A novel reduction method for star sets and its application in neural network verification

Bachelor thesis defence

Vahe Vkhkryan
Supervisor: Erika Ábrahám
Advisor: László Antal

LuFG Theory of Hybrid Systems

WS 2023/24

Table of contents

Star set representation [Bak et al., 2017]

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$$

Star set representation [Bak et al., 2017]

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$$

- $\mathbf{c} \in \mathbb{R}^n$ - *center*

Star set representation [Bak et al., 2017]

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$$

- $\mathbf{c} \in \mathbb{R}^n$ - *center*
- $\mathcal{V} = [\mathbf{v^1}, \ldots, \mathbf{v^m}], \mathbf{v^i} \in \mathbb{R}^n$ (*generator vectors*) $\rightarrow$ *generator matrix*

Star set representation [Bak et al., 2017]

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$$

- $\mathbf{c} \in \mathbb{R}^n$ - *center*
- $\mathcal{V} = [\mathbf{v^1}, \ldots, \mathbf{v^m}], \mathbf{v^i} \in \mathbb{R}^n$ (*generator vectors*) $\rightarrow$ *generator matrix*

$$\mathcal{V} = \begin{bmatrix} v_1^1 & \cdots & v_1^m \\ \vdots & \ddots & \vdots \\ v_n^1 & \cdots & v_n^m \end{bmatrix} \in \mathbb{R}^{n \times m}$$

Star set representation [Bak et al., 2017]

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$$

- $\mathbf{c} \in \mathbb{R}^n$ - *center*
- $\mathcal{V} = [\mathbf{v^1}, \ldots, \mathbf{v^m}], \mathbf{v^i} \in \mathbb{R}^n$ *(generator vectors)* $\rightarrow$ *generator matrix*

$$\mathcal{V} = \begin{bmatrix} v_1^1 & \cdots & v_1^m \\ \vdots & \ddots & \vdots \\ v_n^1 & \cdots & v_n^m \end{bmatrix} \in \mathbb{R}^{n \times m}$$

- $P : \mathbb{R}^m \rightarrow \{\top, \bot\}$ - *predicate*

$$P(\boldsymbol{\alpha}) \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d} \qquad \begin{aligned} &\mathcal{C} \in \mathbb{R}^{p \times m} \\ &\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \\ &\mathbf{d} \in \mathbb{R}^{p \times 1} \end{aligned}$$

Star set representation [Bak et al., 2017]

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$$

- $\mathbf{c} \in \mathbb{R}^n$ - *center*
- $\mathcal{V} = [\mathbf{v^1}, \ldots, \mathbf{v^m}], \mathbf{v^i} \in \mathbb{R}^n$ (*generator vectors*) $\rightarrow$ *generator matrix*

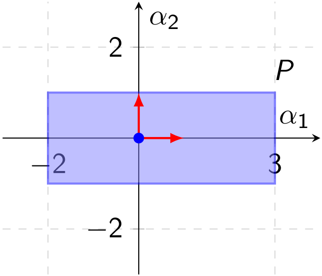
$$\mathcal{V} = \begin{bmatrix} v_1^1 & \cdots & v_1^m \\ \vdots & \ddots & \vdots \\ v_n^1 & \cdots & v_n^m \end{bmatrix} \in \mathbb{R}^{n \times m}$$

- $P : \mathbb{R}^m \rightarrow \{\top, \bot\}$ - *predicate*

$$P(\boldsymbol{\alpha}) \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d} \qquad \begin{aligned} &\mathcal{C} \in \mathbb{R}^{p \times m} \\ &\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \\ &\mathbf{d} \in \mathbb{R}^{p \times 1} \end{aligned}$$

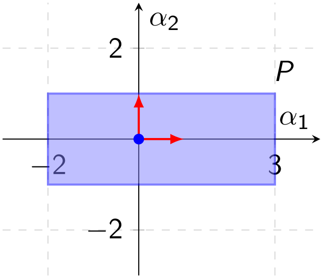
$$[\![\Theta]\!] = \{\mathbf{x} \mid \mathbf{x} = \mathbf{c} + \sum_{i=1}^{m} \alpha_i \mathbf{v^i}, \text{ such that } P(\alpha_1, \ldots, \alpha_m) = \top\}$$

Star set: Example



$$
\begin{cases}
1\alpha_1 + 0\alpha_2 & \leq 3 \\
-1\alpha_1 + 0\alpha_2 & \leq 2 \\
0\alpha_1 + 1\alpha_2 & \leq 1 \\
0\alpha_1 - 1\alpha_2 & \leq 1
\end{cases}
$$

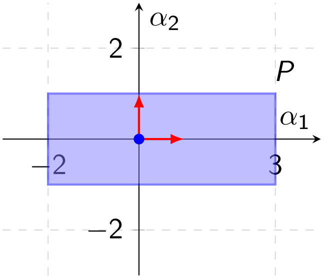
Star set: Example



$$\mathcal{C} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

$$P(\boldsymbol{\alpha}) \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}$$

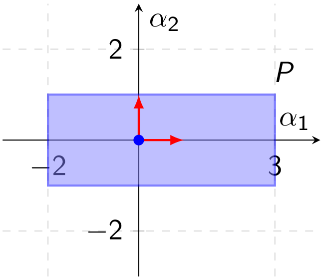
Star set: Example



$$\mathcal{C} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \mathcal{V} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

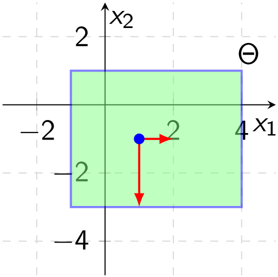
$$P(\boldsymbol{\alpha}) \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}$$

Star set: Example



$$\mathcal{C} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

$$P(\boldsymbol{\alpha}) \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}$$



$$\mathbf{c} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \mathcal{V} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

The problem we aim to solve

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle \;\; P \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d} \qquad \Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle \;\; P' \triangleq \mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d}'$$

$$\mathcal{C} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0.7 & -0.7 & -1 \\ -0.3 & 0.3 & 1 \end{bmatrix} \mathbf{d} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0.5 \\ 0.6 \end{bmatrix} \qquad \mathcal{C}' = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 1 & 0 \\ 0.5 & -0.7 \\ 0.5 & 1.5 \\ -0.5 & 1.5 \end{bmatrix} \mathbf{d}' = \begin{bmatrix} 0 \\ 2 \\ 3 \\ 1.4 \\ 3 \\ 2 \end{bmatrix}$$

$$\mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ 0.71 & 0.71 & 0 \end{bmatrix} \qquad \mathcal{V}' = \begin{bmatrix} 0 & 1 \\ 0.71 & 0 \end{bmatrix}$$

Table of contents

Star set transformation

Proposition 1

Any star set $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$

Star set transformation

Proposition 1

Any star set $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ can be transformed to another star set $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$,

Star set transformation

Proposition 1

Any star set $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ can be transformed to another star set $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$, such that $\mathcal{V} \neq \mathcal{V}', \mathcal{C} \neq \mathcal{C}'$ and $\Theta \equiv \Theta'$.

Star set transformation

Proposition 1

Any star set $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ can be transformed to another star set $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$, such that $\mathcal{V} \neq \mathcal{V}', \mathcal{C} \neq \mathcal{C}'$ and $\Theta \equiv \Theta'$. For the transformation, we use a manually designed invertible matrix $\mathcal{G}$, such that $\mathcal{V}\mathcal{G} = \mathcal{V}'$ and $\mathcal{C}\mathcal{G} = \mathcal{C}'$.

0-column transformation

Proposition 2

For any star set transformation from $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ to $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$ holds:

0-column transformation

Proposition 2

For any star set transformation from $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ to $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$ holds: if $\mathrm{rank}(\mathcal{V}) = r < m$,

0-column transformation

Proposition 2

For any star set transformation from $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ to $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$ holds: if $\text{rank}(\mathcal{V}) = r < m$, then there exists an invertible transformation matrix $\mathcal{G} \in \mathbb{R}^{m \times m}$, such that $\mathcal{V}\mathcal{G} = \mathcal{V}'$,

0-column transformation

Proposition 2

For any star set transformation from $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle,\ P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ to $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle,\ P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d})$ holds: if $\text{rank}(\mathcal{V}) = r < m$, then there exists an invertible transformation matrix $\mathcal{G} \in \mathbb{R}^{m \times m}$, such that $\mathcal{V}\mathcal{G} = \mathcal{V}'$, where $\mathcal{V}'$ has exactly $\mathbf{r}$ linearly independent columns from $\mathcal{V}$ and $\mathbf{m} - \mathbf{r}$ zero-columns.

Table of contents

Remove?

- Remove the corresponding variable in the predicate?

Remove?

- Remove the corresponding variable in the predicate? NO!

Remove?

- Remove the corresponding variable in the predicate? NO!
  Although it has 0-vectors in the generator, however, it implicitly has effects on other variables in the predicate:

Remove?

- Remove the corresponding variable in the predicate? NO!
  Although it has 0-vectors in the generator, however, it implicitly has effects on other variables in the predicate:

$$C_{ij} \cdot \alpha_j \leq d_i - C_{i1}\boldsymbol{\alpha_1} - \sum_{k=2,k\neq j}^{m} C_{ik}\alpha_k$$

Remove?

- Remove the corresponding variable in the predicate? NO!
  Although it has 0-vectors in the generator, however, it implicitly has effects on other variables in the predicate:

$$C_{ij} \cdot \alpha_j \leq d_i - C_{i1}\boldsymbol{\alpha_1} - \sum_{k=2,k\neq j}^{m} C_{ik}\alpha_k$$

- To neutralize these effects we use Fourier-Motzkin variable elimination.

Fourier-Motzkin variable elimination

- Consider the set of linear inequalities in the normal form:

$$\sum_{j=1}^{n} C_{ij} \cdot x_j \leq d_i$$

$$\Rightarrow C_{in} \cdot x_n \leq d_i - \sum_{j=1}^{n-1} C_{ij} \cdot x_j$$

Fourier-Motzkin variable elimination

- Consider the set of linear inequalities in the normal form:

$$\sum_{j=1}^{n} C_{ij} \cdot x_j \leq d_i$$

$$\Rightarrow C_{in} \cdot x_n \leq d_i - \sum_{j=1}^{n-1} C_{ij} \cdot x_j$$

$$\begin{array}{llll} (a)\ C_{in} = 0 & \Rightarrow & \text{constraint } i \text{ puts no bound on } x_n & \\ (b)\ C_{in} > 0 & \Rightarrow & x_n \leq \dfrac{d_i}{C_{in}} - \dfrac{\sum_{j=1}^{n-1} C_{ij} x_j}{C_{in}} & \text{upper bound} \\ (c)\ C_{in} < 0 & \Rightarrow & x_n \geq \dfrac{d_i}{C_{in}} - \dfrac{\sum_{j=1}^{n-1} C_{ij} x_j}{C_{in}} & \text{lower bound} \end{array}$$

Fourier-Motzkin variable elimination

Proposition 3

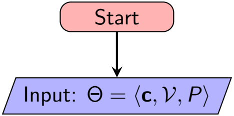
Applying Fourier Motzkin variable elimination on a given system of linear inequalities $\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}$ to eliminate α_i from the system, we get an equisatisfiable system $\mathcal{C}'\boldsymbol{\alpha} \leq \mathbf{d}'$, where all coefficients of α_i are set to zero.

Table of contents

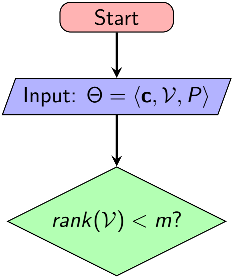
The algorithm

Start

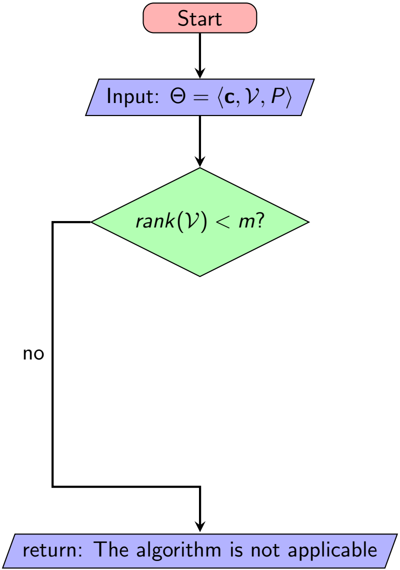
The algorithm



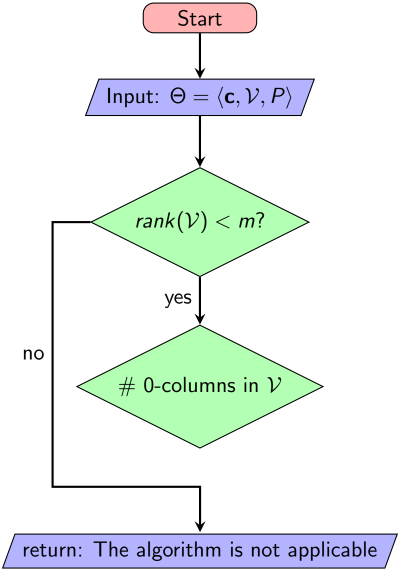
The algorithm



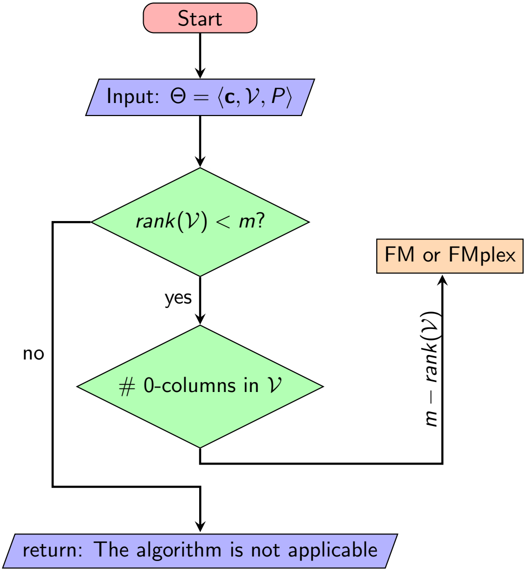
The algorithm



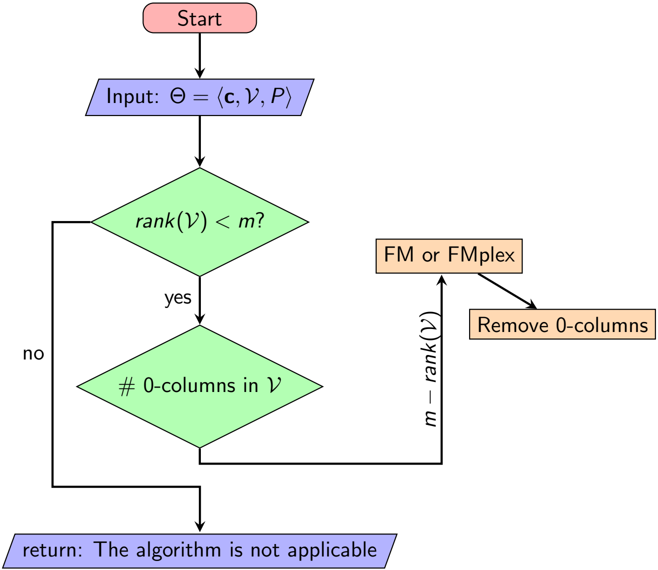
The algorithm



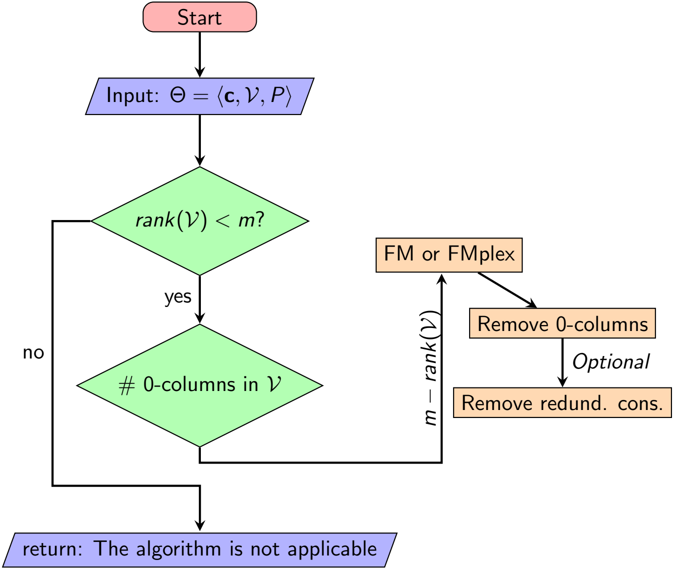
The algorithm



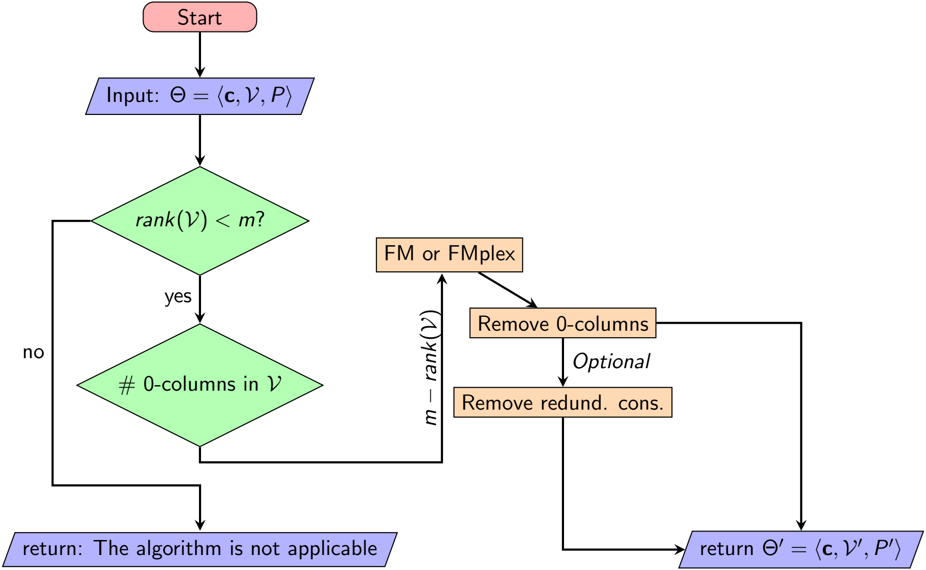
The algorithm

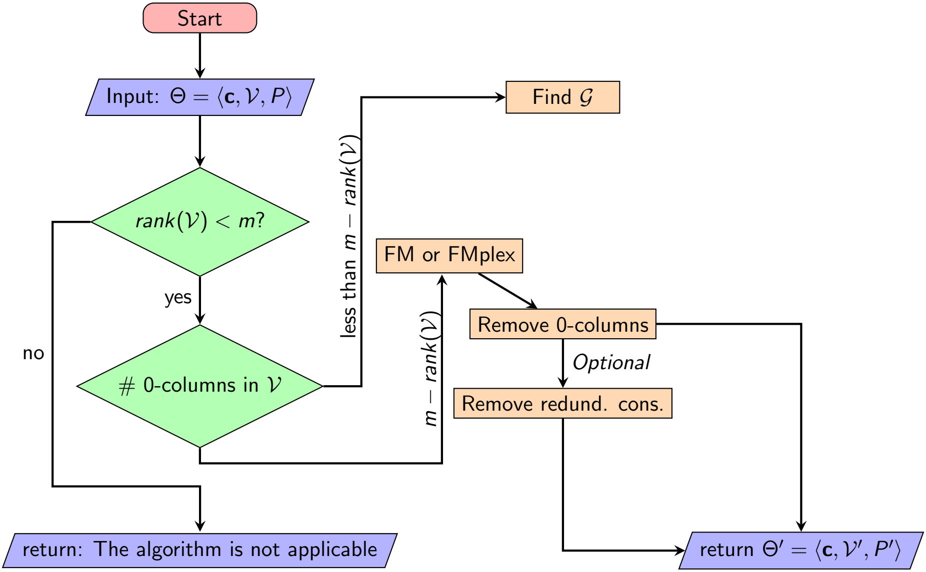


The algorithm

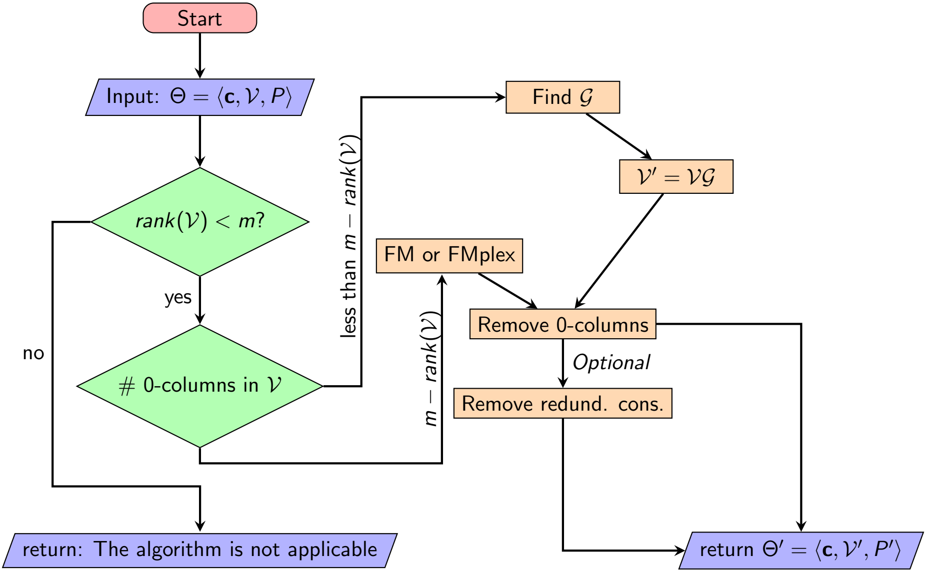


The algorithm

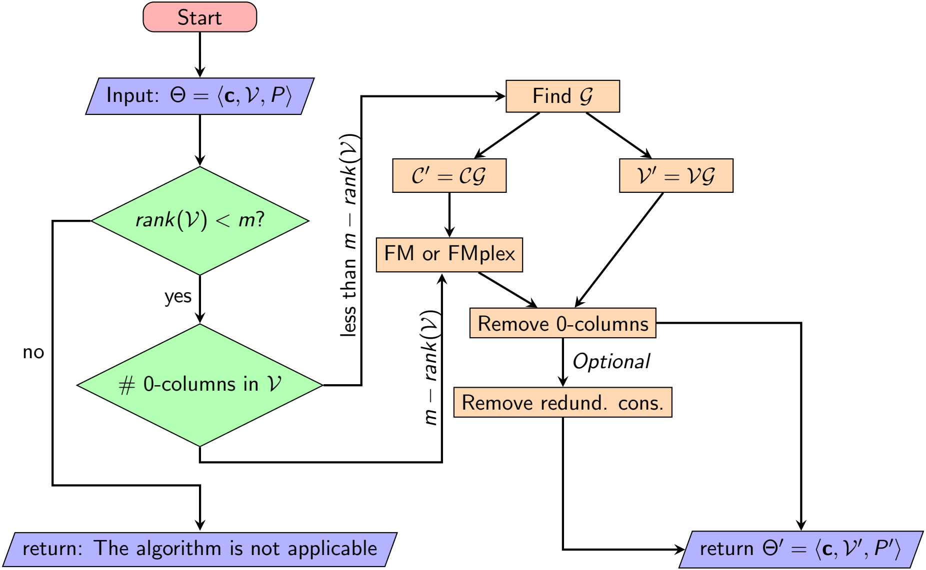



Start
Input: $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$
$rank(\mathcal{V}) < m$?
yes
no
0-columns in $\mathcal{V}$
less than $m - rank(\mathcal{V})$
Find $\mathcal{G}$
$m - rank(\mathcal{V})$
FM or FMplex
Remove 0-columns
Optional
Remove redund. cons.
return: The algorithm is not applicable
return $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle$

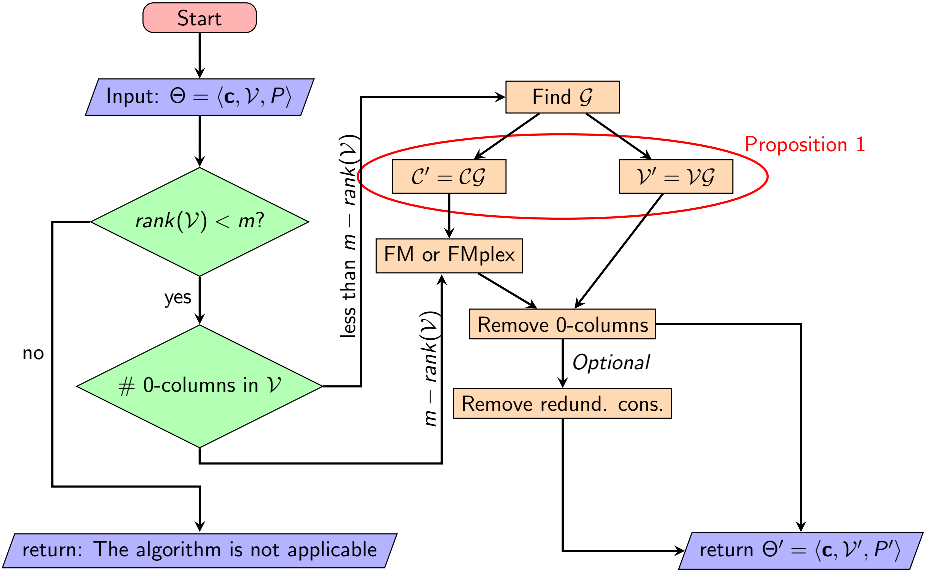
The algorithm



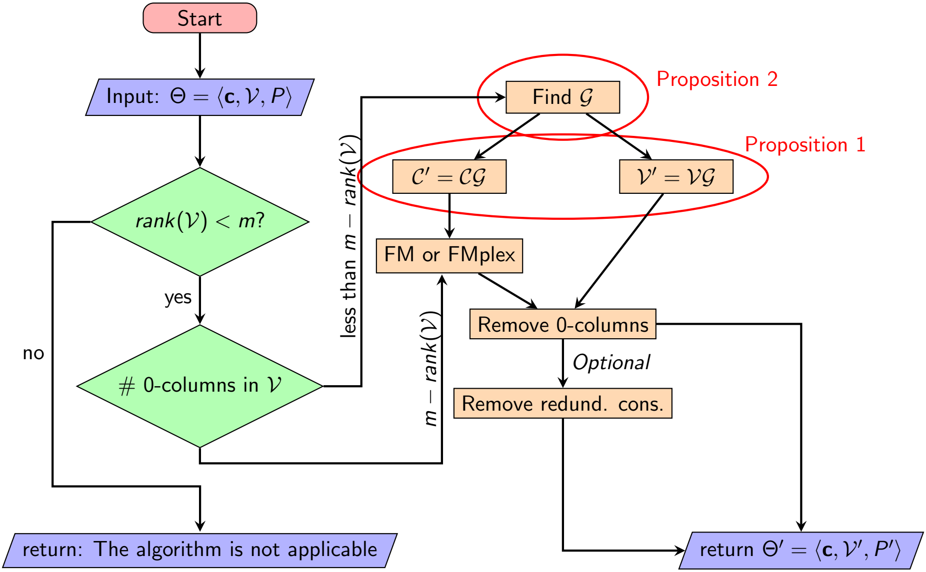
The algorithm



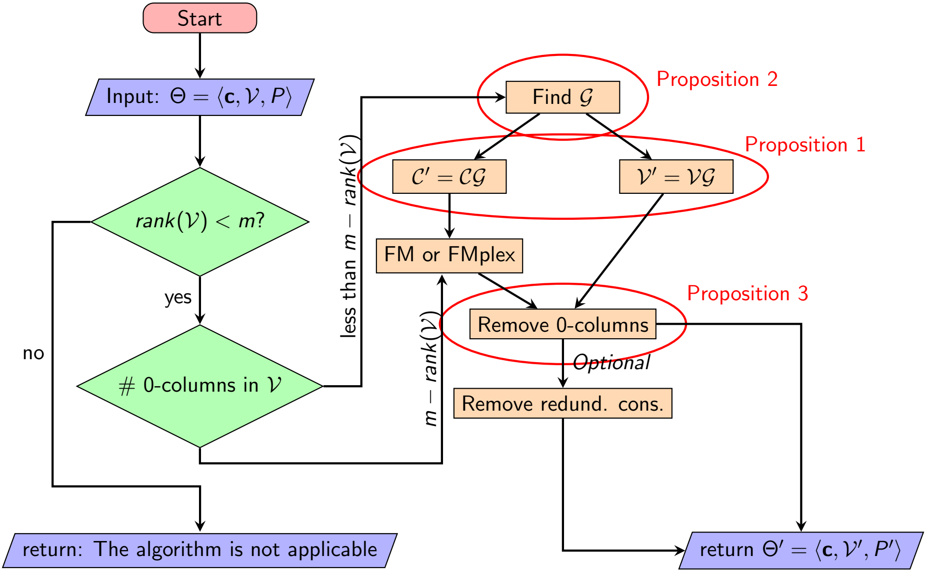
The algorithm



The algorithm



The algorithm



How to find $\mathcal{G}$?

- $\mathcal{V}\mathcal{G} = \mathcal{V}' : \ \mathcal{V} \in \mathbb{R}^{n \times m}, \ \mathcal{G} \in \mathbb{R}^{m \times m}, \ \mathcal{V}' \in \mathbb{R}^{n \times m}$

How to find $\mathcal{G}$?

- $\mathcal{V}\mathcal{G} = \mathcal{V}' : \ \mathcal{V} \in \mathbb{R}^{n\times m}, \ \mathcal{G} \in \mathbb{R}^{m\times m}, \ \mathcal{V}' \in \mathbb{R}^{n\times m}$

$$\begin{bmatrix} v_1^1 & v_1^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & v_n^j & v_n^m \end{bmatrix} \cdot \begin{bmatrix} g_1^1 & \cdots & g_1^m \\ \vdots & \ddots & \vdots \\ g_m^1 & \cdots & g_m^m \end{bmatrix} = \begin{bmatrix} v_1^1 & 0^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & 0^j & v_n^m \end{bmatrix}$$

How to find $\mathcal{G}$?

- $\mathcal{V}\mathcal{G} = \mathcal{V}' : \ \mathcal{V} \in \mathbb{R}^{n \times m}, \ \mathcal{G} \in \mathbb{R}^{m \times m}, \ \mathcal{V}' \in \mathbb{R}^{n \times m}$

$$\begin{bmatrix} v_1^1 & v_1^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & v_n^j & v_n^m \end{bmatrix} \cdot \begin{bmatrix} g_1^1 & \cdots & g_1^m \\ \vdots & \ddots & \vdots \\ g_m^1 & \cdots & g_m^m \end{bmatrix} = \begin{bmatrix} v_1^1 & 0^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & 0^j & v_n^m \end{bmatrix}$$

- $\mathcal{G}$ is $\mathcal{I}_m$, besides the $j-$th column, which needs to fulfil:

$$\mathcal{V}\mathbf{g^j} = \mathbf{0}.$$

How to find $\mathcal{G}$?

- $\mathcal{V}\mathcal{G} = \mathcal{V}' : \ \mathcal{V} \in \mathbb{R}^{n\times m}, \ \mathcal{G} \in \mathbb{R}^{m\times m}, \ \mathcal{V}' \in \mathbb{R}^{n\times m}$

$$\begin{bmatrix} v_1^1 & v_1^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & v_n^j & v_n^m \end{bmatrix} \cdot \begin{bmatrix} g_1^1 & \cdots & g_1^m \\ \vdots & \ddots & \vdots \\ g_m^1 & \cdots & g_m^m \end{bmatrix} = \begin{bmatrix} v_1^1 & 0^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & 0^j & v_n^m \end{bmatrix}$$

- $\mathcal{G}$ is $\mathcal{I}_m$, besides the $j-$th column, which needs to fulfil:
$$\mathcal{V}\mathbf{g^j} = \mathbf{0}.$$
- Choose $\mathbf{g^j}$ from the basis of the kernel of $\mathcal{V}$.

How to find $\mathcal{G}$?

- $\mathcal{V}\mathcal{G} = \mathcal{V}' : \ \mathcal{V} \in \mathbb{R}^{n\times m}, \ \mathcal{G} \in \mathbb{R}^{m\times m}, \ \mathcal{V}' \in \mathbb{R}^{n\times m}$

$$\begin{bmatrix} v_1^1 & v_1^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & v_n^j & v_n^m \end{bmatrix} \cdot \begin{bmatrix} g_1^1 & \cdots & g_1^m \\ \vdots & \ddots & \vdots \\ g_m^1 & \cdots & g_m^m \end{bmatrix} = \begin{bmatrix} v_1^1 & 0^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & 0^j & v_n^m \end{bmatrix}$$

- $\mathcal{G}$ is $\mathcal{I}_m$, besides the $j-$th column, which needs to fulfil:

$$\mathcal{V}\mathbf{g^j} = \mathbf{0}.$$

- Choose $\mathbf{g^j}$ from the basis of the kernel of $\mathcal{V}$.
- The basis has **exactly** $m - rank(\mathcal{V})$ columns.

How to find $\mathcal{G}$?

- $\mathcal{V}\mathcal{G} = \mathcal{V}' : \ \mathcal{V} \in \mathbb{R}^{n\times m}, \ \mathcal{G} \in \mathbb{R}^{m\times m}, \ \mathcal{V}' \in \mathbb{R}^{n\times m}$

$$\begin{bmatrix} v_1^1 & v_1^j & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & v_n^j & v_n^m \end{bmatrix} \cdot \begin{bmatrix} g_1^1 & \cdots & g_1^m \\ \vdots & \ddots & \vdots \\ g_m^1 & \cdots & g_m^m \end{bmatrix} = \begin{bmatrix} v_1^1 & {\color{purple}0^j} & v_1^m \\ \vdots & \vdots & \vdots \\ v_n^1 & {\color{purple}0^j} & v_n^m \end{bmatrix}$$

- $\mathcal{G}$ is $\mathcal{I}_m$, besides the $j-$th column, which needs to fulfil:

$$\mathcal{V}\mathbf{g^j} = \mathbf{0}.$$

- Choose $\mathbf{g^j}$ from the basis of the kernel of $\mathcal{V}$.
- The basis has **exactly** $m - rank(\mathcal{V})$ columns.
- $\mathcal{G}$ is invertible.

Example I

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle \text{ and } P \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}, \text{ where:}$$

$$\mathbf{c} = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}, \ \mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix},$$

$$\mathcal{C} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -1 \\ \frac{-3\sqrt{2}+1}{10} & \frac{3\sqrt{2}-1}{10} & 1 \end{bmatrix}, \ \mathbf{d} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0.5 \\ \frac{3\sqrt{2}-1}{5} \end{bmatrix}$$

Example I

$$\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle \text{ and } P \triangleq \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}, \text{ where:}$$

$$\mathbf{c} = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}, \quad \mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix},$$

$$\mathcal{C} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -1 \\ \frac{-3\sqrt{2}+1}{10} & \frac{3\sqrt{2}-1}{10} & 1 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0.5 \\ \frac{3\sqrt{2}-1}{5} \end{bmatrix}$$

$$\text{rank}(\mathcal{V}) = 2 < 3$$

Example II

- Find $\mathcal{G}$.

$$\mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \xrightarrow{\qquad\qquad\qquad \cdot\mathcal{G} \qquad\qquad\qquad} \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & 0 & 0 \end{bmatrix} = \mathcal{V}'$$

Example II

- Find $\mathcal{G}$.

$$\mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \xrightarrow{\quad\quad\quad\quad \cdot\mathcal{G} \quad\quad\quad\quad} \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & 0 & 0 \end{bmatrix} = \mathcal{V}'$$

$$\mathcal{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example II

- Find $\mathcal{G}$.

$$\mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \xrightarrow{\quad\quad\quad\quad \cdot\mathcal{G} \quad\quad\quad\quad} \begin{bmatrix} 0 & \color{red}{0} & 1 \\ \frac{\sqrt{2}}{2} & \color{red}{0} & 0 \end{bmatrix} = \mathcal{V}'$$

$$\mathcal{G} = \begin{bmatrix} 1 & \color{red}{0} & 0 \\ 0 & \color{red}{1} & 0 \\ 0 & \color{red}{0} & 1 \end{bmatrix}$$

Example II

- Find $\mathcal{G}$.

$$\mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \xrightarrow{\quad\quad\quad\quad \cdot\mathcal{G} \quad\quad\quad\quad} \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & 0 & 0 \end{bmatrix} = \mathcal{V}'$$

$$\mathcal{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{K} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

Example II

- Find $\mathcal{G}$.

$$\mathcal{V} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \xrightarrow{\quad\quad\quad\quad \cdot\mathcal{G} \quad\quad\quad\quad} \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & 0 & 0 \end{bmatrix} = \mathcal{V}'$$

$$\mathcal{G} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example III

- Compute $\mathcal{C}'$.

$$\mathcal{C}\mathcal{G} = \mathcal{C}'$$

$$\begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -1 \\ \frac{-3\sqrt{2}+1}{10} & \frac{3\sqrt{2}-1}{10} & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ \frac{\sqrt{2}}{2} & -\sqrt{2} & -1 \\ -\frac{3\sqrt{2}-1}{10} & \frac{3\sqrt{2}-1}{5} & 1 \end{bmatrix}$$

Example IV

- Apply FM in the predicate to eliminate the linearly dependent variable.

$$\mathcal{C}' = \begin{bmatrix} 0 & \color{blue}{0} & 0 \\ 1 & \color{blue}{0} & 0 \\ 0.5 & \color{blue}{0} & \frac{5(3\sqrt{2}+1)}{17} \\ -1 & \color{blue}{0} & 0 \\ 0 & \color{blue}{0} & 0 \\ -0.5 & \color{blue}{0} & \frac{5(3\sqrt{2}+1)}{17} \\ -0.5 & \color{blue}{0} & -\frac{\sqrt{2}}{2} \\ 0.5 & \color{blue}{0} & -\frac{\sqrt{2}}{2} \\ 0 & \color{blue}{0} & \frac{13\sqrt{2}+10}{34} \end{bmatrix}, \quad \mathbf{d}' = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 2 \\ 2 \\ 2 \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \end{bmatrix}$$

Example V

- Remove 0-columns both in $\mathcal{V}'$ and $\mathcal{C}'$.

$$\mathcal{V}' = \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{2}}{2} & 0 \end{bmatrix}, \quad \mathcal{C}' = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0.5 & \frac{5(3\sqrt{2}+1)}{17} \\ -1 & 0 \\ 0 & 0 \\ -0.5 & \frac{5(3\sqrt{2}+1)}{17} \\ -0.5 & -\frac{\sqrt{2}}{2} \\ 0.5 & -\frac{\sqrt{2}}{2} \\ 0 & \frac{13\sqrt{2}+10}{34} \end{bmatrix}$$

Example VI (Optional)

- Remove redundant constraints in the predicate.

$$\mathcal{C}' = \begin{bmatrix} 1 & 0 \\ 0.5 & \frac{5(3\sqrt{2}+1)}{17} \\ -1 & 0 \\ -0.5 & \frac{5(3\sqrt{2}+1)}{17} \\ 0.5 & -\frac{\sqrt{2}}{2} \\ 0 & \frac{13\sqrt{2}+10}{34} \end{bmatrix}, \quad \mathbf{d}' = \begin{bmatrix} 3 \\ 3 \\ 2 \\ 2 \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \end{bmatrix}$$

Example VII

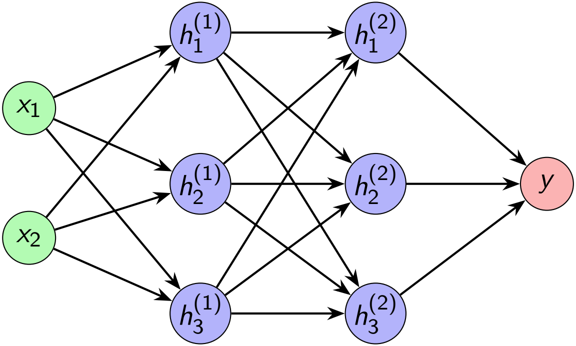
- Return: $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle$; $P' \triangleq \mathcal{C}'\boldsymbol{\alpha'} \leq \mathbf{d}'$, such that:

$$\mathbf{c} = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}, \quad \mathcal{V}' = \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{2}}{2} & 0 \end{bmatrix},$$

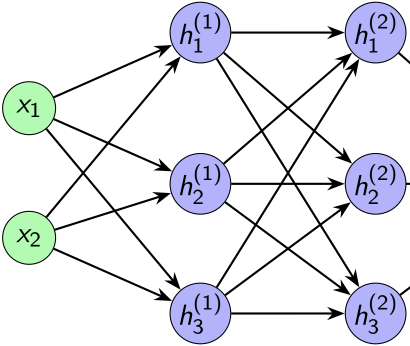
$$\mathcal{C}' = \begin{bmatrix} 1 & 0 \\ 0.5 & \frac{5(3\sqrt{2}+1)}{17} \\ -1 & 0 \\ -0.5 & \frac{5(3\sqrt{2}+1)}{17} \\ 0.5 & -\frac{\sqrt{2}}{2} \\ 0 & \frac{13\sqrt{2}+10}{34} \end{bmatrix}, \quad \mathbf{d}' = \begin{bmatrix} 3 \\ 3 \\ 2 \\ 2 \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \\ \frac{1+2\sqrt{2}}{2\sqrt{2}} \end{bmatrix}$$

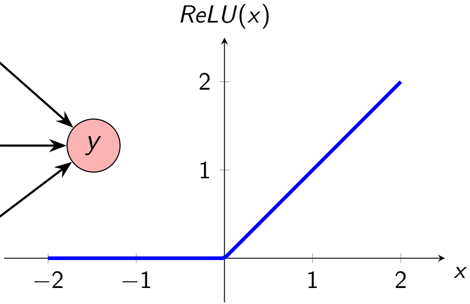
Table of contents

Feedforward neural network

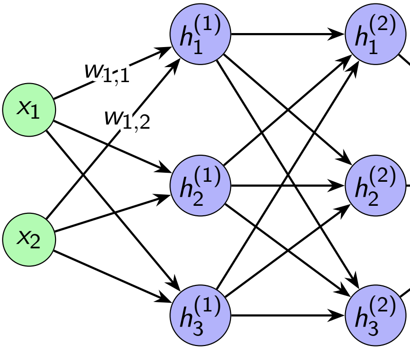


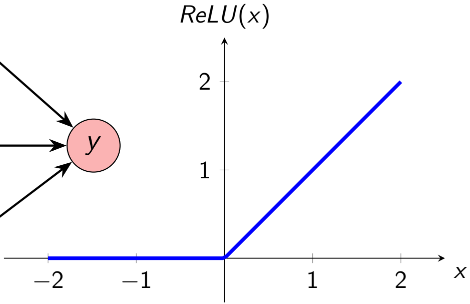
Feedforward neural network



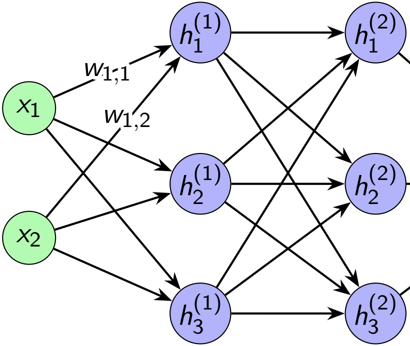


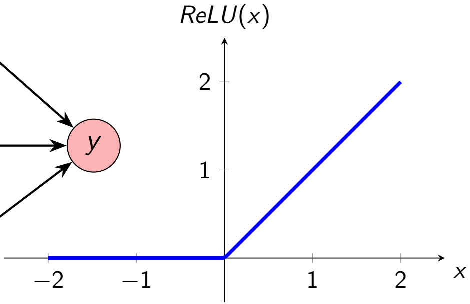
Feedforward neural network





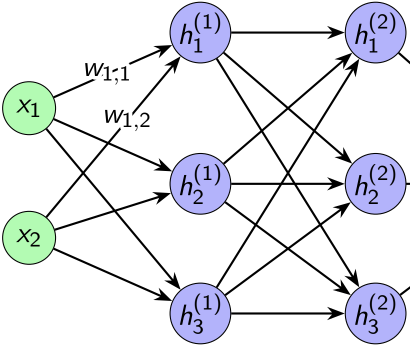
Feedforward neural network

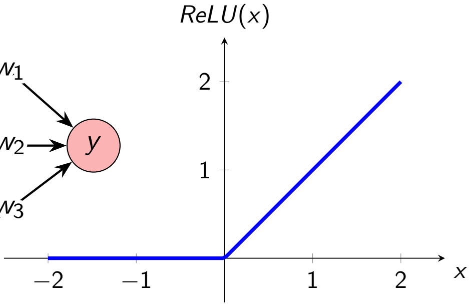




$$h_1^{(1)} = ReLU(w_{1,1} \cdot x_1 + w_{1,2} \cdot x_2 + b_1^{(1)})$$

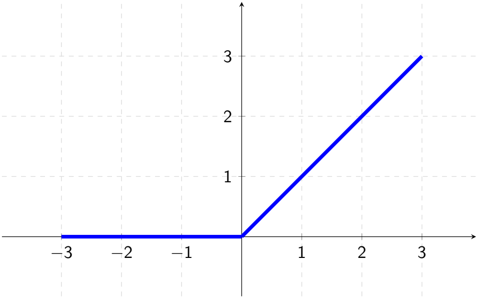
Feedforward neural network



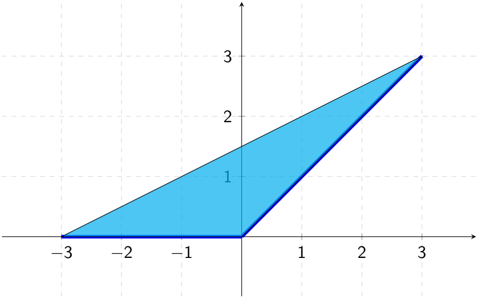


$$h_1^{(1)} = ReLU(w_{1,1} \cdot x_1 + w_{1,2} \cdot x_2 + b_1^{(1)})$$
$$y = ReLU(w_1 \cdot h_1^{(2)} + w_2 \cdot h_2^{(2)} + w_3 \cdot h_3^{(2)} + b_3^{(1)})$$

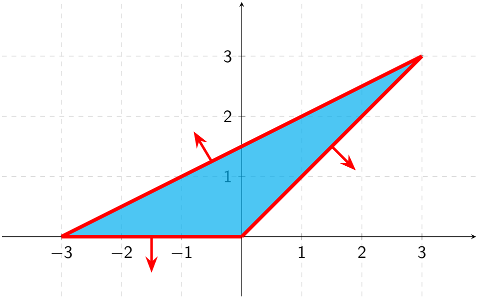
The over-approximate method [Tran et al., 2019]



The over-approximate method [Tran et al., 2019]



The over-approximate method [Tran et al., 2019]



Add a new variable in the predicate to describe these three new inequalities.

Over-approximation analysis cases

Notation: l and u are the lower- and upper bounds of x_i.

Over-approximation analysis cases

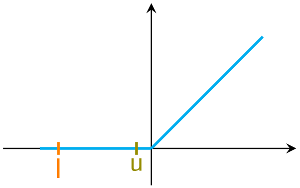
Notation: l and u are the lower- and upper bounds of x_i.

3 cases

Over-approximation analysis cases

Notation: l and u are the lower- and upper bounds of x_i.

3 cases → $u \leq 0 \;\rightarrow$ projection

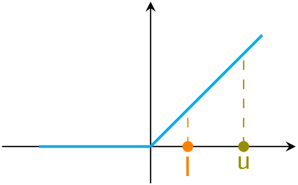


Over-approximation analysis cases

Notation: l and u are the lower- and upper bounds of x_i.

3 cases → $u \leq 0 \rightarrow$ projection

3 cases → $l \geq 0 \rightarrow$ identity

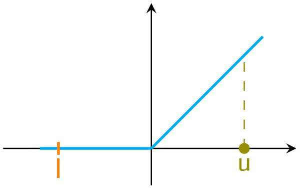


Over-approximation analysis cases

Notation: l and u are the lower- and upper bounds of x_i.

3 cases

- u ≤ 0 $\rightarrow$ projection
- l ≥ 0 $\rightarrow$ identity
- l $\leq 0 \leq$ u $\rightarrow$ over-app.

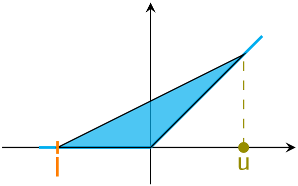


Over-approximation analysis cases

Notation: l and u are the lower- and upper bounds of x_i.

3 cases

- $u \leq 0 \rightarrow$ projection
- $l \geq 0 \rightarrow$ identity
- $l \leq 0 \leq u \rightarrow$ over-app.



Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \left[\begin{array}{c:c} \quad\quad\quad\quad & \quad\quad \\ \hdashline & \\ \hdashline & \end{array}\right]$$

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\begin{array}{l}
\\
\mathcal{V}_{res} =
\end{array}
\begin{array}{cc}
 & \overbrace{\phantom{\mathcal{I}_j}}^{j} \\
\left[\begin{array}{c|c}
0 & \mathcal{I}_j \\
\hline
 & \\
\hline
 &
\end{array}\right] &
\end{array}
\begin{array}{l}
\}\text{j} \\
\\
\\
\end{array}$$

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{cc} \begin{array}{cc} \overbrace{\qquad\quad}^{m} & \overbrace{\qquad}^{j} \end{array} & \\ \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}_{slice} & 0 \\ \hline & \end{array}\right] & \begin{array}{l} \}\, j \\ \}\, \text{n-i-j} \\ \end{array} \end{array}$$

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{c} \begin{array}{cc} \overbrace{\hphantom{\mathcal{V}_{slice}}}^{m} & \overbrace{\hphantom{0}}^{j} \end{array} \\ \left[\begin{array}{c:c} 0 & \mathcal{I}_j \\ \hdashline \mathcal{V}_{slice} & 0 \\ \hdashline 0 & 0 \end{array}\right] \end{array} \begin{array}{l} \\ \}\text{j} \\ \}\text{n-i-j} \\ \}\text{i} \end{array}$$

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{c} \begin{array}{cc} \overbrace{\phantom{\mathcal{V}_{slice}}}^{m} & \overbrace{\phantom{\mathcal{I}_j}}^{j} \end{array} \\ \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}_{slice} & 0 \\ \hline 0 & 0 \end{array}\right] \end{array} \begin{array}{l} \}\text{j} \\ \}\text{n-i-j} \\ \}\text{i} \end{array}$$

$$\textbf{rank}(\mathcal{V}_{res}) = \textbf{rank}(\mathcal{V}_{slice}) + \textbf{rank}(\mathcal{I}_j) = \textbf{rank}(\mathcal{V}_{slice}) + j$$

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{c} \begin{array}{cc} \overbrace{\qquad\quad}^{m} & \overbrace{\qquad}^{j} \end{array} \\ \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}_{slice} & 0 \\ \hline 0 & 0 \end{array}\right] \end{array} \begin{array}{l} \\ \}\text{j} \\ \}\text{n-i-j} \\ \}\text{i} \end{array}$$

$$\mathbf{rank}(\mathcal{V}_{res}) = \mathbf{rank}(\mathcal{V}_{slice}) + \mathbf{rank}(\mathcal{I}_j) = \mathbf{rank}(\mathcal{V}_{slice}) + j$$

- The condition for our method to be applicable is:

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{cc} \begin{array}{cc} \overbrace{\hphantom{\mathcal{V}_{slice}}}^{m} & \overbrace{\hphantom{\mathcal{I}_j}}^{j} \end{array} & \\ \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}_{slice} & 0 \\ \hline 0 & 0 \end{array}\right] & \begin{array}{l} \left.\right\} \text{j} \\ \left.\right\} \text{n-i-j} \\ \left.\right\} \text{i} \end{array} \end{array}$$

$$\textbf{rank}(\mathcal{V}_{res}) = \textbf{rank}(\mathcal{V}_{slice}) + \textbf{rank}(\mathcal{I}_j) = \textbf{rank}(\mathcal{V}_{slice}) + j$$

- The condition for our method to be applicable is:

$$\textbf{rank}(\mathcal{V}_{slice}) < m$$

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{c} \begin{array}{cc} \overbrace{\hphantom{\mathcal{V}_{slice}}}^{m} & \overbrace{\hphantom{\mathcal{I}_j}}^{j} \end{array} \\ \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}_{slice} & 0 \\ \hline 0 & 0 \end{array}\right] \end{array} \begin{array}{l} \}\text{j} \\ \}\text{n-i-j} \\ \}\text{i} \end{array}$$

$$\textbf{rank}(\mathcal{V}_{res}) = \textbf{rank}(\mathcal{V}_{slice}) + \textbf{rank}(\mathcal{I}_j) = \textbf{rank}(\mathcal{V}_{slice}) + j$$

- The condition for our method to be applicable is:

$$\textbf{rank}(\mathcal{V}_{slice}) < m$$

- Compute $\mathcal{G}^*$, such that $\mathcal{V}_{slice}\mathcal{G}^* = \mathcal{V}'_{slice}$.

Generator matrix at the end of the layer

- Assumption: n steps in total, i projections, j over-approximations.

$$\mathcal{V}_{res} = \begin{array}{cl} \begin{array}{cc} \overbrace{\qquad\quad}^{m} & \overbrace{\qquad}^{j} \end{array} & \\ \left[\begin{array}{c:c} 0 & \mathcal{I}_j \\ \hdashline \mathcal{V}_{slice} & 0 \\ \hdashline 0 & 0 \end{array}\right] & \begin{array}{l} \}\,\text{j} \\ \}\,\text{n-i-j} \\ \}\,\text{i} \end{array} \end{array}$$

$$\mathbf{rank}(\mathcal{V}_{res}) = \mathbf{rank}(\mathcal{V}_{slice}) + \mathbf{rank}(\mathcal{I}_j) = \mathbf{rank}(\mathcal{V}_{slice}) + j$$

- The condition for our method to be applicable is:

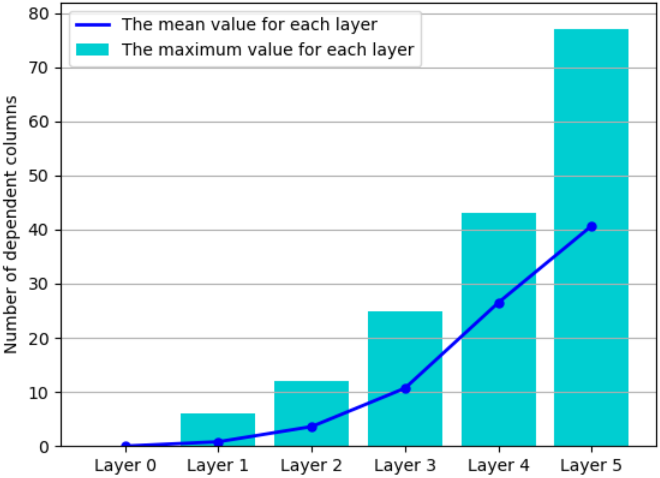
$$\mathbf{rank}(\mathcal{V}_{slice}) < m$$

- Compute $\mathcal{G}^*$, such that $\mathcal{V}_{slice}\mathcal{G}^* = \mathcal{V}'_{slice}$.
- Reconstruct $\mathcal{G}$ using $\mathcal{G}^*$, such that: $\mathcal{V}_{res}\mathcal{G} = \mathcal{V}'_{res}$.

Table of contents

How many times is the condition met?

- ACAS Xu benchmark [Katz et al., 2017].



The performance of FMplex on ACAS Xu

$\mathbf{N^p_{x,y}}$	**Row**	**Col**	**LinDep**	**Time(s)**	**Constraints**
$N^3_{3,1}$	34	13	1	0.022	224
$N^4_{1,9}$	55	20	1	0.408	577
$N^3_{1,2}$	58	21	2	1.333	4797
$N^3_{5,5}$	91	32	2	37.860	16770
$N^3_{2,9}$	34	13	3	1.739	4385
$N^3_{4,1}$	67	24	3	533.361	58984
$N^3_{1,1}$	73	26	3	283	88891
$N^4_{4,7}$	88	31	3	T	-
$N^4_{4,4}$	49	18	4	55.418	54163
$N^3_{2,8}$	55	20	4	114.399	73761
$N^4_{3,7}$	52	19	4	240.006	82496
$N^3_{2,6}$	76	27	4	T	-
$N^3_{2,7}$	52	19	5	460.166	164203
$N^3_{3,9}$	34	13	6	103.906	33458

The performance of FMplex on ACAS Xu

$\mathbf{N^p_{x,y}}$	Row	Col	LinDep	Time(s)	Constraints
$N^3_{3,1}$	34	13	1	0.022	224
$N^4_{1,9}$	55	20	1	0.408	577
$N^3_{1,2}$	58	21	2	1.333	4797
$N^3_{5,5}$	91	32	2	37.860	16770
$N^3_{2,9}$	34	13	3	1.739	4385
$N^3_{4,1}$	67	24	3	533.361	58984
$N^3_{1,1}$	73	26	3	283	88891
$N^4_{4,7}$	88	31	3	T	-
$N^4_{4,4}$	49	18	4	55.418	54163
$N^3_{2,8}$	55	20	4	114.399	73761
$N^4_{3,7}$	52	19	4	240.006	82496
$N^3_{2,6}$	76	27	4	T	-
$N^3_{2,7}$	52	19	5	460.166	164203
$N^3_{3,9}$	34	13	6	103.906	33458

The performance of FMplex on ACAS Xu

$\mathbf{N^p_{x,y}}$	Row	Col	LinDep	Time(s)	Constraints
$N^3_{3,1}$	34	13	1	0.022	224
$N^4_{1,9}$	55	20	1	0.408	577
$N^3_{1,2}$	58	21	2	1.333	4797
$N^3_{5,5}$	91	32	2	37.860	16770
$N^3_{2,9}$	34	13	3	1.739	4385
$N^3_{4,1}$	67	24	3	533.361	58984
$N^3_{1,1}$	73	26	3	283	88891
$N^4_{4,7}$	88	31	3	T	-
$N^4_{4,4}$	49	18	4	55.418	54163
$N^3_{2,8}$	55	20	4	114.399	73761
$N^4_{3,7}$	52	19	4	240.006	82496
$N^3_{2,6}$	76	27	4	T	-
$N^3_{2,7}$	52	19	5	460.166	164203
$N^3_{3,9}$	34	13	6	103.906	33458

Comparison of FM and FMplex on drones benchmark

	Row	Col	LinDep	FMplex	FM1	FM2
$AC5_2$	30	14	4	4268	129	3153
$AC4_2$	57	23	4	214859	608	86147
$AC2_2$	33	15	5	20486	153	4688
$AC6_2$	30	14	6	18932	109	2215
$AC1_1$	69	27	6	T	509	53869
$AC7_2$	42	18	9	T	161	4840
$AC5_1$	81	31	9	T	564	64572
$AC8_2$	49	19	13	T	134	2939
$AC1_2$	42	18	11	T	131	2920
$AC2_1$	105	39	29	T	1004	221277
$AC6_1$	105	39	29	T	1034	232321
$AC7_1$	159	57	43	T	2461	Killed
$AC3_1$	231	81	51	T	5336	Killed

Comparison of FM and FMplex on drones benchmark

	Row	Col	LinDep	FMplex	FM1	FM2
$AC5_2$	30	14	4	4268	129	3153
$AC4_2$	57	23	4	214859	608	86147
$AC2_2$	33	15	5	20486	153	4688
$AC6_2$	30	14	6	18932	109	2215
$AC1_1$	69	27	6	T	509	53869
$AC7_2$	42	18	9	T	161	4840
$AC5_1$	81	31	9	T	564	64572
$AC8_2$	49	19	13	T	134	2939
$AC1_2$	42	18	11	T	131	2920
$AC2_1$	105	39	29	T	1004	221277
$AC6_1$	105	39	29	T	1034	232321
$AC7_1$	159	57	43	T	2461	Killed
$AC3_1$	231	81	51	T	5336	Killed

Number of redundant constr. on thermostat benchmark I

ID	Row	Col	LinDep	Constraints	Redundant
1	4	2	1	2	0
2	7	3	1	8	2
3	10	4	1	16	4
4	13	5	1	26	5
5	13	5	1	26	6
6	10	4	1	16	10
7	19	7	1	52	18
8	29	9	1	86	20
9	31	11	1	128	27
10	31	11	1	128	29
11	31	11	1	128	30
12	31	11	1	128	101
13	31	11	1	128	-

Number of redundant constr. on thermostat benchmark I

ID	Row	Col	LinDep	Constraints	Redundant
1	4	2	1	2	0
2	7	3	1	8	2
3	10	4	1	16	4
4	13	5	1	26	5
5	13	5	1	26	6
6	10	4	1	16	10
7	19	7	1	52	18
8	29	9	1	86	20
9	31	11	1	128	27
10	31	11	1	128	29
11	31	11	1	128	30
12	31	11	1	128	101
13	31	11	1	128	-

Number of redundant constr. on thermostat benchmark II

ID	Row	Col	LinDep	Constraints	Redundant
14	16	6	2	150	-
15	19	7	2	205	-
16	22	8	2	291	-
17	22	8	3	791	-
18	69	9	4	677463	-
19	28	10	4	4372	-
20	22	8	4	1797	-
21	25	9	4	2871	-
22	28	10	5	8987	-
23	31	11	5	14415	-
24	39	14	7	134571	-

Table of contents

Discussion

Discussion

- Non finishing analysis of the network.
- Exponential number of returned constraints.
- Redundancies not always detectable.
- Unbounded values causing errors while solving LP instances.
- Bad running time.

Discussion

- Non finishing analysis of the network.
- Exponential number of returned constraints.
- Redundancies not always detectable.
- Unbounded values causing errors while solving LP instances.
- Bad running time.
- Reduced number of variables in the equivalent star set.
- Test FMplex on star-based benchmarks.
- Efficient and customizable star set transformation without elimination.

Future work

Future work

1. Redundancy check **during** FMplex.
2. Bind the numbers **during** FMplex and network analysis.

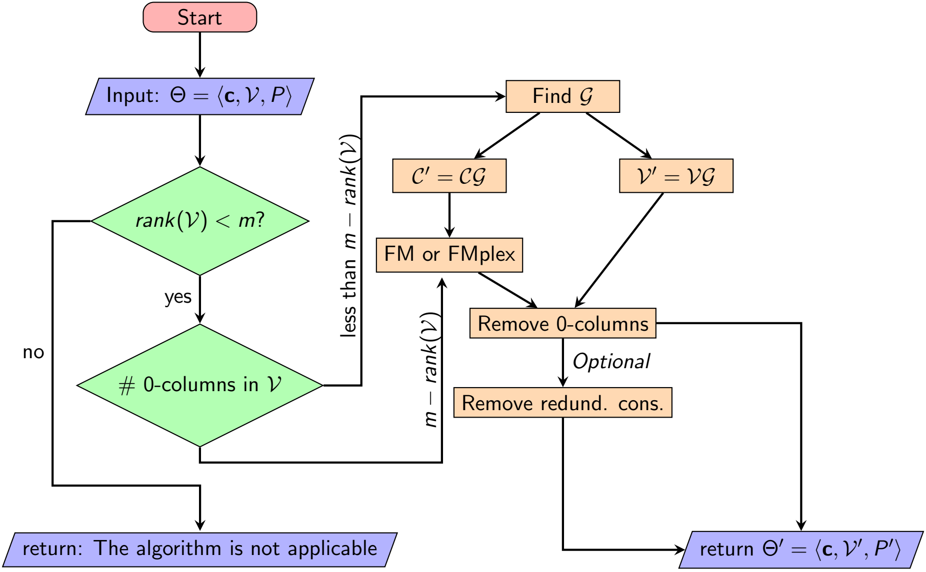
Future work

1. Redundancy check **during** FMplex.
2. Bind the numbers **during** FMplex and network analysis.
3. If complete analysis possible:
    - Violating dimensions: use for adversarial training.

Future work

1. Redundancy check **during** FMplex.
2. Bind the numbers **during** FMplex and network analysis.
3. If complete analysis possible:
   - Violating dimensions: use for adversarial training.
4. Apply the method after (some) projection steps during the layer reachability.

Thank you for your attention!



Star set transformation

Proposition 1

Any star set $\Theta = \langle \mathbf{c}, \mathcal{V}, P \rangle$, $P \triangleq (\mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d})$ can be transformed to another star set $\Theta' = \langle \mathbf{c}, \mathcal{V}', P' \rangle$, $P' \triangleq (\mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d}')$, such that $\mathcal{V} \neq \mathcal{V}', \mathcal{C} \neq \mathcal{C}'$ and $\Theta \equiv \Theta'$. For the transformation, we use a manually designed invertible matrix $\mathcal{G}$, such that $\mathcal{V}\mathcal{G} = \mathcal{V}'$ and $\mathcal{C}\mathcal{G} = \mathcal{C}'$.

$$
\begin{aligned}
\Theta &= \{\mathbf{x} = \mathbf{c} + \mathcal{V}\boldsymbol{\alpha} && \mid \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}\} \\
&= \{\mathbf{x} = \mathbf{c} + \underbrace{\mathcal{V}\mathcal{G}}_{=\mathcal{V}'} \underbrace{\mathcal{G}^{-1}\boldsymbol{\alpha}}_{=\boldsymbol{\alpha}'} && \mid \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}\} \\
&= \{\mathbf{x} = \mathbf{c} + \mathcal{V}'\boldsymbol{\alpha}' && \mid \mathcal{C}\boldsymbol{\alpha} \leq \mathbf{d}\} \qquad \underbrace{\mathcal{G}\mathcal{G}^{-1}}_{\mathcal{I}_m}\boldsymbol{\alpha} = \mathcal{G}\boldsymbol{\alpha}' \Rightarrow \boldsymbol{\alpha} = \mathcal{G}\boldsymbol{\alpha}' \\
&= \{\mathbf{x} = \mathbf{c} + \mathcal{V}'\boldsymbol{\alpha}' && \mid \underbrace{\mathcal{C}\mathcal{G}}_{=\mathcal{C}'}\boldsymbol{\alpha}' \leq \mathbf{d}\} \\
&= \{\mathbf{x} = \mathcal{V}'\boldsymbol{\alpha}' + \mathbf{c} && \mid \mathcal{C}'\boldsymbol{\alpha}' \leq \mathbf{d}\} = \Theta'
\end{aligned}
$$

Invertability of $\mathcal{G}$ I

Find an invertible matrix $\mathcal{G}$, such that $\mathcal{V}\mathcal{G} = \mathcal{V}'$ and $\mathcal{V}'$ has the desired property.
Proof: Let J be the set of linearly dependent and I linearly independent columns in $\mathcal{V}$. $\mathcal{G} = [\,\mathbf{g_1}, \ldots, \mathbf{g_m}\,],\ \mathbf{g_i} \in \mathbb{R}^m,\ 1 \leq i \leq m$, such that:

$$\mathbf{g_i} := \begin{cases} \mathbf{k_i}, & i \in J, \\ \mathbf{e_i}, & i \in I \end{cases}$$

Prove: the columns of $\mathcal{G}$ linearly independent. This means, for $\alpha_j, \beta_i \in \mathbb{R}$ and $j \in J, i \in I$:

$$\sum_{j \in J} \alpha_j \mathbf{k_j} + \sum_{i \in I} \beta_i \mathbf{e_i} = \mathbf{0} \implies \alpha_j = 0,\ \beta_i = 0\ \forall j \in J, i \in I.$$

Invertability of $\mathcal{G}$ II

$$\sum_{j\in J}\alpha_j\mathbf{k_j}+\sum_{i\in I}\beta_i\mathbf{e_i}=\mathbf{0}\implies \alpha_j=0,\ \beta_i=0\ \forall j\in J, i\in I.$$

Assume, the linear combination equals $\mathbf{0}$. By multiplying the linear combination by the matrix $\mathcal{V}$ we obtain:

$$\mathcal{V}\left(\sum_{j\in J}\alpha_j\mathbf{k_j}+\sum_{i\in I}\beta_i\mathbf{e_i}\right)=\mathbf{0}$$

$$\iff \sum_{j\in J}\alpha_j\underbrace{\mathcal{V}\mathbf{k_j}}_{=\mathbf{0}}+\sum_{i\in I}\beta_i\mathcal{V}\mathbf{e_i}=\mathbf{0}\implies \sum_{i\in I}\beta_i\mathcal{V}\mathbf{e_i}=\mathbf{0}.$$

I is the set of indices of linearly independent columns of $\mathcal{V}$ and $\mathcal{V}\mathbf{e_i}$ is the $i-th$ column of $\mathcal{V}$. Therefore, $\beta_i=0\ \forall i\in I$. That means:

$$\sum_{j\in J}\alpha_j\mathbf{k_j}+\sum_{i\in I}\beta_i\mathbf{e_i}=\sum_{j\in J}\alpha_j\mathbf{k_j}=\mathbf{0}$$

From the linear independence of the vectors $\mathbf{k_j}$, $j\in J$ follows, that:

$$\alpha_j=0:\ \forall j\in J.$$

Construction of $\mathcal{G}$ out of $\mathcal{G}^*$

$$\mathcal{G} = \left[\begin{array}{c|c} \mathcal{G}^* & 0 \\ \hline 0 & \mathcal{I}_j \end{array}\right]$$

$$\mathcal{V}\mathcal{G} = \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}_{\text{slice}} & 0 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{c|c} \mathcal{G}^* & 0 \\ \hline 0 & \mathcal{I}_j \end{array}\right] =$$

$$\left[\begin{array}{c|c} 0 \cdot \mathcal{G}^* + \mathcal{I}_j \cdot 0 & 0 \cdot 0 + \mathcal{I}_j \cdot \mathcal{I}_j \\ \mathcal{G}^* \cdot \mathcal{V}_{\text{slice}} + 0 \cdot 0 & \mathcal{V}_{\text{slice}} \cdot 0 + 0 \cdot \mathcal{I}_j \\ 0 \cdot \mathcal{G}^* + 0 \cdot 0 & 0 \cdot 0 + 0 \cdot \mathcal{I}_j \end{array}\right] = \left[\begin{array}{c|c} 0 & \mathcal{I}_j \\ \hline \mathcal{V}'_{\text{slice}} & 0 \\ \hline 0 & 0 \end{array}\right]$$