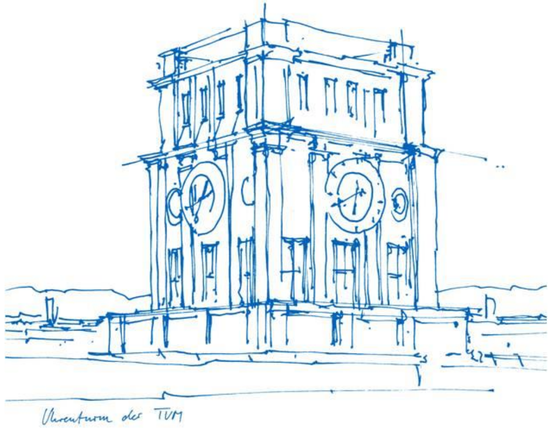
Technische
Universität
München



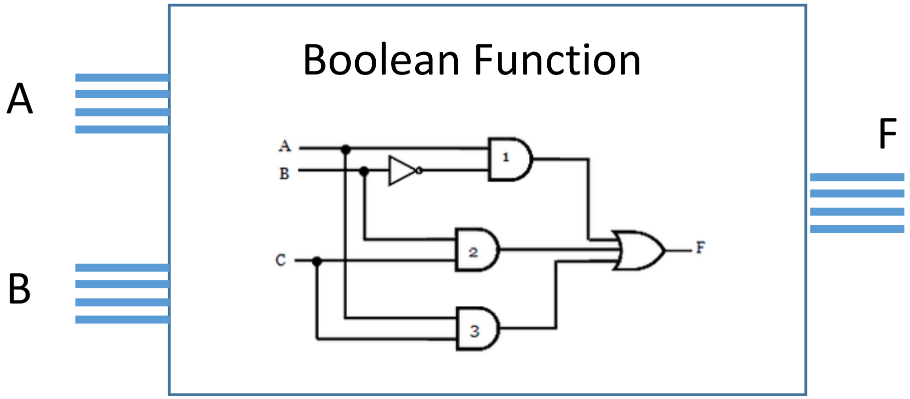
Reachability Analysis for Logical Systems Using Logical Zonotopes and their Polynomial Extension

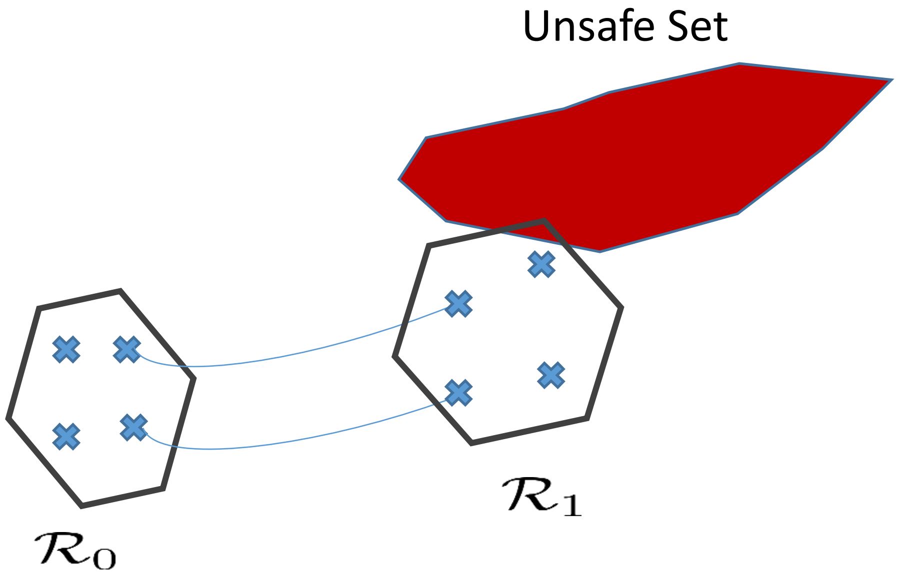
Amr Alanwar
Assistant Professor
Technical University of Munich

International Online Seminar on Interval Methods in Control Engineering

Safety Guarantees through Reachability Analysis

- We aim to guarantee that F does not go to unsafe set given set of inputs





Uncertainty in Logic

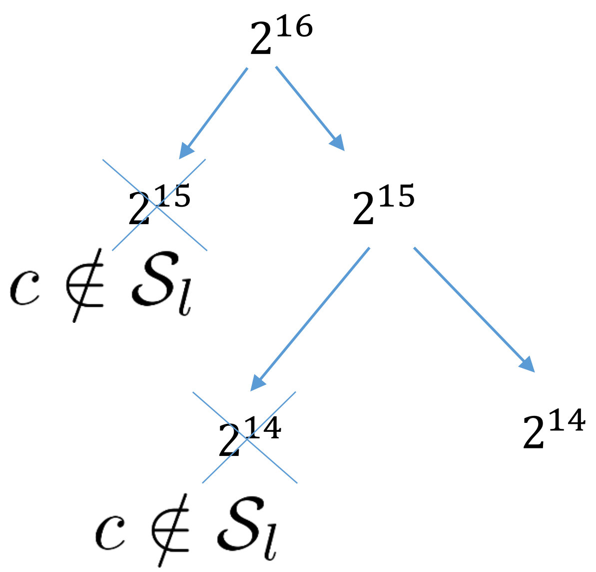
$$P_1 = \neg(P_2 \lor P_4) \land A_1 \ldots$$

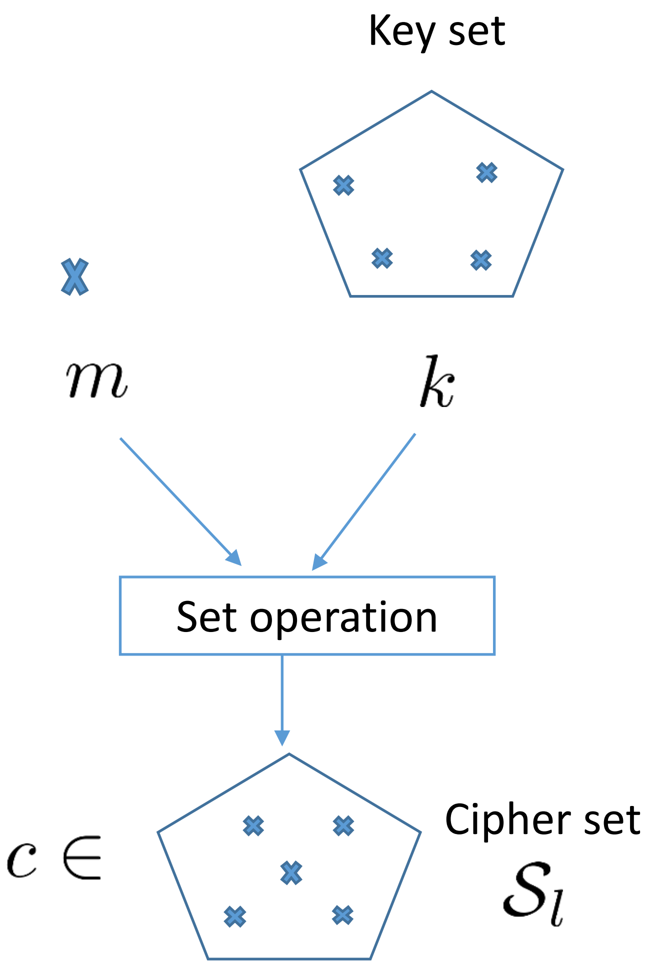
$$P_2 = 0 \qquad P_2 = 1$$

How can we handle such uncertainty on a large scale?

Further Motivation in Cryptography

- Chosen Plaintext-Ciphertext attack: We have a message m and its ciphertext c and we aim to find the 16-bits key





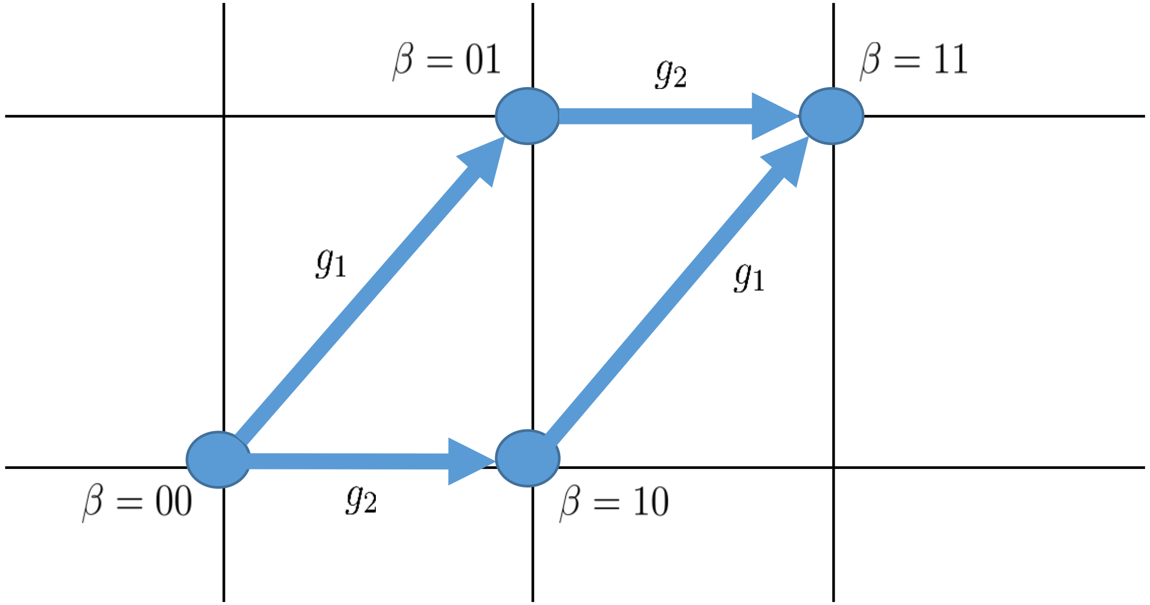
Logical Zonotope

A **logical zonotope** $\mathcal{L} = \langle c, G \rangle$ is a set

$$\mathcal{L} = \left\{ x \in \mathbb{B}^n \;\middle|\; x = c \bigoplus_{i=1}^{\gamma} g_i \beta_i,\ \beta_i \in \{0,1\} \right\}.$$

where $c \in \mathbb{B}^n$ is a point and $G = [g_1 \quad \dots \quad g_\gamma] \in \mathbb{B}^{n \times \gamma}$ the generator vectors

Logical zonotopes can represent up to 2^γ binary vectors with γ generators



Logical Zonotopes Examples

$$\mathcal{L} = \left\{ x \in \mathbb{B}^n \;\middle|\; x = c \bigoplus_{i=1}^{\gamma} g_i \beta_i,\ \beta_i \in \{0,1\} \right\}.$$

One generator

$$\mathcal{L}_1 = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Two generators

$$\mathcal{L}_2 = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \rangle$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \oplus 0 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \oplus 1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \oplus 0 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \oplus 1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

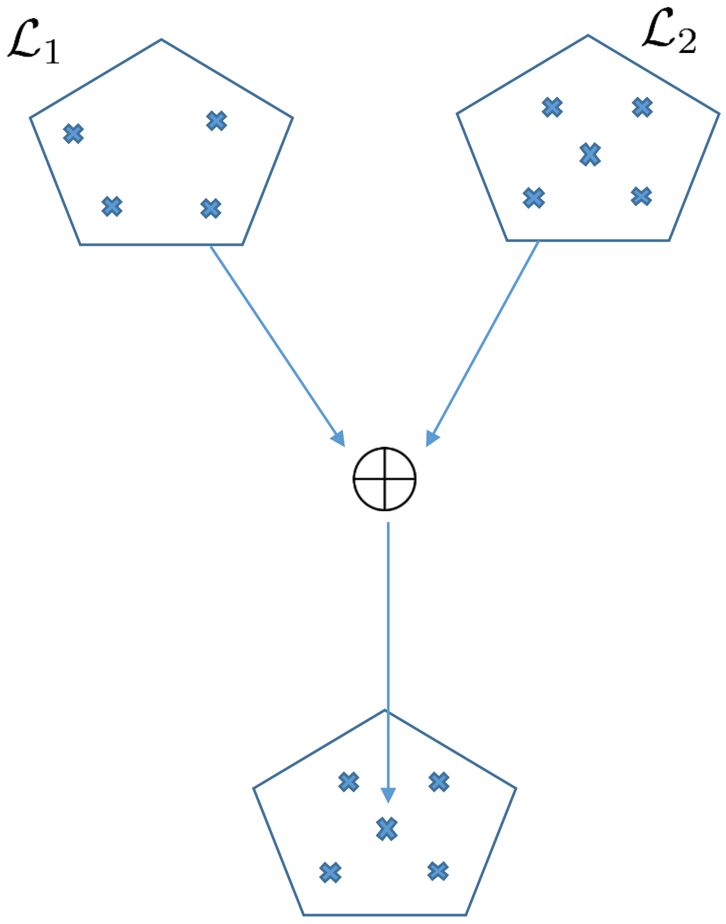
Minkowski XOR

- Definition

$$\mathcal{L}_1 \oplus \mathcal{L}_2 = \{z_1 \oplus z_2 | z_1 \in \mathcal{L}_1, z_2 \in \mathcal{L}_2\}$$

- Computation

$$\mathcal{L}_1 \oplus \mathcal{L}_2 = \left\langle c_1 \oplus c_2, \left[G_1, G_2\right] \right\rangle$$



Example - Minkowski XOR

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

$$\mathcal{L}_1 = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle \qquad \mathcal{L}_2 = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \rangle$$

Logical zonotope

$$\mathcal{L}_1 \oplus \mathcal{L}_2 = \left\langle c_1 \oplus c_2, [G_1, G_2] \right\rangle$$

$$\mathcal{L}_1 \oplus \mathcal{L}_2 = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \rangle$$

$$= \langle \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \rangle$$

$$\mathcal{L}_1 \oplus \mathcal{L}_2 \rightarrow \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Pointwise

$$\mathcal{L}_1 \rightarrow P_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \qquad \mathcal{L}_2 \rightarrow P_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$P_1 \oplus P_2 \rightarrow \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

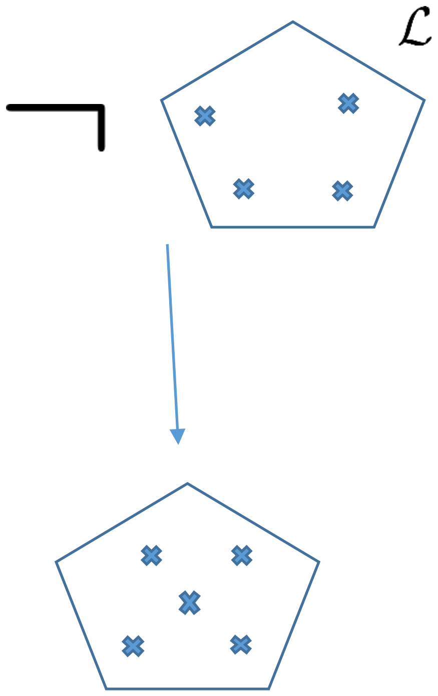
Minkowski NOT

- Definition

$$\neg\mathcal{L} = \{\neg z | z \in \mathcal{L}\}$$

- Computation

$$\neg\mathcal{L} = \left\langle c \oplus 1, G \right\rangle$$



Example - Minkowski NOT

$$\mathcal{L}_1 = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle$$

Logical zonotope

$$\begin{aligned} \neg\mathcal{L} &= \langle c \oplus 1, G \rangle \\ &= \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix} \oplus 1, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle \\ &= \langle \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle \end{aligned}$$

↓

$$\neg\mathcal{L}_1 \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Pointwise

$$\mathcal{L}_1 \rightarrow P_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

↓

$$\neg P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Can we do more than Minkowski XOR and NOT?

Minkowski AND

- Definition

$$\mathcal{L}_1\mathcal{L}_2 = \{z_1 z_2 | z_1 \in \mathcal{L}_1, z_2 \in \mathcal{L}_2\}$$

- Computations

$$\mathcal{L}_1\mathcal{L}_2 \subseteq \langle c_1 c_2, \Big[c_1 g_{2,1}, \dots, c_1 g_{2,\gamma_2}, c_2 g_{1,1}, \dots, c_2 g_{1,\gamma_1}, g_{1,1} g_{2,1}, g_{1,1} g_{2,2}, \dots, g_{1,\gamma_1} g_{2,\gamma_2}\Big]\rangle$$

Minkowski NAND

- Definition

$$\mathcal{L}_1 \barwedge \mathcal{L}_2 = \{z_1 \barwedge z_2 | z_1 \in \mathcal{L}_1, z_2 \in \mathcal{L}_2\}$$

- Computation

$$\mathcal{L}_1 \barwedge \mathcal{L}_2 = \neg(\mathcal{L}_1 \mathcal{L}_2)$$

- NAND is a universal gate. Thus, we are able to do other logical operations:

OR $\quad \mathcal{L}_1 \vee \mathcal{L}_2 = (\mathcal{L}_1 \barwedge \mathcal{L}_1) \barwedge (\mathcal{L}_2 \barwedge \mathcal{L}_2)$

NOR $\quad \mathcal{L}_1 \not\vee \mathcal{L}_2 = \neg(\mathcal{L}_1 \vee \mathcal{L}_2)$

Reachability Analysis

Consider a system with a logical function

$$x(k+1) = f\big(x(k), u(k)\big)$$

Theorem *Given a logical function* $f : \mathbb{B}^{n_x} \times \mathbb{B}^{n_u} \rightarrow \mathbb{B}^{n_x}$ *and starting from initial set* $\hat{\mathcal{R}}_0$ *where* $x(0) \in \hat{\mathcal{R}}_0$*, then the reachable region computed as*

$$\hat{\mathcal{R}}_{k+1} = f\big(\hat{\mathcal{R}}_k, \mathcal{U}_k\big)$$

using logical zonotopes operations over-approximates the exact reachable set, i.e., $\hat{\mathcal{R}}_{k+1} \supseteq \mathcal{R}_{k+1}$.

Semi-tensor Product

Given two matrices $M \in \mathbb{B}^{m\times n}$ and $N \in \mathbb{B}^{p\times q}$, the semi-tensor product is defined as:

$$M \ltimes N = (M \otimes I_{s_1})(N \otimes I_{s_2}),$$

with s as the least common multiple of n and p, $s_1 = s/n$, and $s_2 = s/p$

$\otimes$: Kronecker product

Semi Tensor Product between Logical Zonotopes

- Definition

$$\mathcal{L}_1 \ltimes \mathcal{L}_2 = \{z_1 \ltimes z_2 | z_1 \in \mathcal{L}_1, z_2 \in \mathcal{L}_2\}$$

- Computation

$$\mathcal{L}_1 \ltimes \mathcal{L}_2 \subseteq \left\langle c_1 \ltimes c_2, G_{\ltimes} \right\rangle$$

where

$$G_{\ltimes} = \Big[c_1 \ltimes g_{2,1}, \dots, c_1 \ltimes g_{2,\gamma_1}, g_{1,1} \ltimes c_2, \dots, g_{1,\gamma_1} \ltimes c_2$$
$$g_{1,1} \ltimes g_{2,1}, \dots, g_{1,\gamma_1} \ltimes g_{2,\gamma_2}\Big]$$

Can we have an exact ANDing?

ANDing Problem

- ANDing Proof

$$\exists \hat{\beta}_1 : z_1 = c_1 \bigoplus_{i=1}^{\gamma_1} g_{1,i}\hat{\beta}_{1,i} \qquad \exists \hat{\beta}_2 : z_2 = c_2 \bigoplus_{i=1}^{\gamma_2} g_{2,i}\hat{\beta}_{2,i}$$

$$\longrightarrow \quad z_1 z_2 = c_1 c_2 \bigoplus_{i=1}^{\gamma_2} c_1 g_{2,i}\hat{\beta}_{2,i} \bigoplus_{i=1}^{\gamma_1} c_2 g_{1,i}\hat{\beta}_{1,i} \bigoplus_{i=1,j=1}^{\gamma_1,\gamma_2} g_{1,i} g_{2,j}\hat{\beta}_{1,i}\hat{\beta}_{2,j} \, .$$

- What if we allow for ANDing between factors?

$$\mathcal{P} = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \alpha_1 \oplus \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \alpha_1 \alpha_2 \middle| \alpha_1, \alpha_2 \in \{0,1\} \right\}.$$

Polynomial Logical Zonotope

A **polynomial logical zonotope** $\mathcal{P} = \langle c, G, E \rangle$ is a set

$$\mathcal{P} = \left\{ x \in \mathbb{B}^n \;\middle|\; x = c \bigoplus_{i=1}^{h} \left(\prod_{k=1}^{p} \alpha_k^{E_{(k,i)}} \right) g_i, \alpha_k \in \{0,1\} \right\}.$$

- $c \in \mathbb{B}^n$ is a point
- $G = \begin{bmatrix} g_1 & \dots & g_q \end{bmatrix} \in \mathbb{B}^{n \times h}$ is a dependent generator matrix
- $E \in \mathbb{B}^{p \times h}$ is an exponent matrix

Example

- Consider the following polynomial logical zonotope

$$\bar{\mathcal{P}}_1 = \left\langle \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \right\rangle$$

- This translates to the following set

$$\bar{\mathcal{P}}_1 = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \alpha_1 \oplus \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \alpha_1 \alpha_2 \middle| \alpha_1, \alpha_2 \in \{0, 1\} \right\}.$$

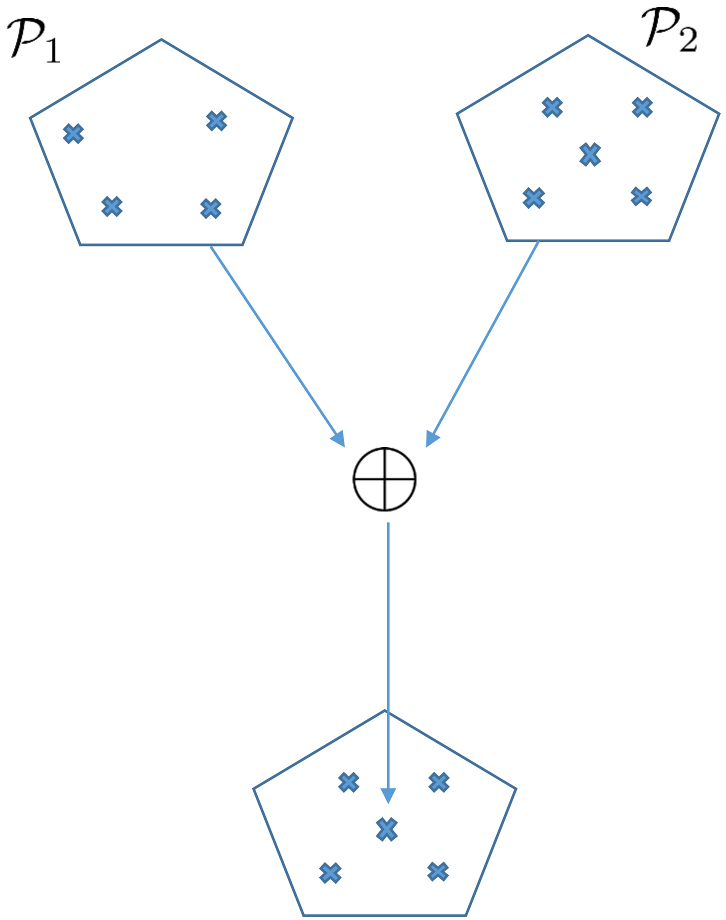
Minkowski XOR

- Definition

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \{z_1 \oplus z_2 | z_1 \in \mathcal{P}_1, z_2 \in \mathcal{P}_2\}$$

- Computation

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \left\langle c_1 \oplus c_2, [G_1, G_2], \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix} \right\rangle$$



Example - XOR

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

$$\mathcal{P}_1 = \langle 0, 1, 1 \rangle$$

Polynomial Logical zonotope

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \left\langle c_1 \oplus c_1, [G_1, G_2], \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix} \right\rangle$$

$$\mathcal{P}_1 \oplus \mathcal{P}_1 = \left\langle 0, [1, 1], \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\rangle$$

$$\mathcal{P}_1 \oplus \mathcal{P}_1 \rightarrow \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Pointwise

$$\mathcal{P}_1 \oplus \mathcal{P}_1 \rightarrow 0$$

Dependency Problem

- Dependent factors

$$\alpha g_1 \oplus \alpha g_2 = \alpha(g_1 \oplus g_2)$$

- Independent factors

$$\alpha_1 g_1 \oplus \alpha_2 g_2 = \begin{bmatrix} g_1 & g_2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

- Inspired by the traditional polynomial zonotopes, we need to identify the factors with ids

Niklas Kochdumper and Matthias Althoff. Sparse polynomial zonotopes: A novel set representation for reachability analysis. IEEE Transactions on Automatic Control, 66(9):4043–4058, 202

Polynomial Logical Zonotope

A **polynomial logical zonotope** $\mathcal{P} = \langle c, G, E, id \rangle$ is a set

$$\mathcal{P} = \left\{ x \in \mathbb{B}^n \;\middle|\; x = c \bigoplus_{i=1}^{h} \left(\prod_{k=1}^{p} \alpha_k^{E_{(k,i)}} \right) g_i, \alpha_k \in \{0,1\} \right\}.$$

- $c \in \mathbb{B}^n$ is a point
- $G = \begin{bmatrix} g_1 & \dots & g_q \end{bmatrix} \in \mathbb{B}^{n \times h}$ is a dependent generator matrix
- $E \in \mathbb{B}^{p \times h}$ is an exponent matrix
- Id vector for identifying the dependent factors

Merge IDs

$$\bar{\mathcal{P}}_1 = \left\langle \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0 & 1\\1 & 1\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 1\\0 & 1\end{bmatrix}, \begin{bmatrix}1 & 2\end{bmatrix} \right\rangle$$

$$\downarrow$$

$$\mathcal{P}_1 = \left\langle \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0 & 1\\1 & 1\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 1\\0 & 1\\0 & 0\end{bmatrix}, \begin{bmatrix}1 & 2 & 3\end{bmatrix} \right\rangle$$

$$\bar{\mathcal{P}}_2 = \left\langle \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\\1 & 0\end{bmatrix}, \begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 3\end{bmatrix} \right\rangle$$

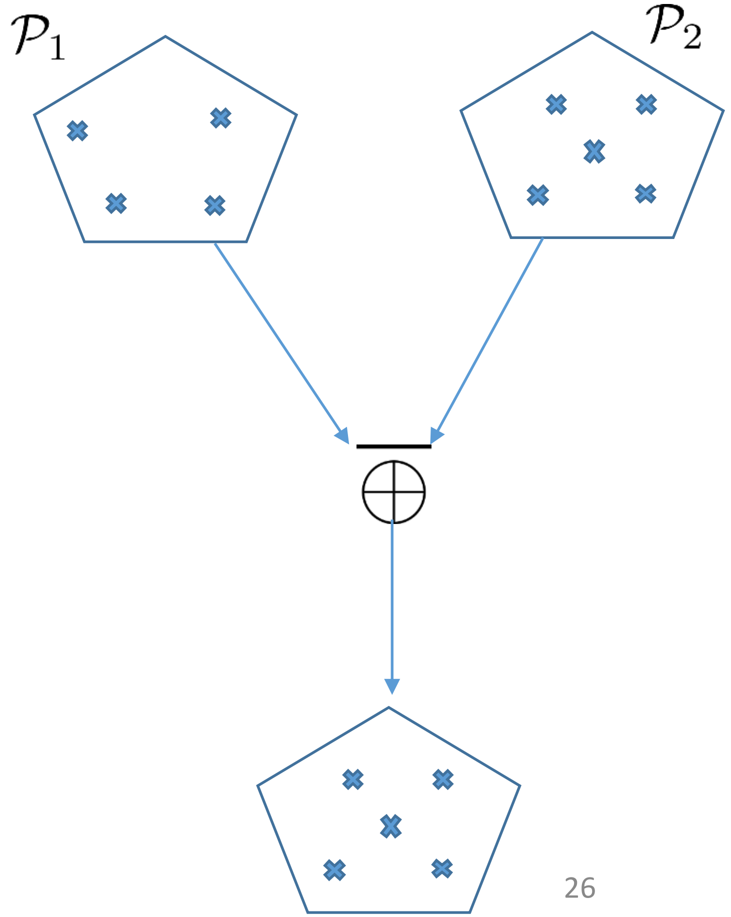
$$\downarrow$$

$$\mathcal{P}_2 = \left\langle \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\\1 & 0\end{bmatrix}, \begin{bmatrix}0 & 1\\0 & 0\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 2 & 3\end{bmatrix} \right\rangle$$

Exact XOR

- Given two polynomial logical zonotopes $\mathcal{P}_1 = \langle c_1, G_1, E_1, id \rangle$ and $\mathcal{P}_2 = \langle c_2, G_2, E_2, id \rangle$ with a common identifier vector id, the exact XOR is computed as:

$$\mathcal{P}_1 \overline{\oplus} \mathcal{P}_2 = \left\langle c_1 \oplus c_2, \left[G_1, G_2\right], \left[E_1, E_2\right], id \right\rangle$$

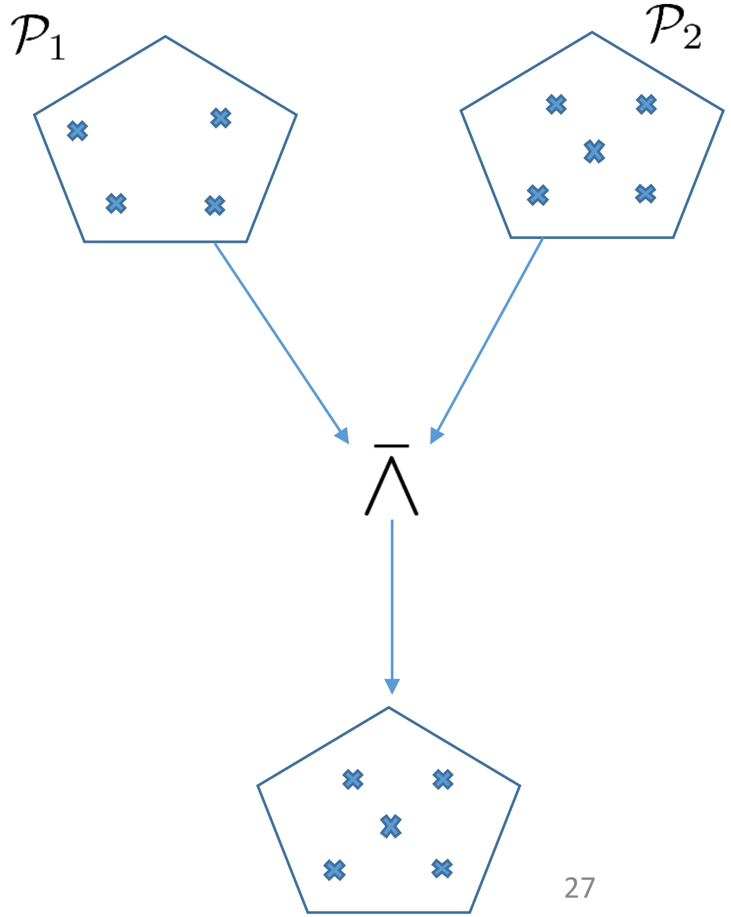


Exact AND

- Given two polynomial logical zonotopes $\mathcal{P}_1 = \langle c_1, G_1, E_1, id \rangle$ and $\mathcal{P}_2 = \langle c_2, G_2, E_2, id \rangle$

with a common identifier vector id, the exact AND is computed and lead to the following $\mathcal{P}_{\bar{\wedge}} = \langle c_{\bar{\wedge}}, G_{\bar{\wedge}}, E_{\bar{\wedge}}, id \rangle$ where

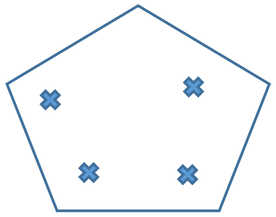
$$
\begin{aligned}
c_{\bar{\wedge}} =& c_1 c_2, \\
G_{\bar{\wedge}} =& \Big[c_1 g_{2,1}, \dots, c_1 g_{2,h_2}, c_2 g_{1,1}, \dots, c_2 g_{1,h_1}, \\
& g_{1,1} g_{2,1}, \dots, g_{1,h_1} g_{2,h_2} \Big], \\
E_{\bar{\wedge}} =& \Big[E_{2,(.,1)}, \dots, E_{2,(.,h_2)}, E_{1,(.,1)}, \dots, E_{1,(.,h_1)}, \\
& \max\big(E_{1,(.,1)}, E_{2,(.,1)}\big), \dots, \max\big(E_{1,(.,h_1)}, E_{2,(.,h_2)}\big) \Big]
\end{aligned}
$$



Enclose Points by a Logical Zonotope

Given a list $\mathcal{S}_p$ of p binary vectors, the logical zonotope $\mathcal{L}_p = \langle c_p, G_p \rangle$ with is given by $\mathcal{S}_{p,i} \in \mathcal{L}_p, \forall i = \{1, \dots, p\}$

$$c_p = \mathcal{S}_{p,1},$$
$$g_{p,i-1} = \mathcal{S}_{p,i} \oplus c_p, \ \forall i = \{2, \dots, p\}.$$



β_3	β_2	β_1
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$$c_p = \mathcal{S}_{p,1}$$
$$g_{p,1} \oplus c_p = \mathcal{S}_{p,2} \oplus c_p \oplus c_p$$
$$g_{p,2} \oplus c_p = \mathcal{S}_{p,3} \oplus c_p \oplus c_p$$
$$g_{p,3} \oplus c_p = \mathcal{S}_{p,4} \oplus c_p \oplus c_p$$

Reduce the Number of Generators

Algorithm 1: Function `reduce` to decrease the number of generators of a logical zonotope.

Input: A logical zonotope $\mathcal{L} = \langle c_{\mathcal{L}}, G_{\mathcal{L}} \rangle$ with large number $\gamma_{\mathcal{L}}$ of generators.

Output: A logical zonotope $\mathcal{L}_r = \langle c_{\mathcal{L}_r}, G_{\mathcal{L}_r} \rangle$ with $\gamma_{\mathcal{L}_r} \leq \gamma_{\mathcal{L}}$ generators.

$B_{\mathcal{L}}=$ `evaluate`$(\mathcal{L})$ // Compute a list $B_{\mathcal{L}}$ of all binary vectors contained in $\mathcal{L}$.

for $i = 1 : \gamma_{\mathcal{L}}$ **do**

    $B_{\mathcal{L}_r}=$ `evaluate`$(\mathcal{L} \setminus g_{\mathcal{L}}^{(i)})$ // Compute a list $B_{\mathcal{L}_r}$ of all binary vectors contained in $\mathcal{L}$ without the generator $g_{\mathcal{L}}^{(i)}$.

    if *isequal*$(B_{\mathcal{L}}, B_{\mathcal{L}_r})$ **then**

        $g_{\mathcal{L}} =$ `removeGenerator`$(g_{\mathcal{L}}^{(i)})$

$\mathcal{L}_r = \langle c_{\mathcal{L}}, G_{\mathcal{L}} \rangle$

Binary Decision Diagram (BDD)

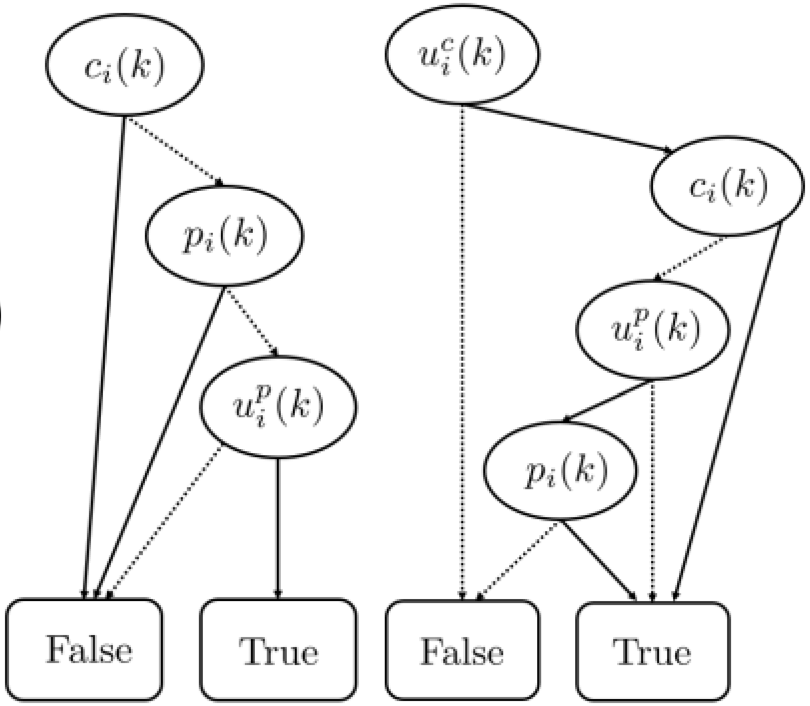
- Introduced by Randal E. Bryant in mid-80s
- A data structure that is used to represent a Boolean function
- Given a proper variable ordering, BDDs can evaluate Boolean functions with linear complexity in the number of variables

Alan John Hu. 1996. *Techniques for efficient formal verification using binary decision diagrams*. Stanford university

Reduced Binary Decision Diagram

$$p_i(k+1) = u_i^p(k)\neg p_i(k)\neg c_i(k)$$

$$c_i(k+1) = \neg p_i(k+1)(u_i^c(k) \vee (\neg p_i(k) p_i(k+1)))$$



(a) $p_i(k+1)$ **(b)** $c_i(k+1)$

Boolean Control Network (BCN)

h_i : Logical function

- We consider the BCN with the following dynamics

$$x_1(k+1) = h_1(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)),$$
$$\vdots$$
$$x_n(k+1) = h_n(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)),$$

Fangfei Li and Yang Tang. 2017. Robust Reachability of Boolean Control Networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics

Exact Reachable Set of BCN

- There exits a unique matrix, named structural matrix L such that

$$x(k+1) = L \ltimes u(k) \ltimes x(k)$$

where $x = \ltimes_{i=1}^{n} x_i$ and $u = \ltimes_{j=1}^{m} u_i$

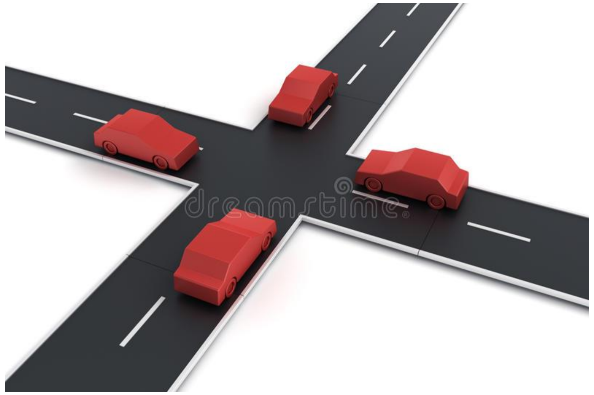
- Given a set of initial states $\mathcal{X}_0 \subset \mathbb{Z}^n$ and a set of possible inputs $\mathcal{U} \subset \mathbb{Z}^m$, the exact reachable set

$$\mathcal{R}_N = \{x(N) \in \mathbb{B}^n \mid \forall k \in \{0, \ldots, N-1\} : \\ x(k+1) = L \ltimes u(k) \ltimes x(k), x(0) \in \mathcal{X}_0, u(k) \in \mathcal{U}\}$$

Intersection Crossing Protocol

$$p_i(k+1) = u_i^p(k)\neg p_i(k)\neg c_i(k)$$
$$c_i(k+1) = \neg p_i(k+1)(u_i^c(k) \vee (\neg p_i(k)p_i(k+1)))$$

$$u_1^p(k) \in \{0,1\},\ u_2^p(k) \in \{0,1\},\ u_1^c(k) \in \{0,1\},\ u_2^c(k) \in \{0,1\},\ k = 0,\ldots,N$$



Intersection Crossing Protocol – 4 Cars

Table 1: Execution Time (seconds) and number of points in each set (size) for verifying an intersection crossing protocol.

	Zonotope		Poly. Zonotope		BDD		BCN	
Steps N	Time	Size	Time	Size	Time	Size	Time	Size
5	0.05	16	0.15	13	1.17	14	3.40	14
10	0.06	16	0.18	14	3.32	14	7.75	14
50	0.15	16	0.25	14	19.87	14	48.40	14
100	0.26	16	0.45	14	39.78	14	104.91	14
1000	1.84	16	2.84	14	406.60	14	1142.10	14

$$p_i(k+1) = u_i^p(k)\neg p_i(k)\neg c_i(k)$$

$$c_i(k+1) = \neg p_i(k+1)(u_i^c(k) \vee (\neg p_i(k)p_i(k+1)))$$

High-Dimensional Boolean Function

We initially assign sets of 10 possible values to $B_1(0), B_2(0), B_3(0)$

$$B_1(k+1) = U_1(k) \vee (B_2(k) \odot B_1(k)), \tag{1}$$

$$B_2(k+1) = B_2(k) \odot (B_1(k) \wedge U_2(k)), \tag{2}$$

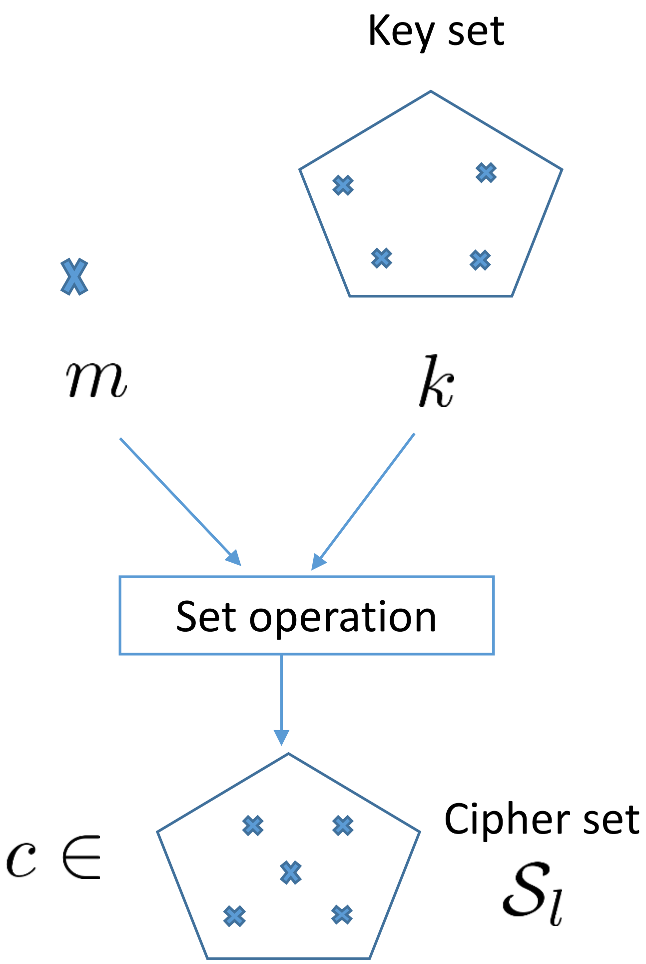
$$B_3(k+1) = B_3(k) \mathbin{\wedge\!\!\!\!\sim} (U_2(k) \odot U_3(k)). \tag{3}$$

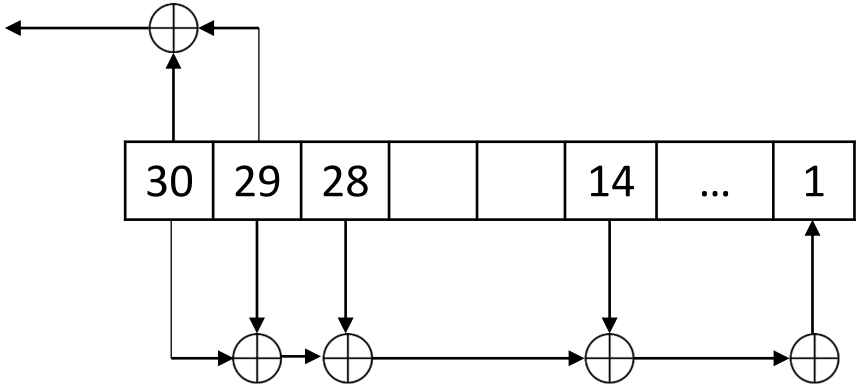
High-Dimensional Boolean Function

Table 2: Execution Time (seconds) for reachability analysis of a high-dimensional Boolean function (*estimated times).

	Zonotope		Poly. Zonotope		BDD	
Steps N	Time	Size	Time	Size	Time	Size
2	0.04	768	0.05	211	0.34	211
3	0.05	896	0.06	580	1.86×10^5	580
4	0.06	896	0.07	580	2.44×10^6*	-
5	0.07	896	0.56	580	$> 10^6$*	-

Key Search


Key set
m
k
Set operation
$c \in$
Cipher set
$\mathcal{S}_l$


30
29
28
14
...
1

Execution Time

Table 3: Execution Time (seconds) of exhaustive key search (*estimated times).

Key Size	Zonotope	Traditional Search
30	1.97	$1.18 \times 10^{6}*$
60	4.76	$1.26 \times 10^{15}*$
120	7.95	$1.46 \times 10^{33}*$

Acknowledgement

Karl Henrik Johansson

Frank Jiang

Samy Amin

1. “Logical zonotopes: A Set Representation for the Formal Verification of Boolean Functions” A Alanwar, FJ Jiang, S Amin, KH Johansson
2. “Polynomial Logical Zonotopes: A Set Representation for Reachability Analysis of Logical Systems” A Alanwar, FJ Jiang, KH Johansson

Conclusions

- Logical zonotope set representation
- Polynomial logical zonotope set representation
- Logical operations on generators instead of iterating over points
- Applications of logical zonotopes

https://github.com/aalanwar/Logical-Zonotope

https://sites.google.com/view/amr-alanwar/


THANKS