Using The Star Set Representation For The Analysis Of Stochastic Hybrid Automata

Defense Talk

Sinan David Lindemann
Examiner: Prof. Ábrahám

LuFG Theory of Hybrid Systems

02.02.2026

RWTH AACHEN UNIVERSITY

Contents

Rectangular Automata - Definition

Definition Rectangular Automata

$$\mathcal{H}_{\mathcal{R}} = (Loc\ , Var\ , Lab\ , Flow\ , Inv\ , Edge\ , Init)$$

- *Loc* - a finite set of locations.
- *Var* - a vector of variables $Var = (x_0, \ldots, x_{d-1})$
- *Lab* - a finite set of labels.
- $Flow : Loc \rightarrow Pred^{R}_{Var \cup \dot{Var}}$ - specifies the continuous evolution.
- $Inv : Loc \rightarrow Pred^{R}_{Var}$ - specifies the Invariant for each location.
- $Edge \subseteq Loc \times Lab \times Pred^{R}_{Var} \times Pred^{R}_{Var, Var'} \times Loc$ - defines the transitions or jumps ($(l, a, g, r, l') \in Edge$).
- $Init : Loc \rightarrow Pred^{R}_{Var}$ - defines the initial states.

Hybrid Automata - Operational Semantics Flow

Rule For The Flow

$$\frac{\begin{array}{c} l \in Loc \quad v, v' \in \mathbb{R}^d \quad f : [0, \tau] \to \mathbb{R}^d \\ \partial f / \partial t = \dot{f} : (0, \tau) \to \mathbb{R}^d \quad f(0) = v \quad f(\tau) = v' \\ \forall \epsilon \in (0, \tau). f(\epsilon), \dot{f}(\epsilon) \models Flow(l) \\ \forall \epsilon \in [0, \tau]. f(\epsilon) \models Inv(l) \end{array}}{(l, v) \xrightarrow{\tau} (l, v')}$$

Hybrid Automata - Operational Semantics Jump

Rule For The Jump

$$\frac{e = (l, a, g, r, l') \in Edge \quad l, l' \in Loc \quad v, v' \in \mathbb{R}^d \qquad v \models g \quad v, v' \models r \quad v' \models Inv(l')}{(l, v) \xrightarrow{e} (l', v')}$$

Hybrid Automata - Path

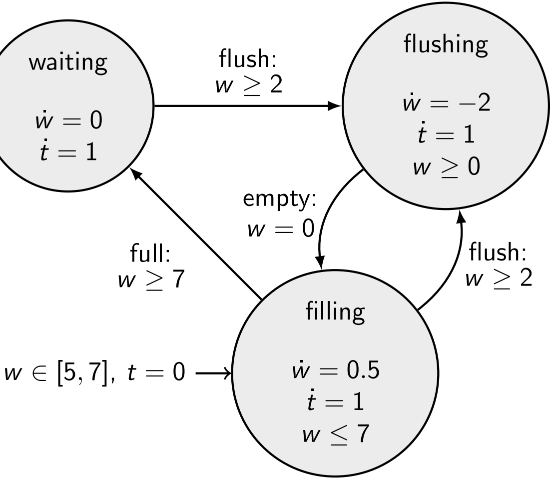
Definition Path

A *path* π of $\mathcal{H}$ is a sequence of states of $\mathcal{H}$ that are connected through alternating flow and jump steps:

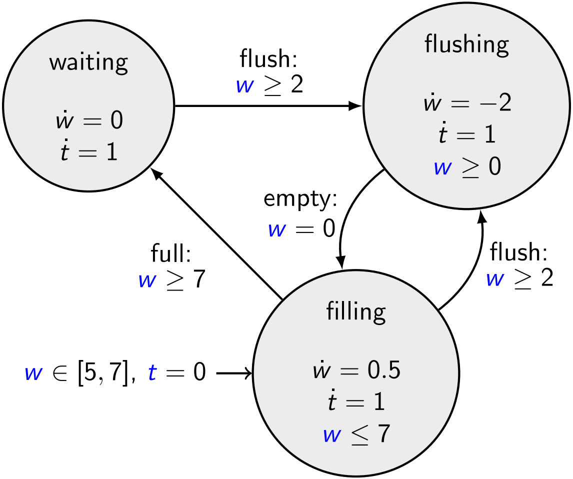
$$\pi = \sigma_0 \xrightarrow{\tau_0} \sigma_1 \xrightarrow{e_1} \sigma_2 \xrightarrow{\tau_2} \sigma_3 \xrightarrow{e_3} \dots$$

with $\sigma_i = (l_i, v_i) \in Loc \times \mathbb{R}^d$ being states of $\mathcal{H}$, $\tau_i \in \mathbb{R} \geq 0, e_i \in Edge$, and $v_0 \models Inv(l_0)$. A path is *initial* if additionally $v_0 \models Init(l_0)$.

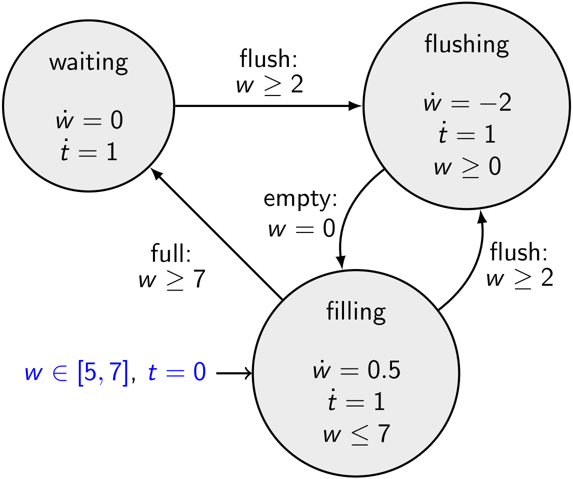
Hybrid Automata - Example



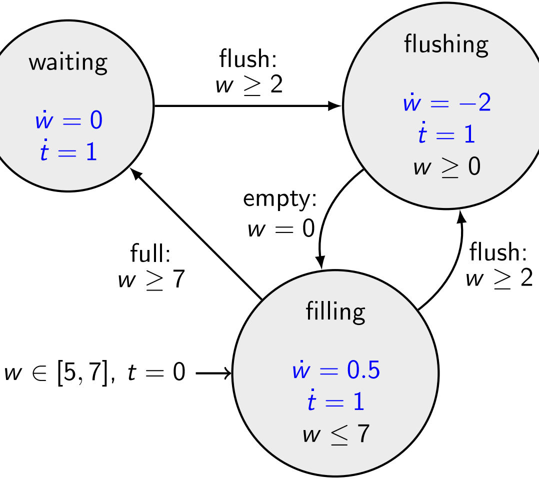
Hybrid Automata - Example



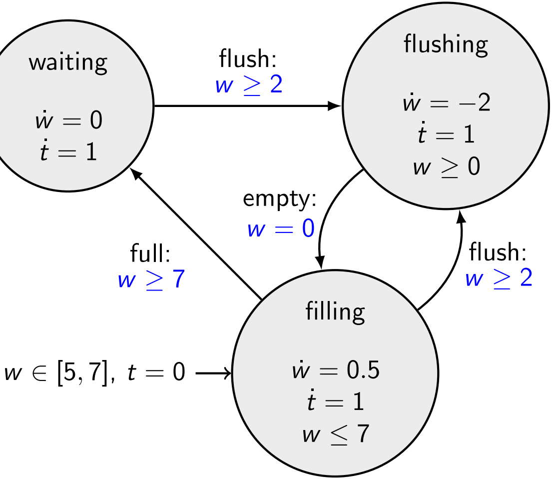
Hybrid Automata - Example



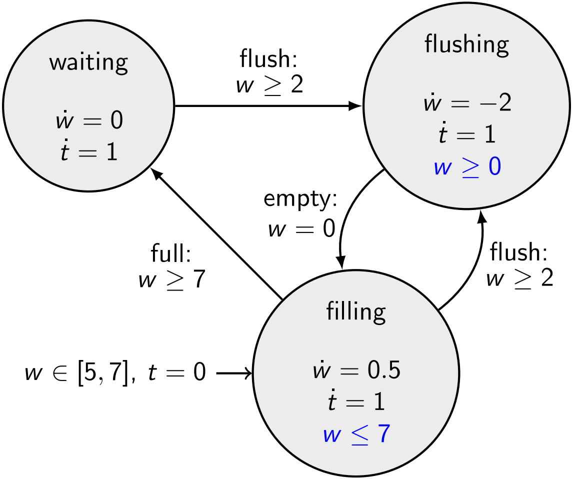
Hybrid Automata - Example



Hybrid Automata - Example



Hybrid Automata - Example



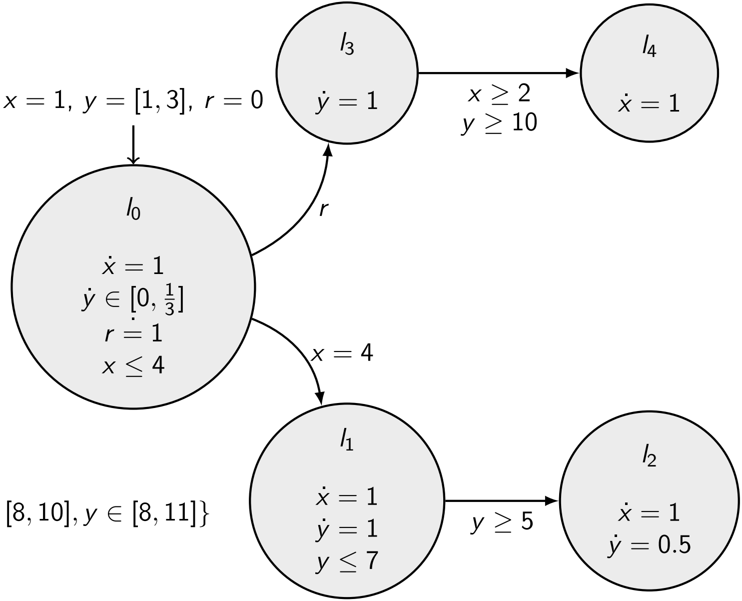
Hybrid Automata - Stochastic Hybrid Automata

Definition Stochastic Hybrid Automata

$$\mathcal{H}_{\mathcal{S}} = (\mathcal{H}_{\mathcal{R}},\ Var_R,\ Flow_R,\ Edge_R,\ Distr_R)$$

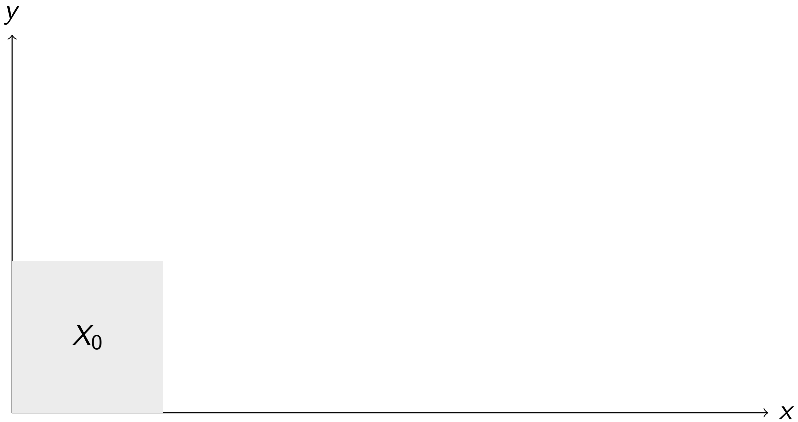
- Var_R - a finite set $Var_R = \{r_1, \ldots, r_d\}$ of random clocks.
- $Flow_R : Loc \to \{0,1\}_{Var_R}$ - specifies the flow of the random clocks.
- $Edge_R \subseteq Loc \times Var_R \times Loc$ - defines the stochastic jumps $e = (l, r, l')$.
- $Distr_R : Var_R \to \mathbb{F}_c$ - assigns a continuous probability distribution to each random clock r.

Hybrid Automata - Stochastic Hybrid Automata Example

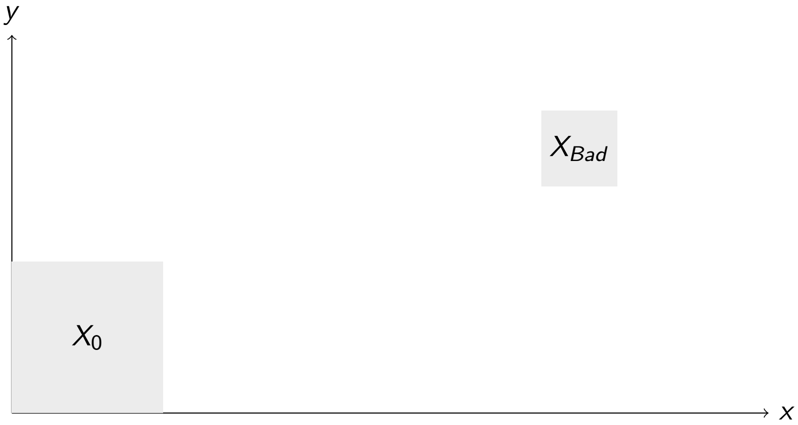


$X_{Bad} = \{l_i, x \in [8, 10], y \in [8, 11]\}$

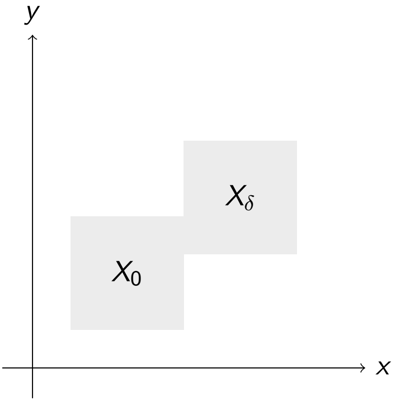
Reachability



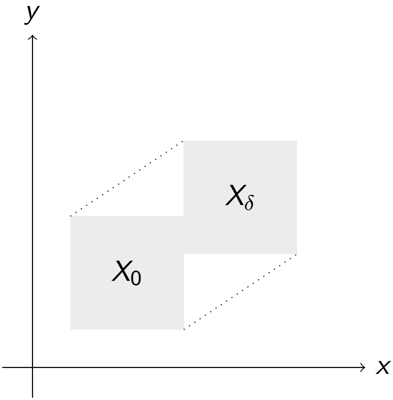
Reachability



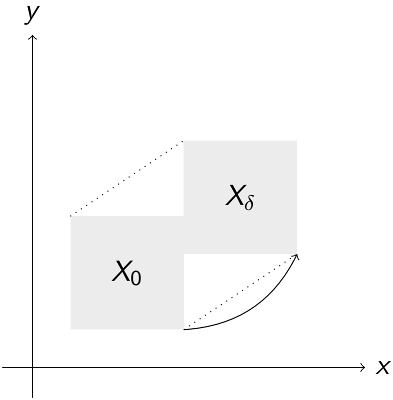
Flowpipe Construction



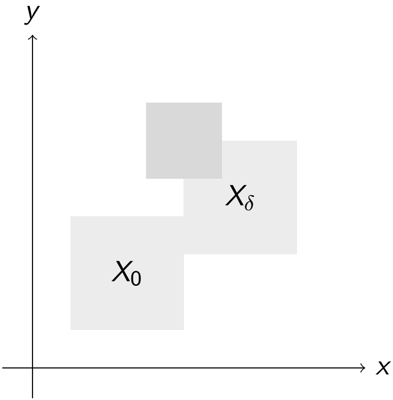
Flowpipe Construction



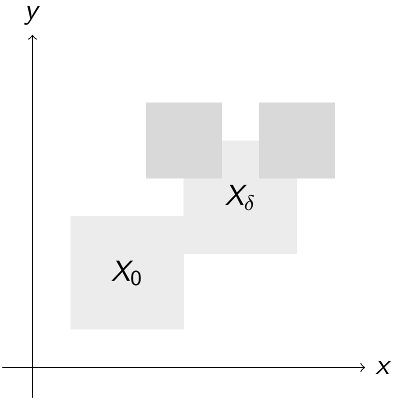
Flowpipe Construction



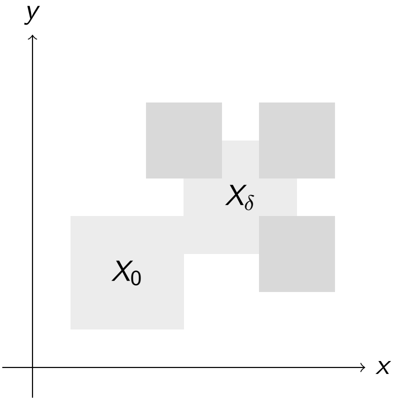
Flowpipe Construction



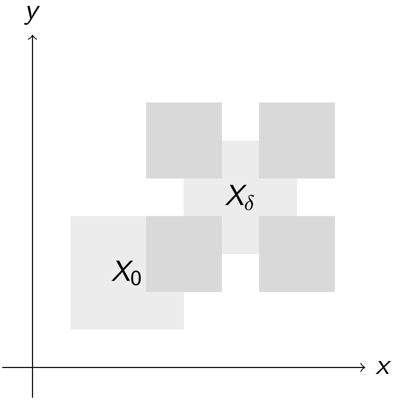
Flowpipe Construction



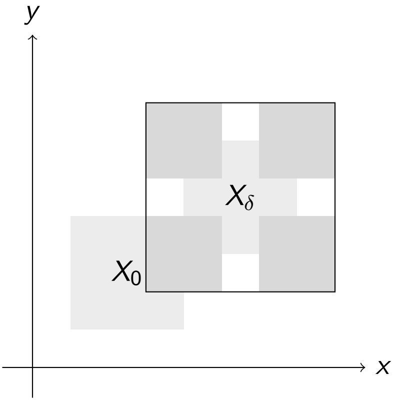
Flowpipe Construction



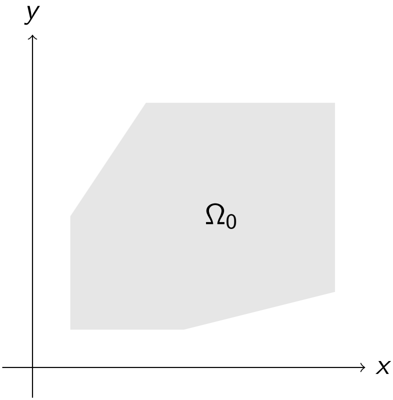
Flowpipe Construction



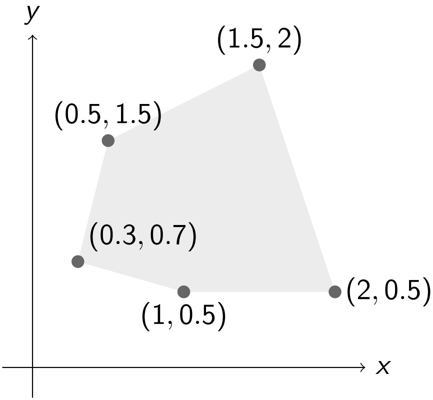
Flowpipe Construction



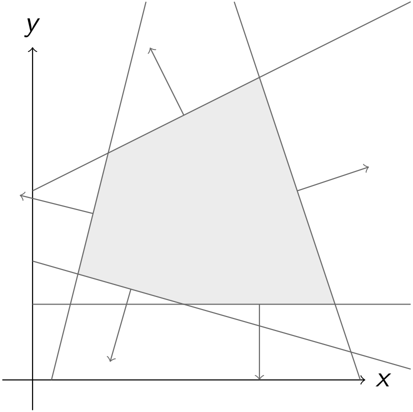
Flowpipe Construction



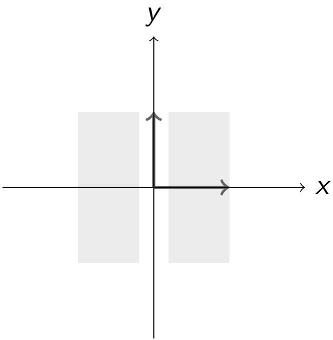
State Set Representation - VPolytopes



State Set Representation - HPolytopes



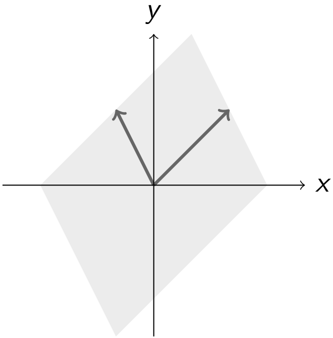
State Set Representation - Star Set



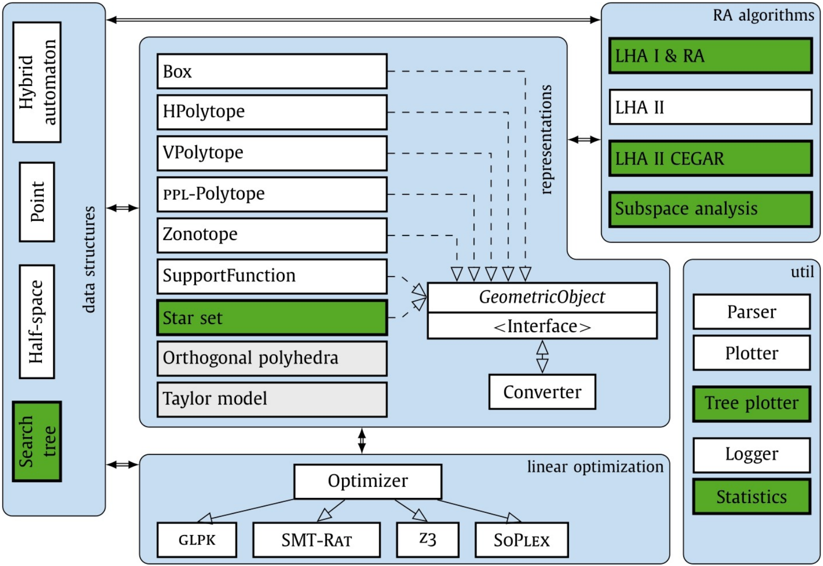
State Set Representation - Star Set Restriction

$$P = \{\alpha \in \mathbb{R}^m | C\alpha \leq d\} \text{ with } d \in \mathbb{R}^p \text{ and } C \in \mathbb{R}^{p \times m} \text{ for } p, m \in \mathbb{N}$$

State Set Representation - Star Set

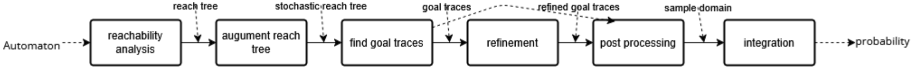


HyPro



Hybrid automaton
Point
Half-space
Search tree
data structures
Box
HPolytope
VPolytope
PPL-Polytope
Zonotope
SupportFunction
Star set
Orthogonal polyhedra
Taylor model
representations
GeometricObject
<Interface>
Converter
RA algorithms
LHA I & RA
LHA II
LHA II CEGAR
Subspace analysis
util
Parser
Plotter
Tree plotter
Logger
Statistics
linear optimization
Optimizer
GLPK
SMT-RAT
Z3
SOPLEX

RealySt


Automaton
reachability analysis
reach tree
augument reach tree
stochastic reach tree
find goal traces
goal traces
refinement
refined goal traces
post processing
sample domain
integration
probability

Implementation - RealySt

- Templating in the integration task
- Templating for stochastic reach tree nodes

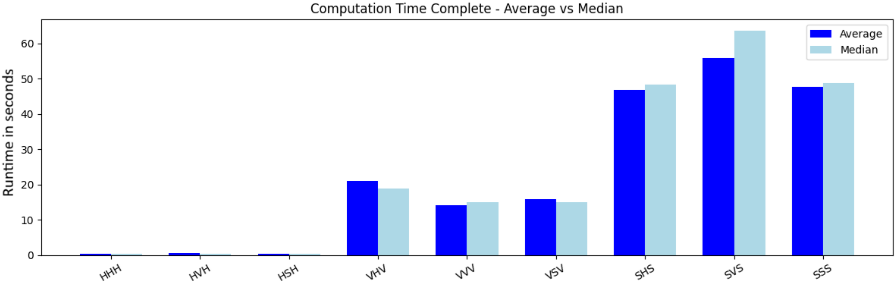
Implementation - HyPro

- Inconsistencies resolved
- Missing methods added
- Serialisation added
- Missing constructors added
- Missing conversions added
- Contains method changed

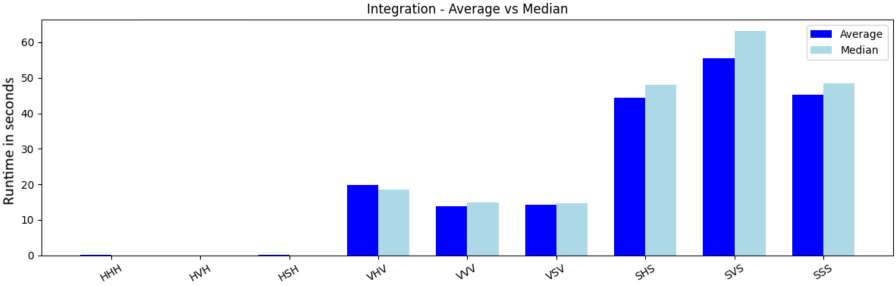
Benchmarks - Simple Running Example

Total Computation Time	HyPro HPolytope	HyPro VPolytope	HyPro Starset
RealySt HPolytope	Average: $\approx 0.35s$ Median: $\approx 0.23s$	Average: $\approx 0.60s$ Median: $\approx 0.30s$	Average: $\approx 0.32s$ Median: $\approx 0.35s$
RealySt VPolytope	Average: $\approx 20.96s$ Median: $\approx 18.73s$	Average: $\approx 14.07s$ Median: $\approx 14.99s$	Average: $\approx 15.77s$ Median: $\approx 15.04s$
RealySt Starset	Average: $\approx 46.90s$ Median: $\approx 48.29s$	Average: $\approx 55.74s$ Median: $\approx 63.55s$	Average: $\approx 47.61s$ Median: $\approx 48.74s$

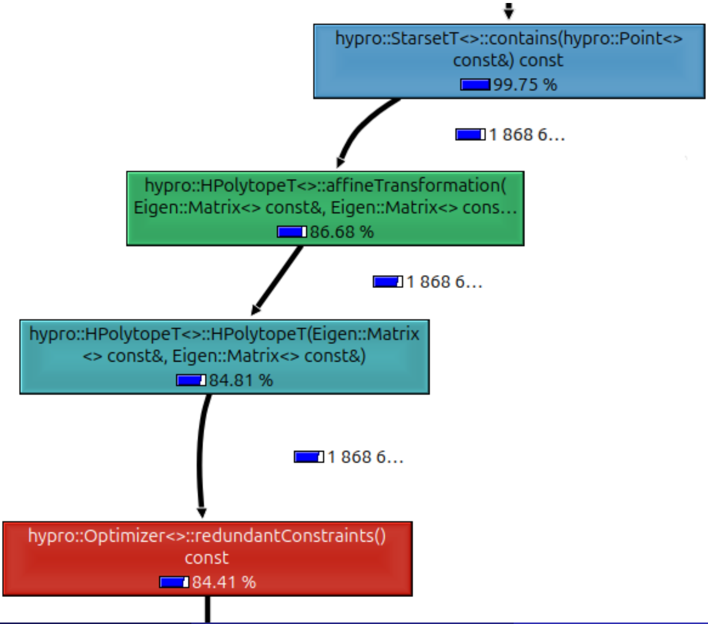
Benchmarks - Simple Running Example



Benchmarks - Simple Running Example



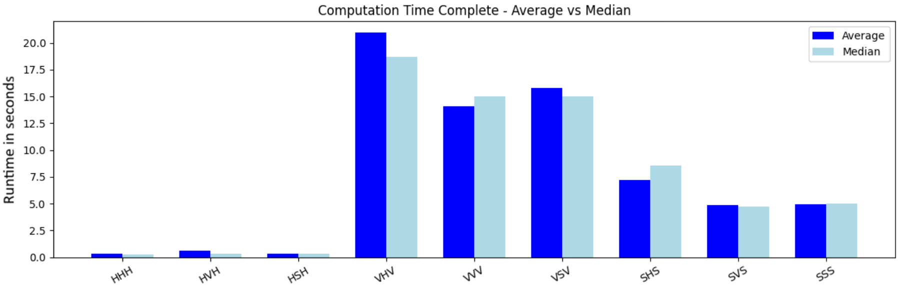
Benchmarks - Simple Running Example



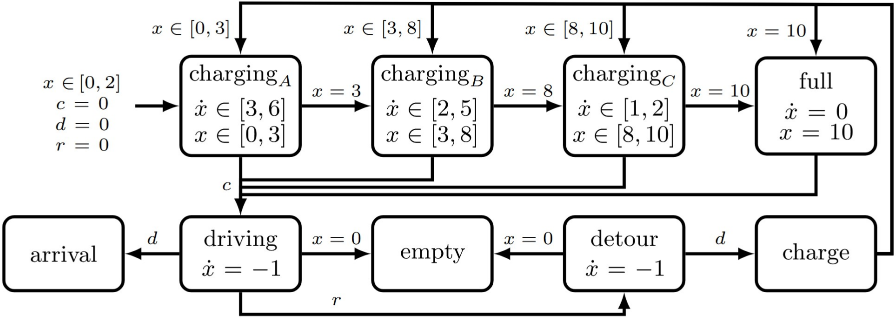
Benchmarks - Simple Running Example

```
template <typename Number, typename Converter, typename Setting>
bool StarsetT<Number, Converter, Setting>::contains( const Point<
    Number>& point ) const {
    HPolytopeT<Number, Converter, HPolytopeOptimizerCaching>
        transformedStar = this->constraints().
        affineTransformation(mGenerator, mCenter);
    hypro::Optimizer<Number> optimizer( transformedStar.matrix(),
         transformedStar.vector());
    return optimizer.checkPoint(point);
}
```

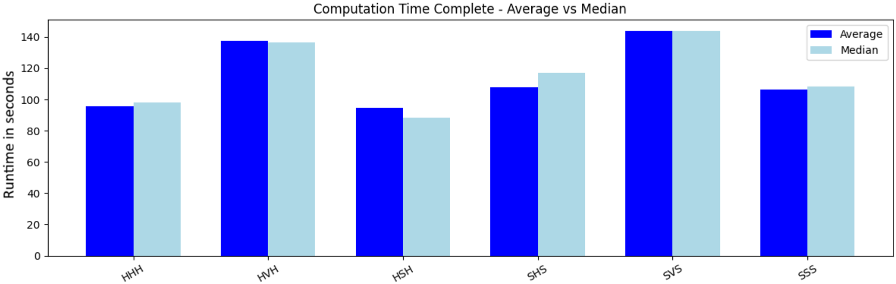
Benchmarks - Simple Running Example



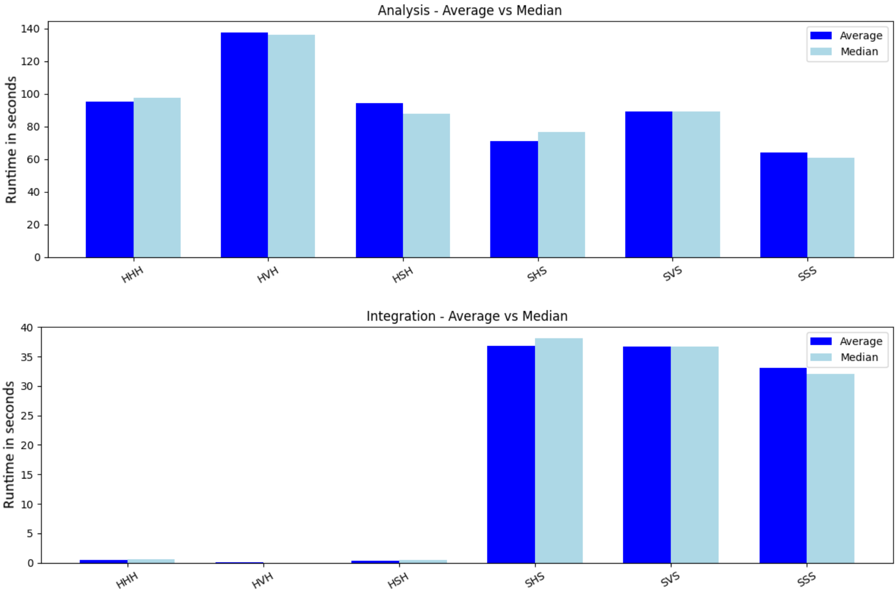
Benchmarks - Car Running Example



Benchmarks - Car Running Example



Benchmarks - Car Running Example



Future Work

1. Improve conversions
2. Improve contains method
3. Further benchmarking for error insight
4. Further tracing
5. Exploring different state set representation