Automatic Dynamic Parallelotope Bundles for Reachability Analysis of Nonlinear Systems

19th International Conference on Formal Modeling and Analysis of Timed Systems

Edward Kim[1]
Stanley Bak[2]
Parasara Sridhar Duggirala[1]

[1]University of North Carolina at Chapel Hill
[2]Stony Brook University

Background: Reachability with Template Polyhedra

1. Reachable set computation is an instrumental tool in performing safety analysis over non-linear systems.
2. One of many techniques in computing the overapproximation of the reachable sets for discrete non-linear systems is to use **template polyhedra** to bound the reachable set.
3. We are particularly interested in *parallelotopes* as our template polyhedra.

Background: Parallelotopes

Definition

A *parallelotope* in $\mathbb{R}^n$ is represented as a tuple $\langle \mathcal{T}, c_l, c_u \rangle$ where $\mathcal{T} \in \mathbb{R}^{n \times n}$ are called *template directions* and $c_l, c_u \in \mathbb{R}^n$ such that $\forall_{1 \leq i \leq n}\ c_l[i] \leq c_u[i]$ are called *bounds*. The half-space representation defines the set of states

$$P = \{\, x \in \mathbb{R}^n \mid c_l[i] \leq \mathcal{T}_i x \leq c_u[i],\ 1 \leq i \leq n \}.$$

Definition

A *parallelotope bundle* Q is a set of parallelotopes $\{P_1, \ldots, P_m\}$. The set of states represented by a parallelotope bundle is given as the intersection

$$Q = \bigcap_{i=1}^{m} P_i$$

Background: Drawbacks

1. Hitherto only *static* parallelotopes have been considered. In other words, the template directions specifying the parallelotopes are to be given as user input at the beginning of the reachable set computation.
2. The template directions chosen are generally the axis-aligned, diagonal directions. However, it's not clear that these directions necessary yield good overapproximations.
3. Since the template directions are set at the beginning, they cannot adapt to the behavior of the dynamics. This could yield overapproximations which are too conservative for any practical use.

Contributions

1. We present a method which is both *dynamic* and *automatic*. Our method utilizes the Principal Component Analysis (PCA) and Local Linear Approximations.
2. We extend our tool *Kaa* to leverage NASA's *Kodiak* to perform the optimization procedure.
3. We parallelize our implementation to scale with an increasing number of parallelotopes in our bundles.