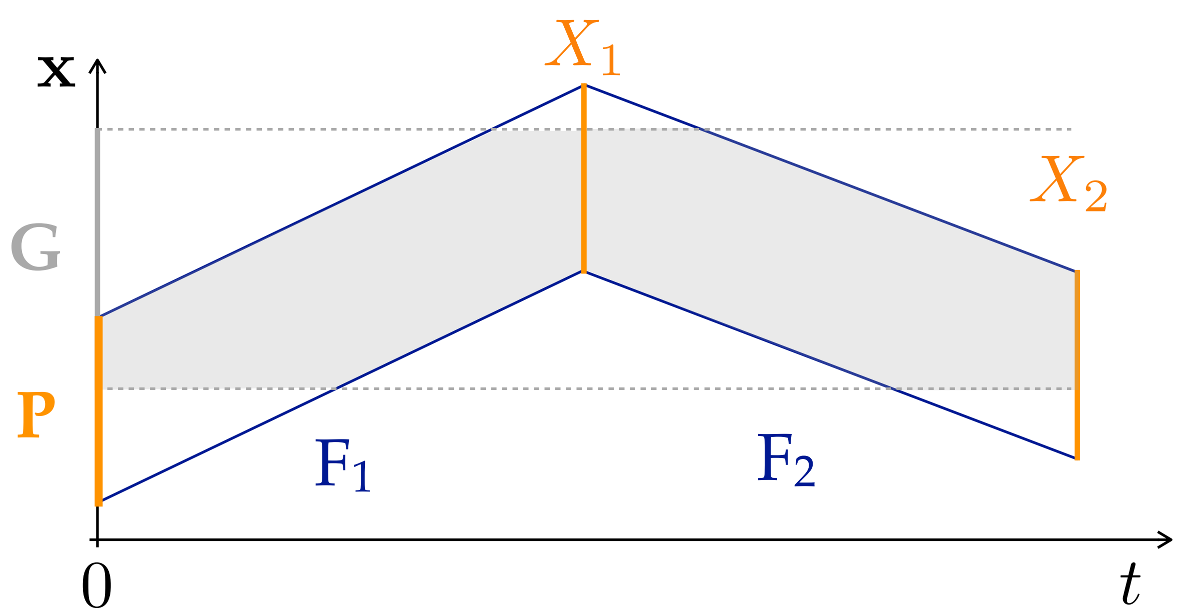
# Reachability computation




$G_1 = F_1 \cap G$ $\quad$ $G_2 = F_2 \cap G$

$A^{aux} = POST\ (P, G_1, Flow(q_1))$

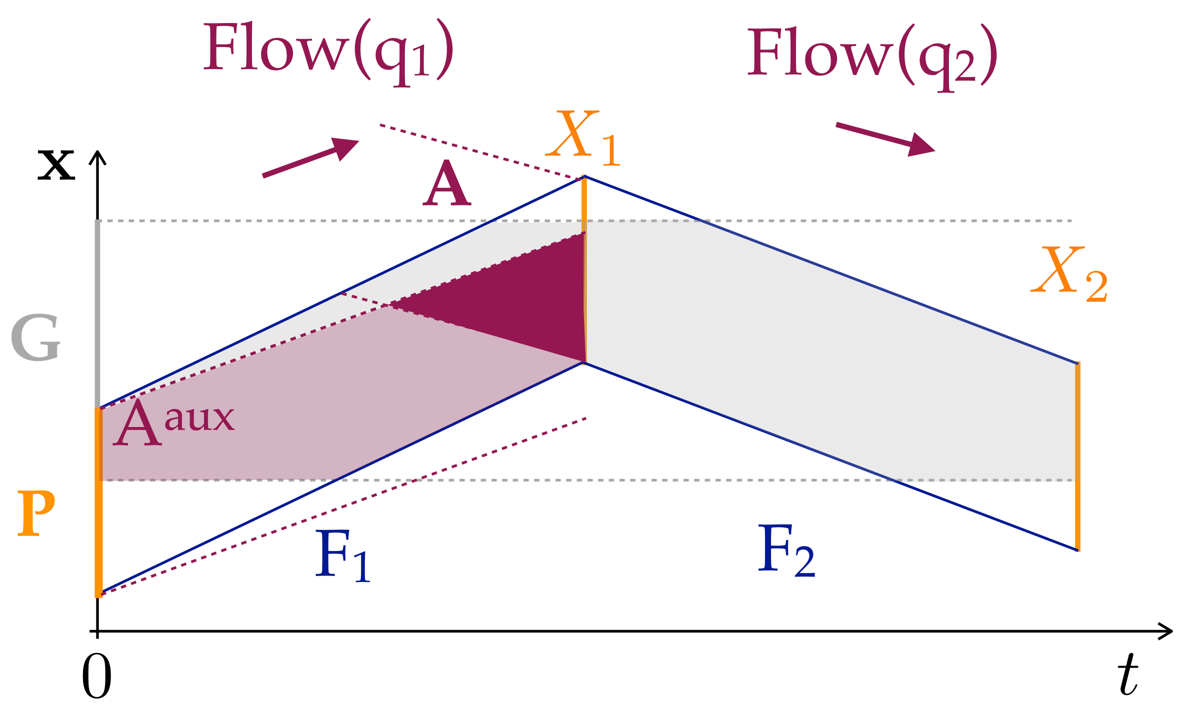
$A = PRE\ (X_1, A^{aux}, Flow(q_2))$




$B^{aux} = PRE\ (X_1, P, Flow(q_1))$

$B = POST\ (B^{aux}, G_2, Flow(q_1))$

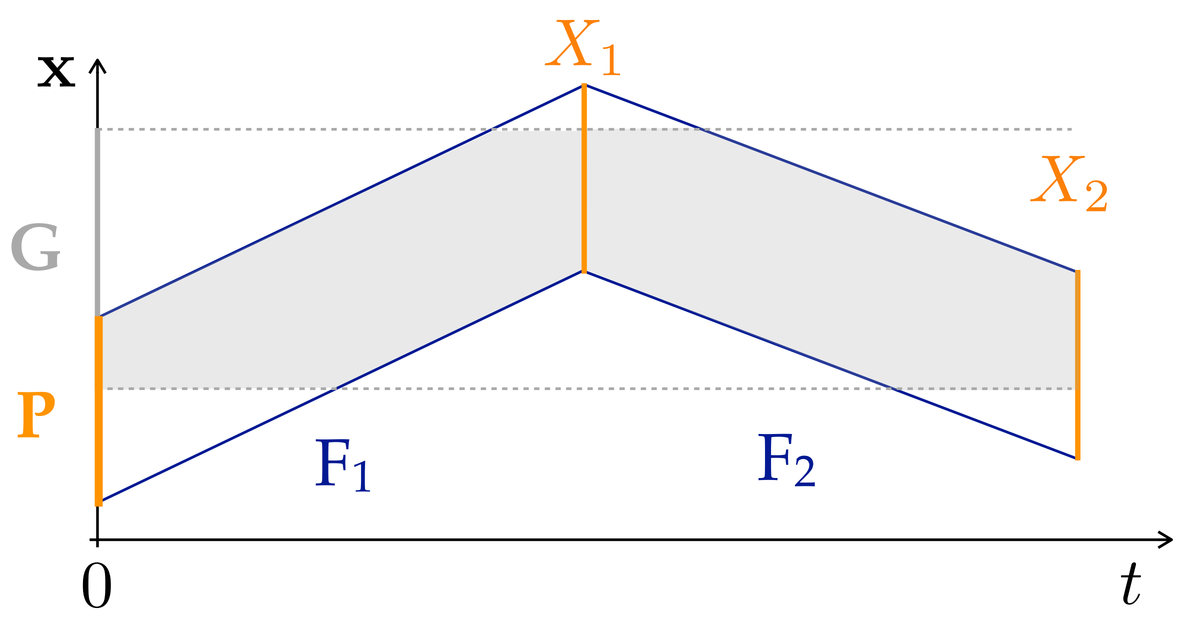
# Reachability computation




$$A^{aux} = \text{POST}\ (P, G_1, \text{Flow}(q_1))$$

$$A = \text{PRE}\ (X_1, A^{aux}, \text{Flow}(q_2))$$

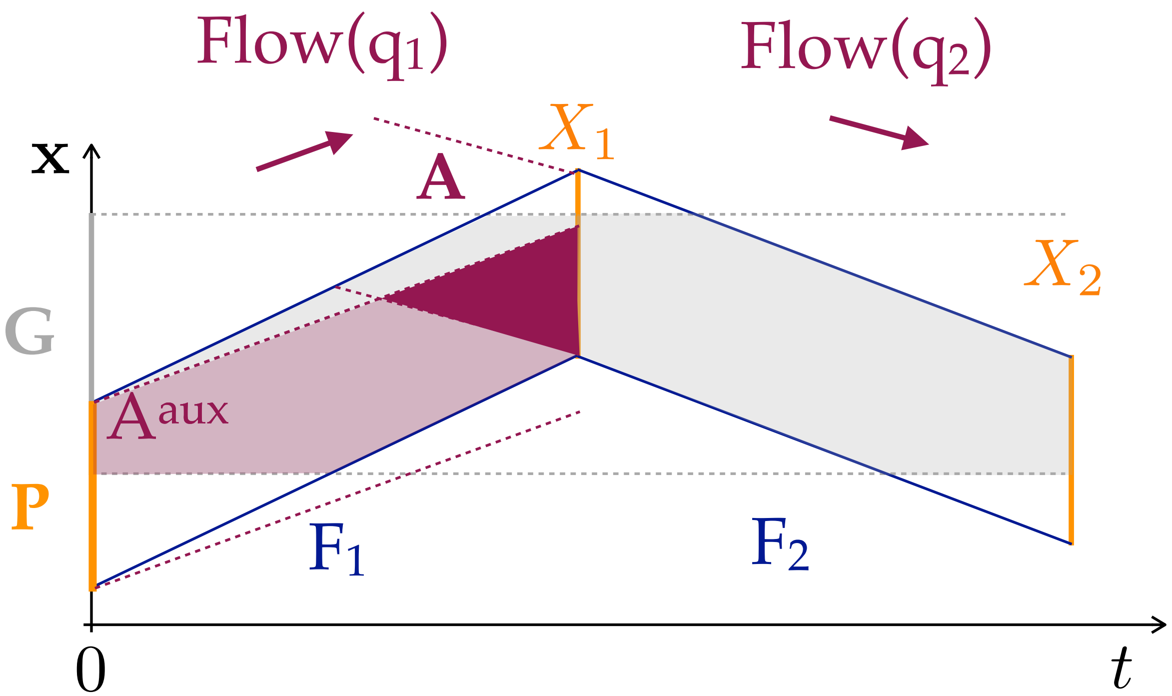
$G_1 = F_1 \cap G$ $\quad$ $G_2 = F_2 \cap G$




$$B^{aux} = \text{PRE}\ (X_1, P, \text{Flow}(q_1))$$

$$B = \text{POST}\ (B^{aux}, G_2, \text{Flow}(q_1))$$

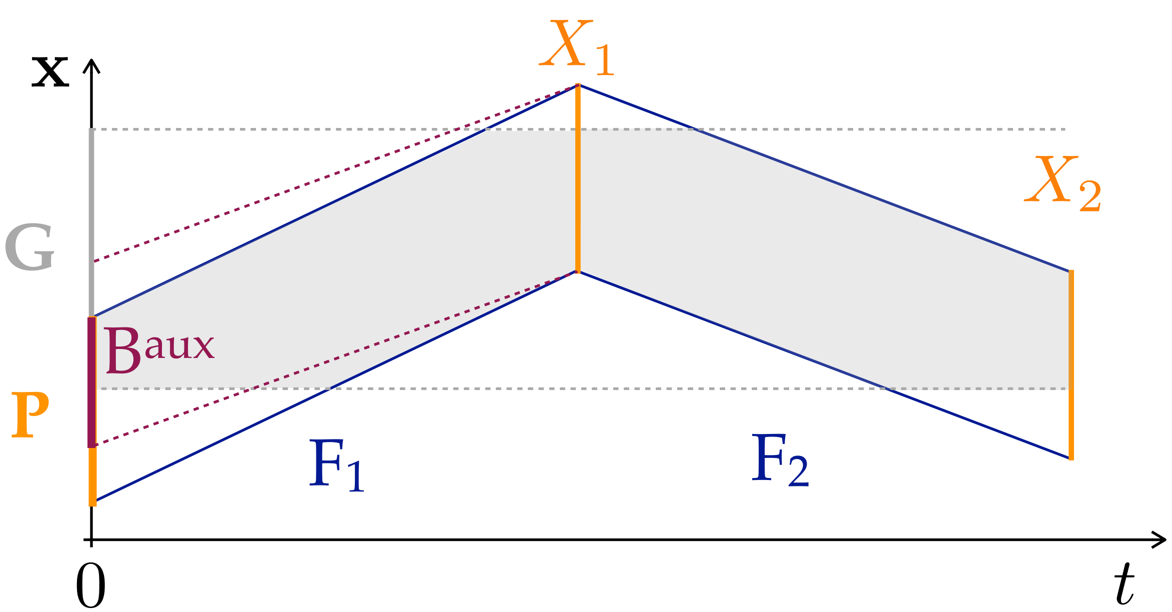
# Reachability computation




$A^{aux}$ = POST (P, $G_1$, Flow($q_1$))

A = PRE ($X_1$, $A^{aux}$, Flow($q_2$))

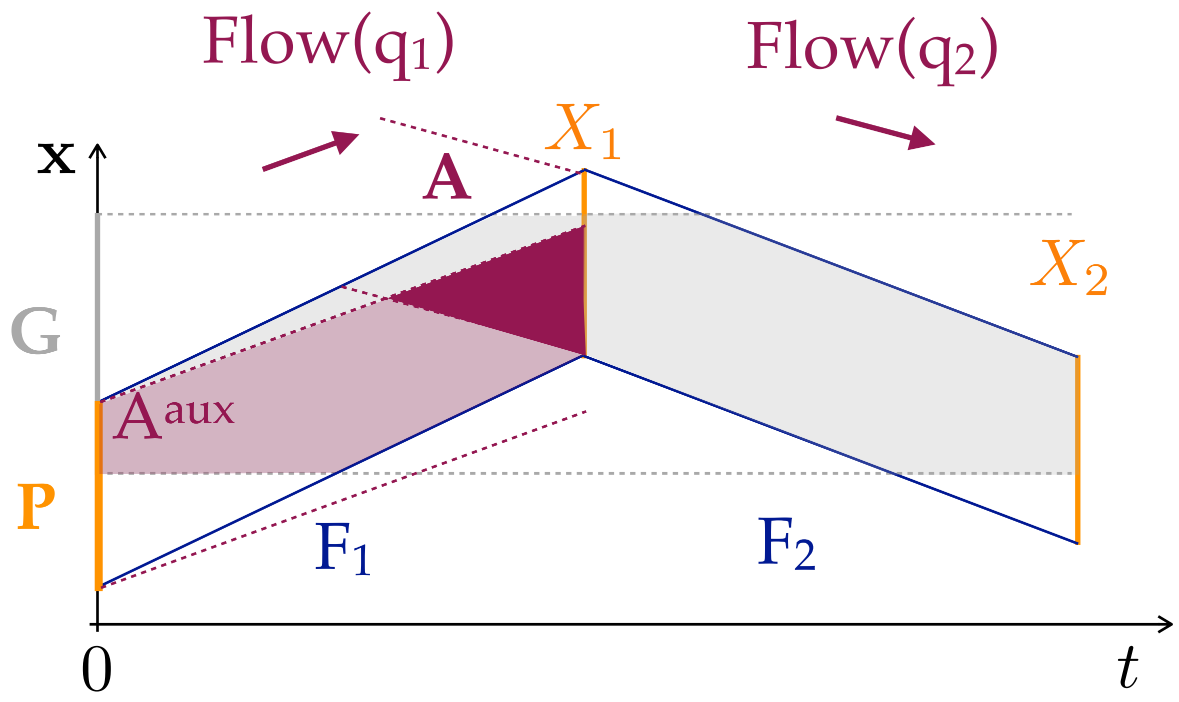
$G_1 = F_1 \cap G$ $G_2 = F_2 \cap G$




$B^{aux}$ = PRE ($X_1$, P, Flow($q_1$))

B = POST ($B^{aux}$, $G_2$, Flow($q_1$))

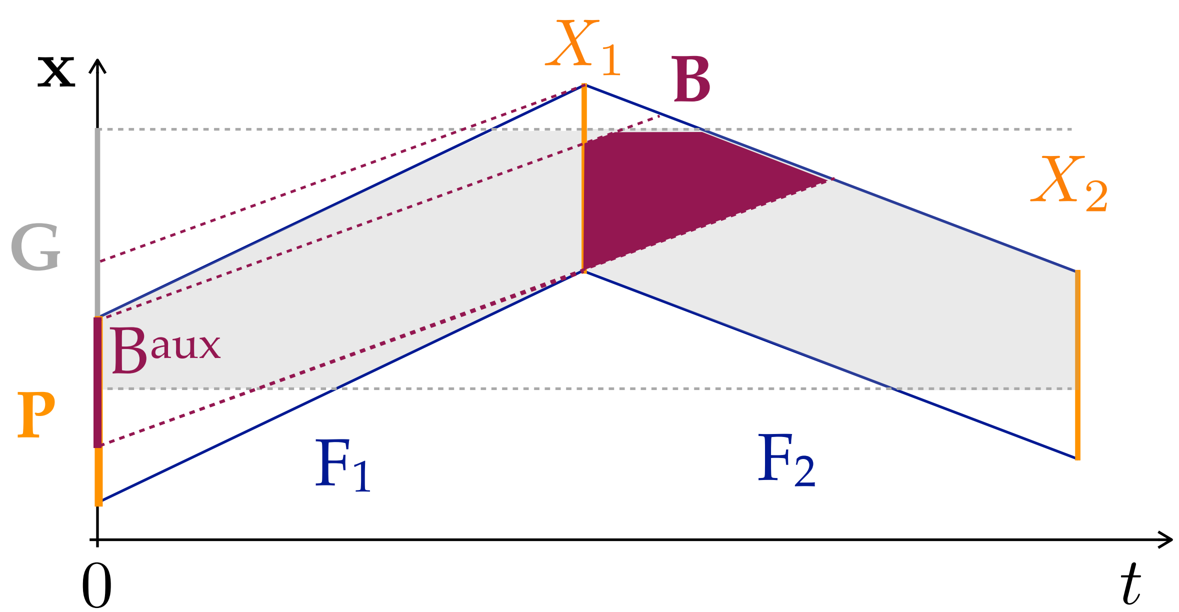
# Reachability computation




$A^{aux} = \text{POST}\ (P, G_1, \text{Flow}(q_1))$

$A = \text{PRE}\ (X_1, A^{aux}, \text{Flow}(q_2))$

$G_1 = F_1 \cap G \qquad G_2 = F_2 \cap G$

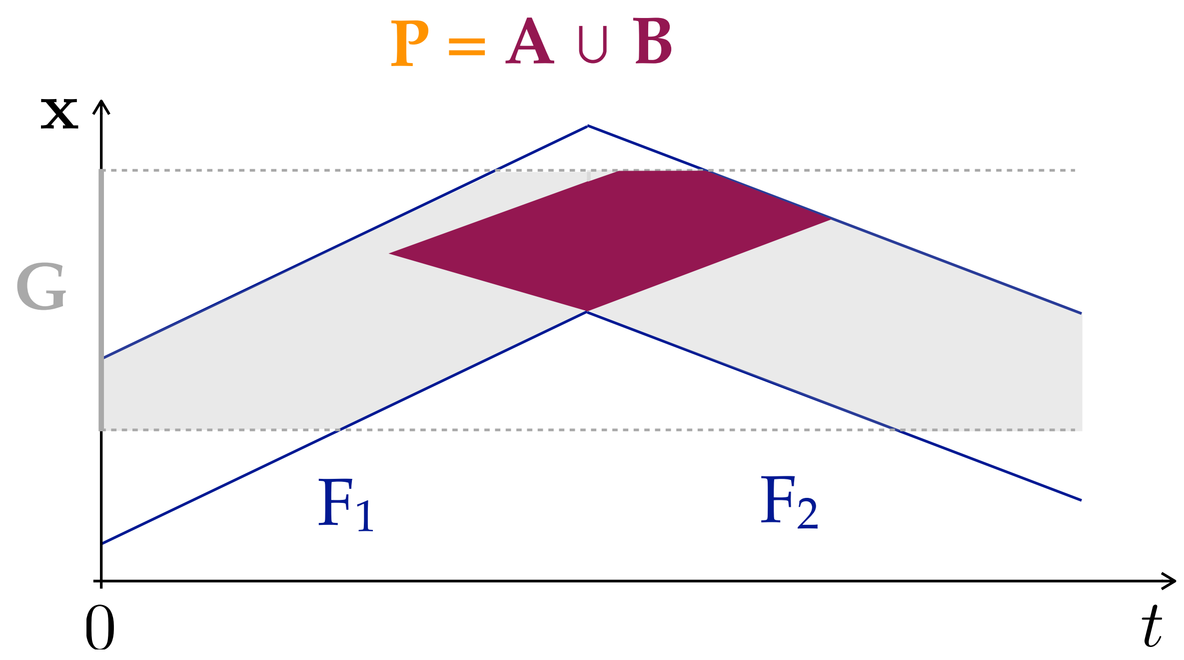



$B^{aux} = \text{PRE}\ (X_1, P, \text{Flow}(q_1))$

$B = \text{POST}\ (B^{aux}, G_2, \text{Flow}(q_1))$

# Reachability computation

**Reachable switching set**



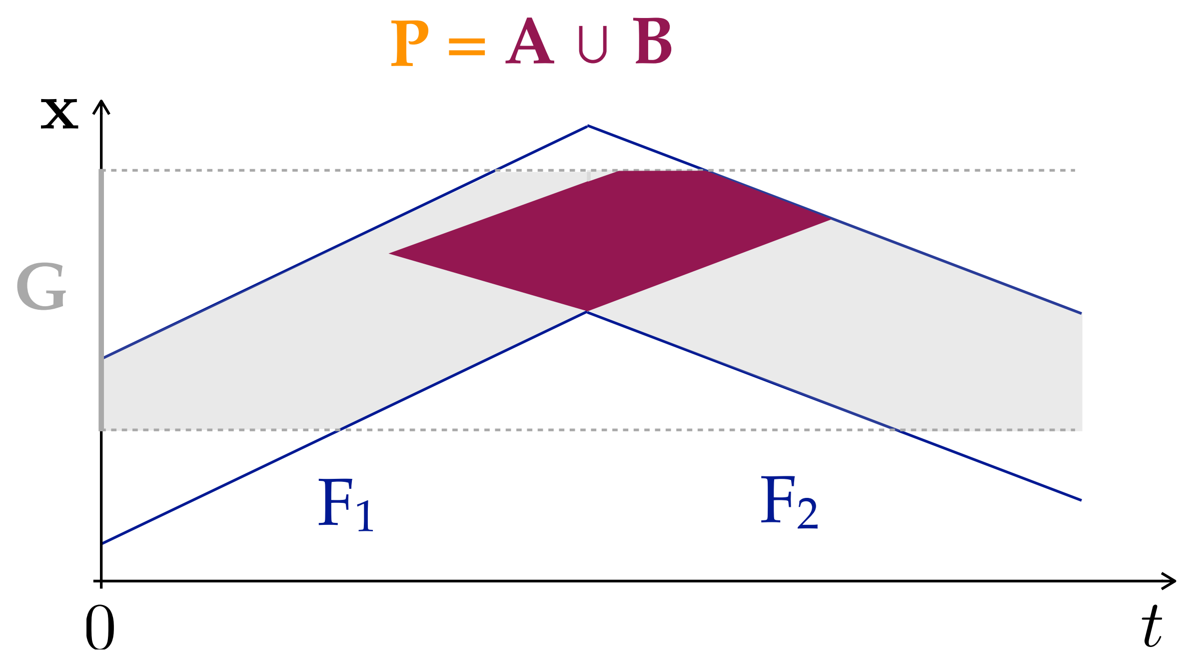

- Compute iteratively the reachable switching set P until the last ε-tube piece:

Last $\mathbf{P} \neq \varnothing \Rightarrow$ **membership True**

Last $\mathbf{P} = \varnothing \Rightarrow$ **membership False**

# Reachability computation

**Reachable switching set**




- Compute iteratively the reachable switching set P until the last ε-tube piece:

Last $\mathbf{P} \neq \varnothing \Rightarrow$ **membership True**

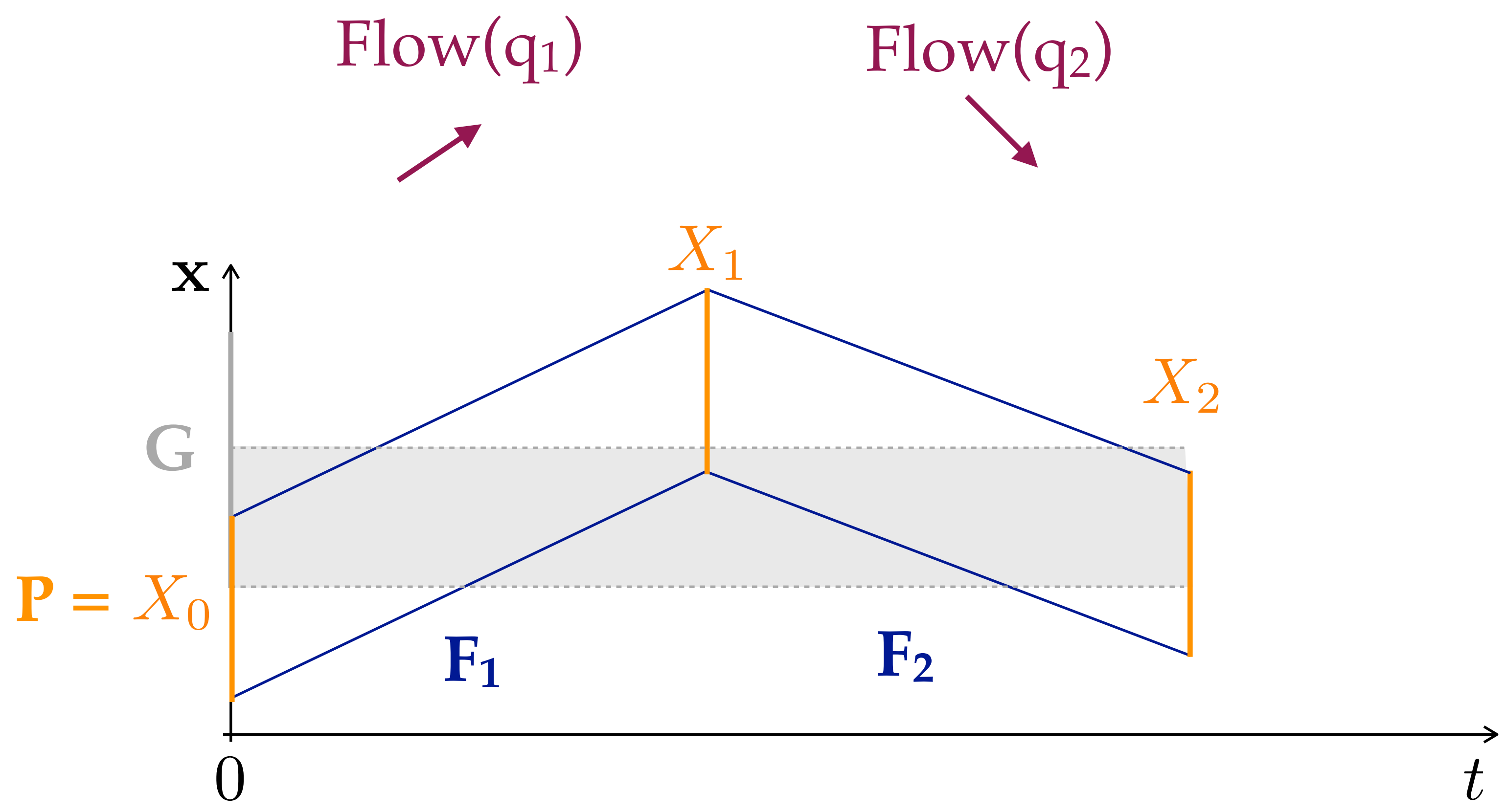
Last $\mathbf{P} = \varnothing \Rightarrow$ **membership False**

*If membership of f in H is False, then relax the LHA model.*

# Relaxation

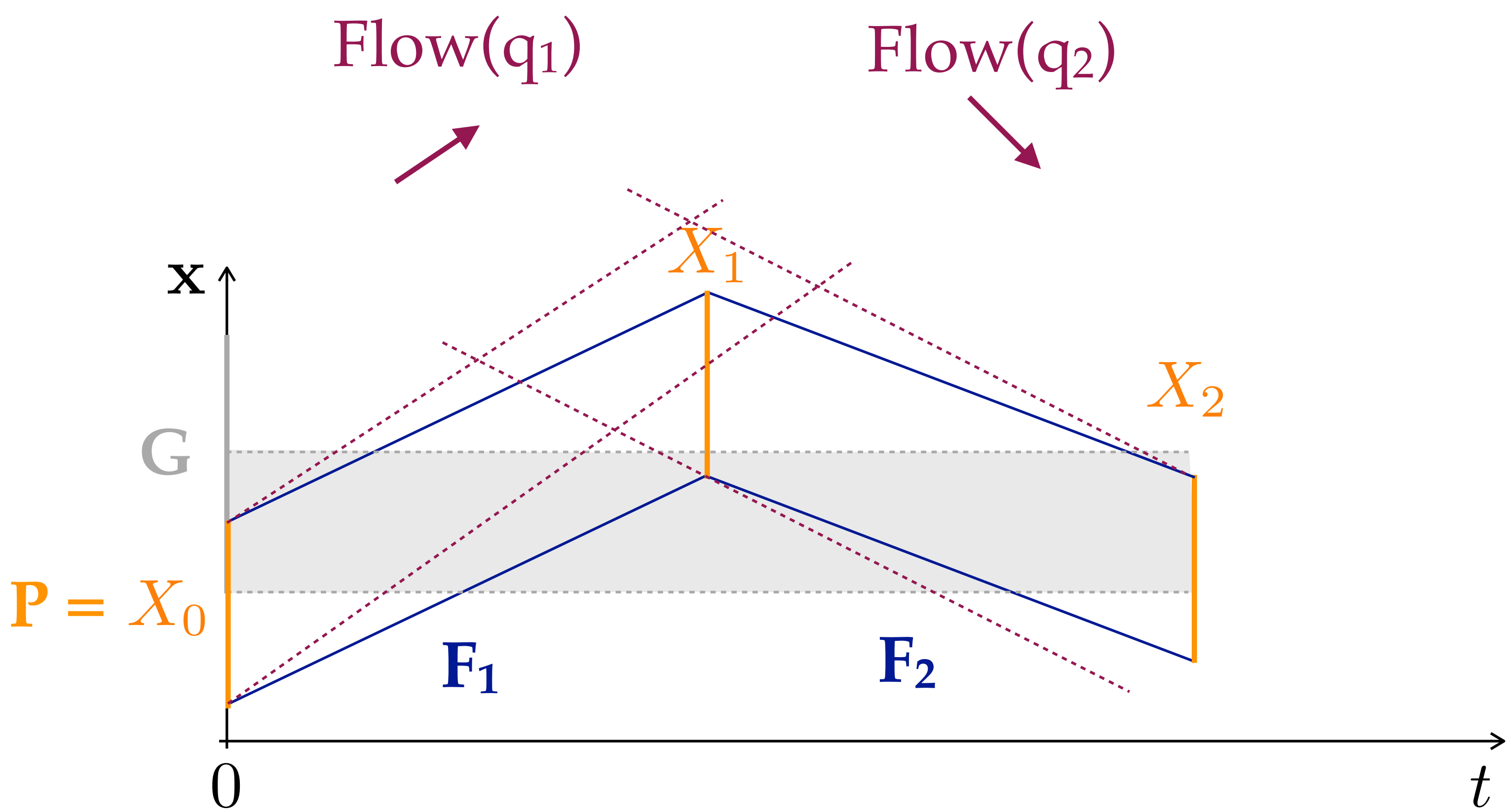
# Relaxation problem

Given an LHA $H$ and a m-PWL function $f$, does there exist an execution $\sigma$ in $H$ consistent with $f$ and such that $d(f, \sigma) \leq \varepsilon$ by expanding the guards and invariants of $H$?



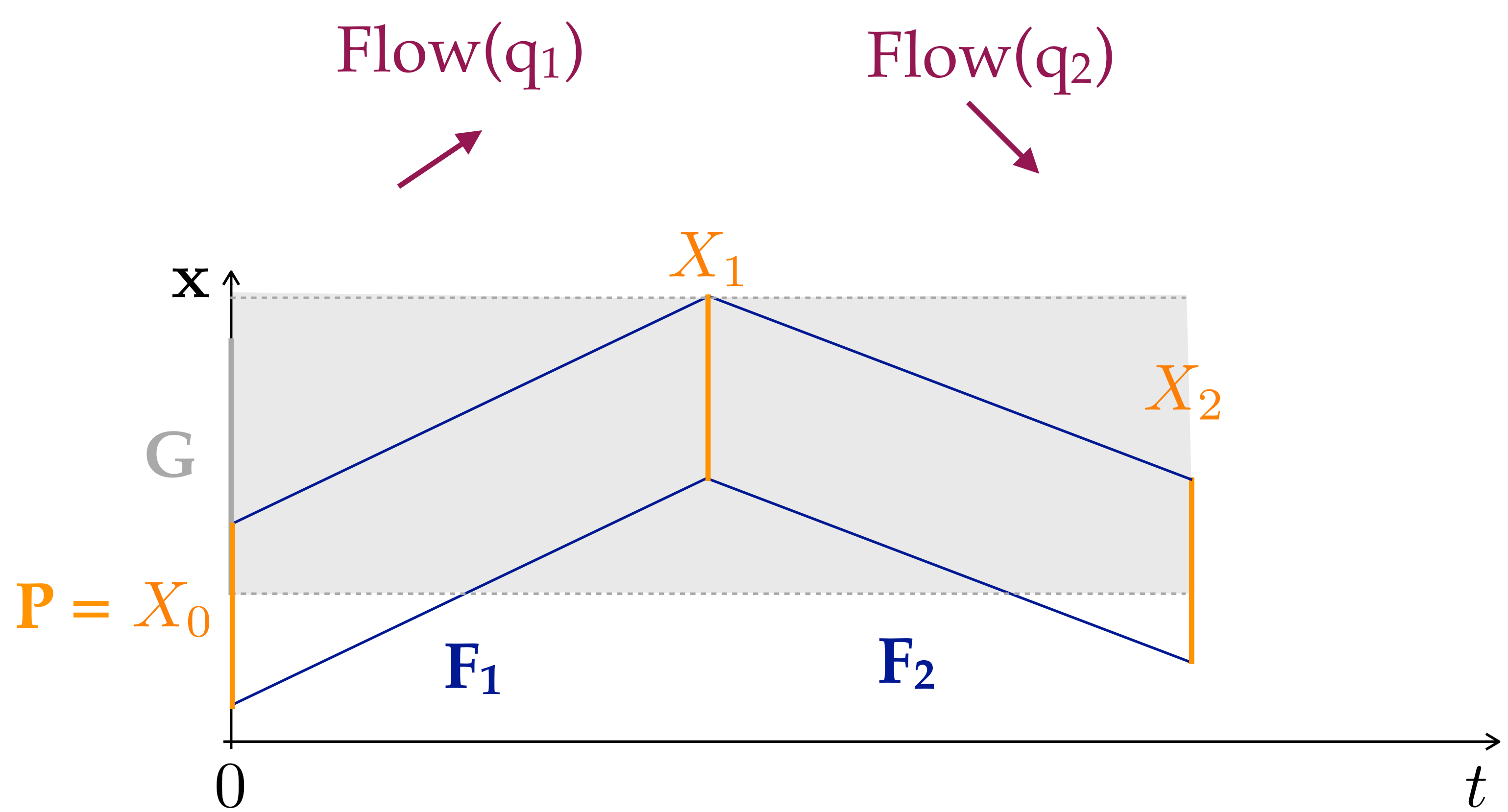
# Relaxation problem

Given an LHA $H$ and a m-PWL function $f$, does there exist an execution $\sigma$ in $H$ consistent with $f$ and such that $d(f, \sigma) \leq \varepsilon$ by expanding the guards and invariants of $H$?



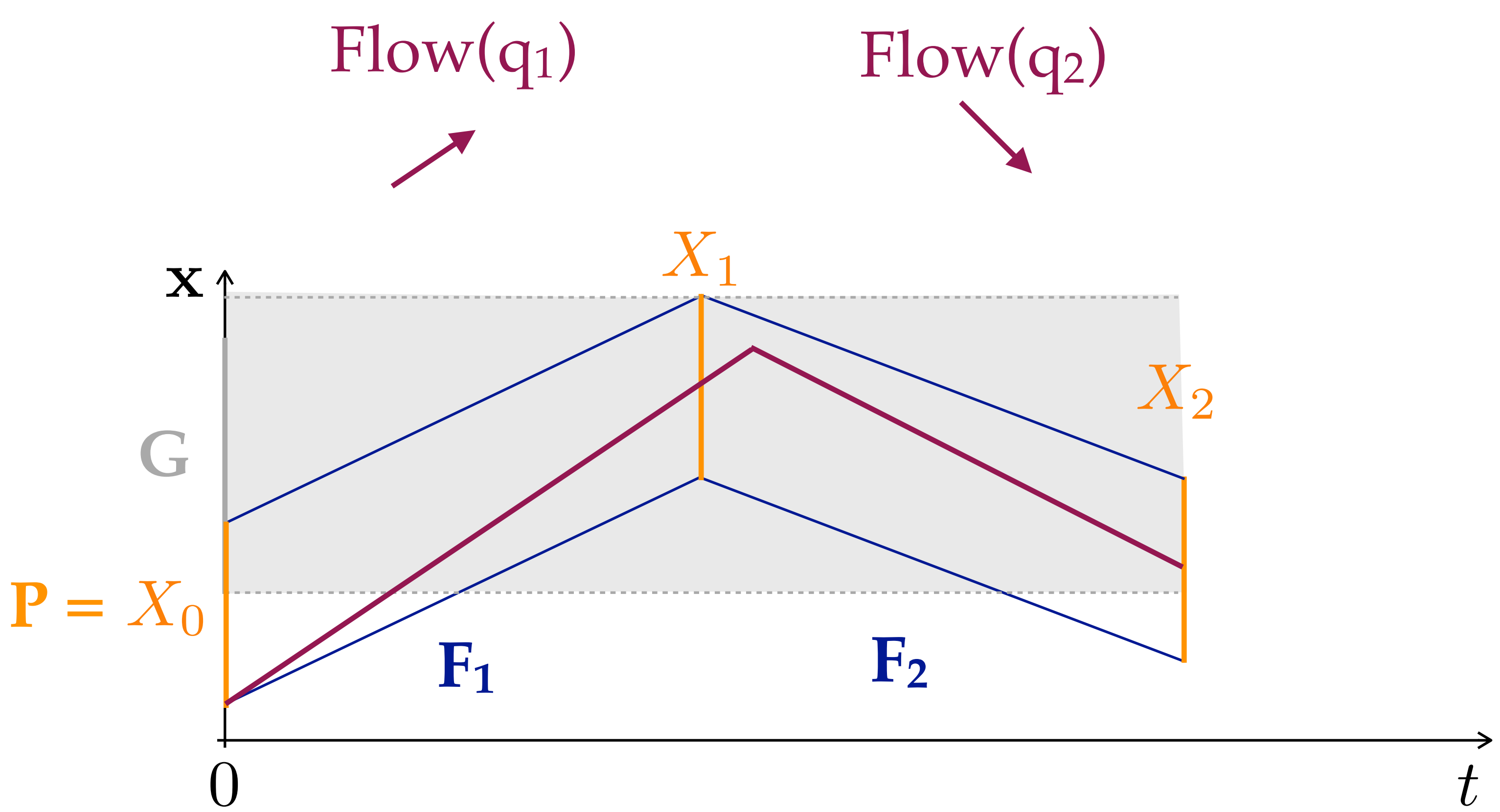
# Relaxation problem

Given an LHA $H$ and a m-PWL function $f$, does there exist an execution $\sigma$ in $H$ consistent with $f$ and such that $d(f, \sigma) \leq \varepsilon$ by expanding the guards and invariants of $H$?



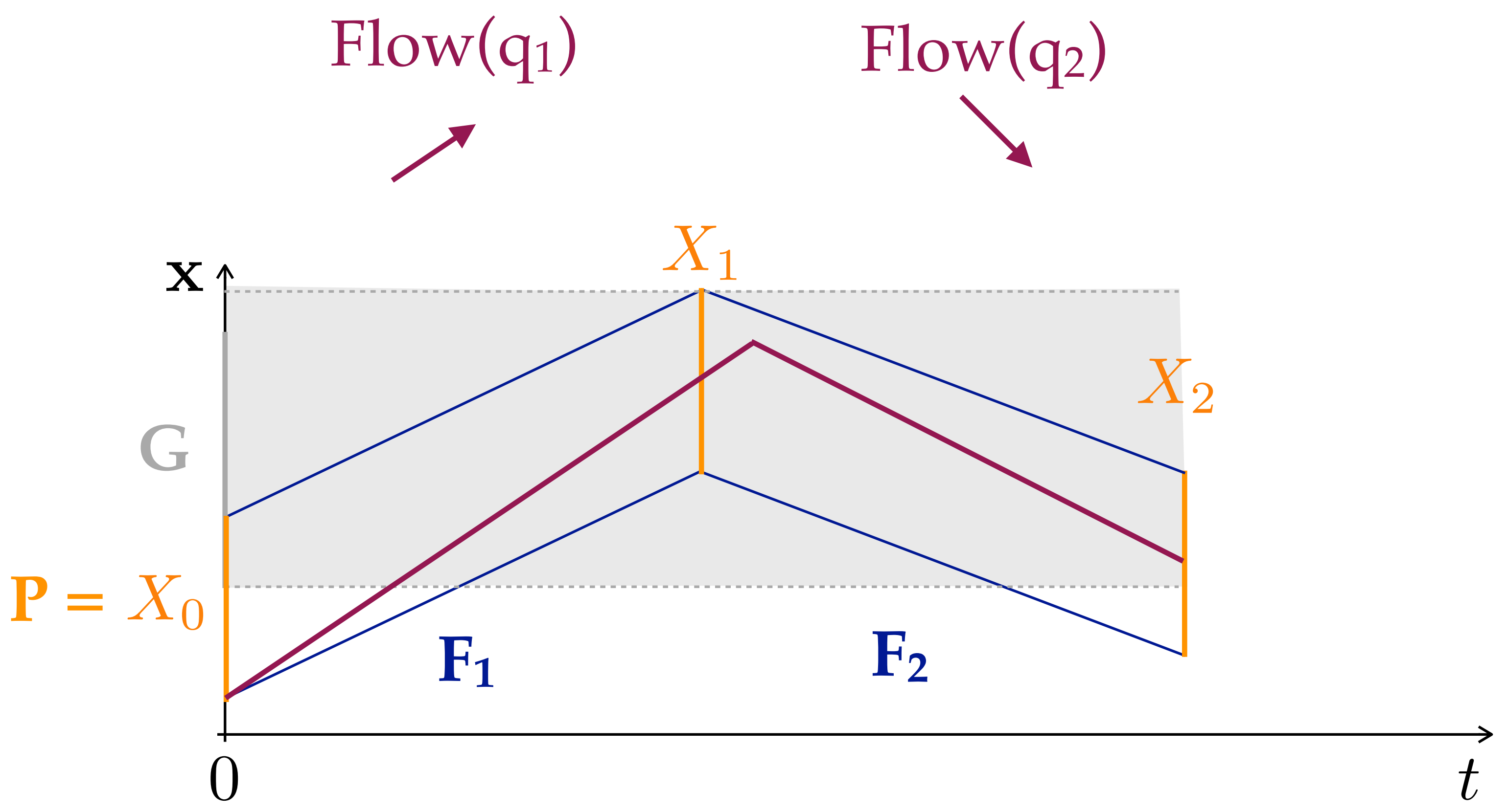
# Relaxation problem

Given an LHA $H$ and a m-PWL function $f$, does there exist an execution σ in $H$ consistent with $f$ and such that $d(f, \sigma) \leq \varepsilon$ by expanding the guards and invariants of $H$?



# Relaxation problem

Given an LHA $H$ and a m-PWL function $f$, does there exist an execution $\sigma$ in $H$ consistent with $f$ and such that $d(f, \sigma) \leq \varepsilon$ by expanding the guards and invariants of $H$?




*If relaxation of H fails, then modify the discrete structure of the LHA model.*

# Adaptation

# Adaptation problem

Given an LHA $H$, a PWL function $f$ and a path $\pi$ in $H$, construct a path $\pi'$ by preserving locations in $\pi$ such that there exists an execution $\sigma$ in $H$ for $\pi'$ with $d(f, \sigma) \leq \varepsilon$

**Adaptation procedure**

- Recall $f \equiv p_1, \ldots, p_m$ and $\pi \equiv q_1, \ldots, q_m$
- Construct $\pi' = \pi$
- Compute iteratively the reachable switching sets $P_i$ until emptiness
- If $P_i = \varnothing$, then:
    1. Replace $q_i$ by $q'_i$ in $\pi'$
    2. Replace $q_{i-1}$ by $q'_{i-1}$ in $\pi'$
    3. From $q_{i-2}$ to $q_1$
    4. Until $P_i \neq \varnothing$

# Adaptation problem

Given an LHA *H*, a PWL function *f* and a path $\pi$ in *H*, construct a path $\pi'$ by preserving locations in $\pi$ such that there exists an execution $\sigma$ in *H* for $\pi'$ with $d(f, \sigma) \leq \varepsilon$

**Adaptation procedure**

- Recall $f \equiv p_1, \ldots, p_m$ and $\pi \equiv q_1, \ldots, q_m$
- Construct $\pi' = \pi$
- Compute iteratively the reachable switching sets $P_i$ until emptiness
- If $P_i = \emptyset$, then:
  1. Replace $q_i$ by $q'_i$ in $\pi'$
  2. Replace $q_{i-1}$ by $q'_{i-1}$ in $\pi'$
  3. From $q_{i-2}$ to $q_1$
  4. Until $P_i \neq \emptyset$

**Replacement** consists of 2 different strategies:

1. Existing $q'_i$ with $Flow(q'_i) \cong slope(p_i)$
2. New $q'_i$ with $Flow(q'_i) = slope(p_i)$

# Adaptation problem

Given an LHA $H$, a PWL function $f$ and a path $\pi$ in $H$, construct a path $\pi'$ by preserving locations in $\pi$ such that there exists an execution $\sigma$ in $H$ for $\pi'$ with $d(f, \sigma) \leq \varepsilon$

**Adaptation procedure**

- Recall $f \equiv p_1, \ldots, p_m$ and $\pi \equiv q_1, \ldots, q_m$
- Construct $\pi' = \pi$
- Compute iteratively the reachable switching sets $P_i$ until emptiness
- If $P_i = \emptyset$, then:
  1. Replace $q_i$ by $q'_i$ in $\pi'$
  2. Replace $q_{i-1}$ by $q'_{i-1}$ in $\pi'$
  3. From $q_{i-2}$ to $q_1$
  4. Until $P_i \neq \emptyset$

**Replacement** consists of 2 different strategies:

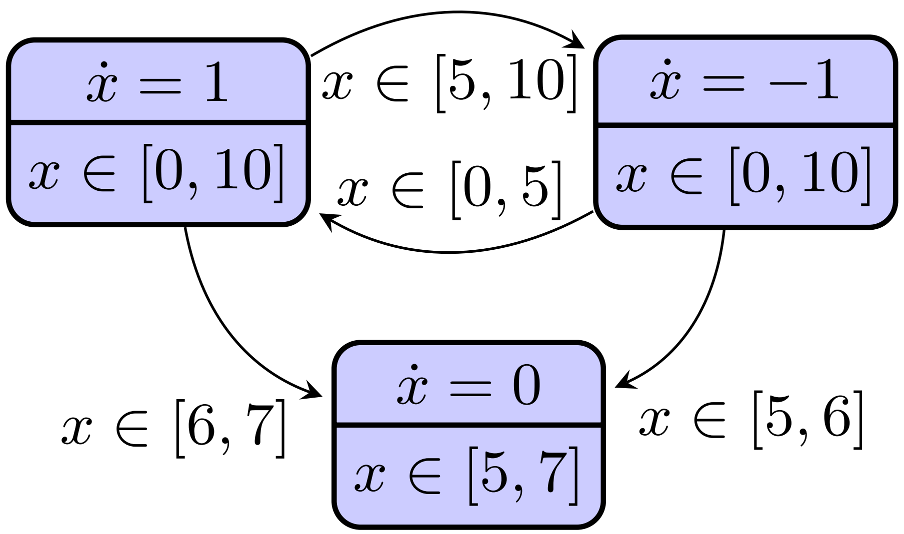
1. Existing $q'_i$ with $Flow(q'_i) \cong slope(p_i)$
2. New $q'_i$ with $Flow(q'_i) = slope(p_i)$

*Modify the LHA $H$ to include the location path $\pi'$.*

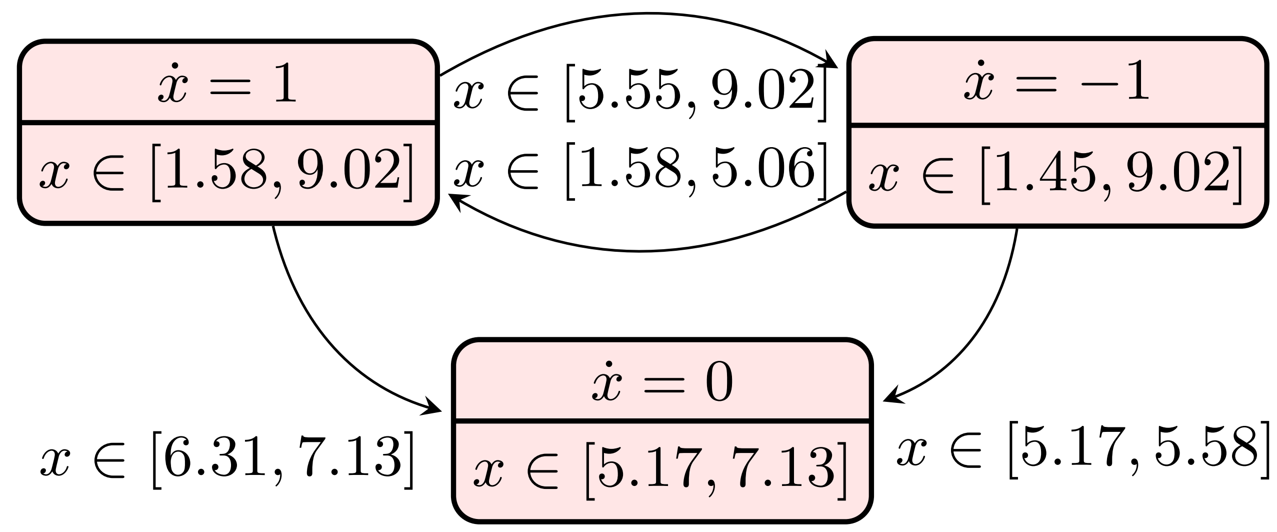
# Experiments

# Synthetic data

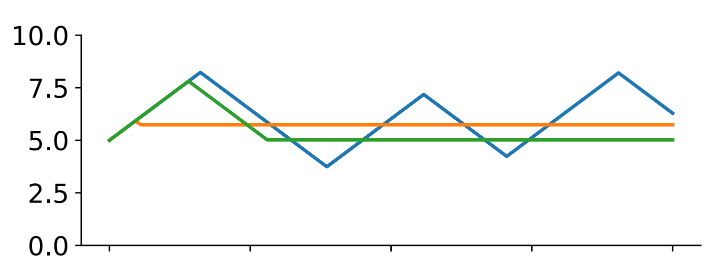
Original LHA



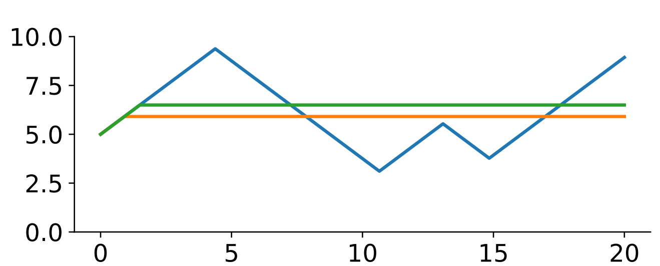

Synthesized LHA after 10 iterations



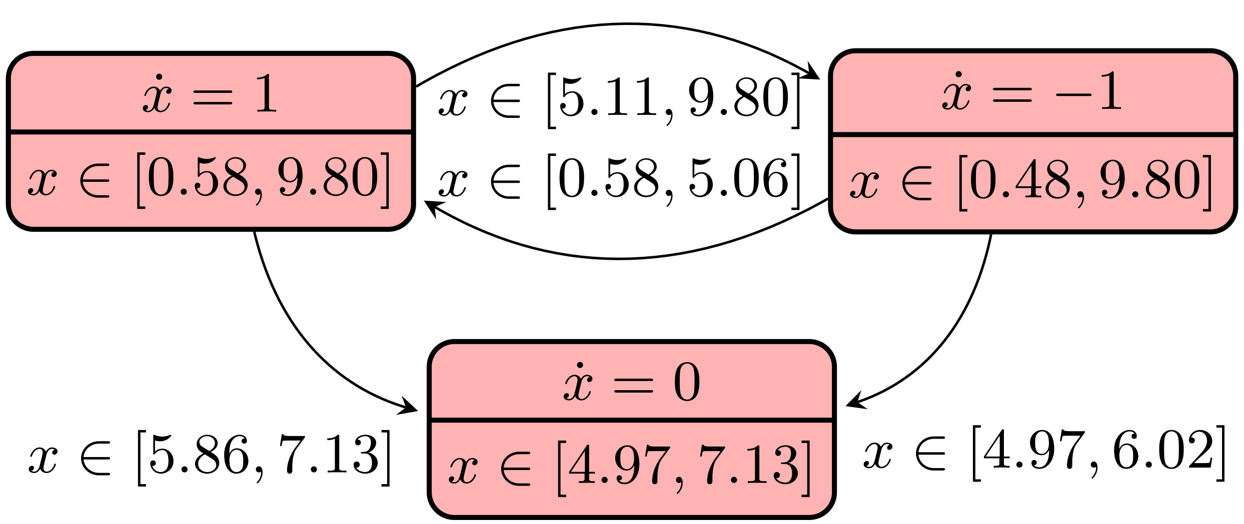

Executions of original LHA




Executions of synthesized LHA

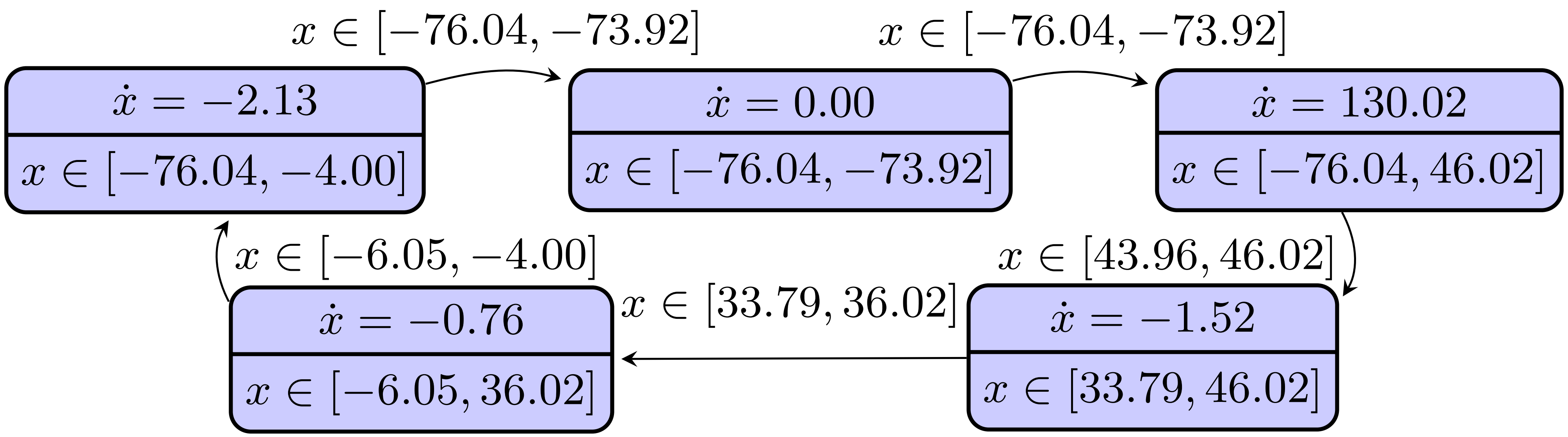



Synthesized LHA after 100 iterations

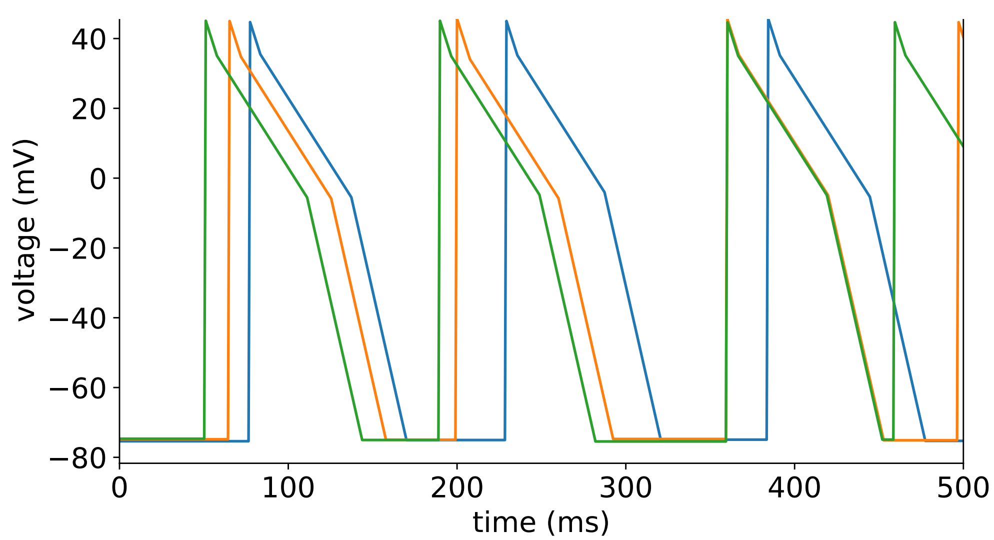


# Cell model

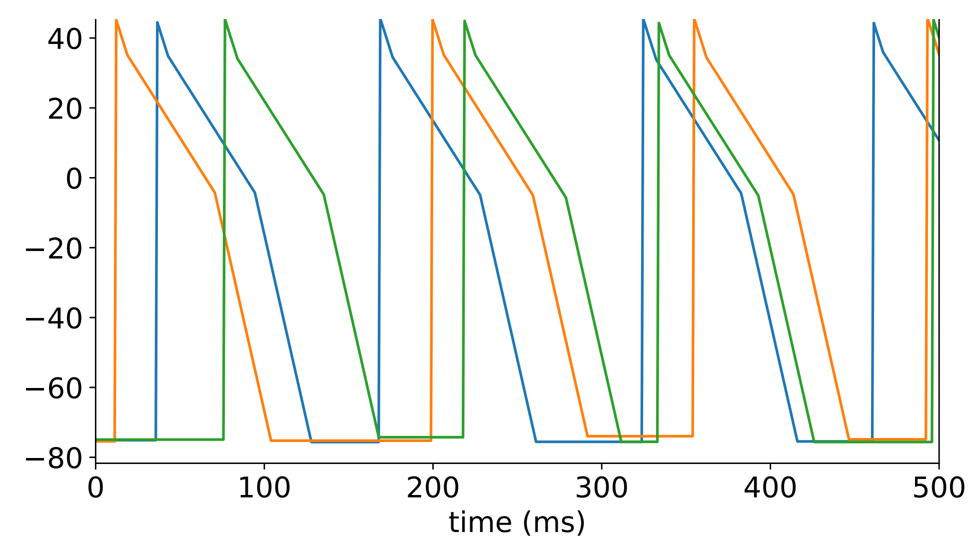
Synthesized LHA




Input PWL functions




Simulations of the synthesized LHA



# Conclusion

# Summary

- Development of **two automatic approaches for synthesizing** linear hybrid automata from experimental data
- **Nondeterministic** guards and invariants
- **Arbitrary** number and topology of **locations**
- The synthesized automaton reproduce the data up to a **tolerance**
- Both algorithms well-suited for **online and synthesis-in-the-loop applications**
- **SMT-based approach minimizes** the size of the **model** but it is feasible for limited data sets
- **Membership-based approach trades-off** between **size and precision** of the model

# Questions?