Fourier -Motzkin

Eine kleine Zusammenfassung

dimension_reduction(A,b)

Eliminiert die letzte Spalte von A $\in \mathbb{R}^{m \times n}$

```
Input: A, b
Output: A', b'

For i=1,…,m:
   If A[i,n] == 0:
      A'.add_row(A[i])
      b'.add_row(b[i])
      continue
   For j=i+1,…,m:
      If sign(A[i,n]) != sign(A[j,n]):
         A'.add_row(1/A[i,n] * A[i] - 1/A[j,n] * A[j])
         B'.add_row(1/A[i,n] * b[i] - 1/A[j,n] * b[j])
A' = A'[:,:-1]
Return A', b'
```

dimension_reduction(A,b)

Man kann A' und b' auch durch Linksmultiplikation mit einer Matrix M erhalten.

$A' = M \cdot A$ und $b' = M \cdot b$

Die Matrix M wird so konstruiert:

Entsteht die k-te Zeile von A' durch $\frac{1}{\lambda}$A[i] + $\frac{1}{\mu}$A[j] so ist

$$M[k,i] = \frac{1}{\lambda}, M[k,j] = \frac{1}{\mu} \text{ und } M[k,l] = 0 \text{ sonst}$$

__image(A,M)

A ist die Matrix, die das Polyhedron P beschreibt und M die lineare Abbildung, die A transformiert.

$$M \cdot A = \{y \mid \exists x : y = Mx, \underbrace{Ax \geq b}_{x \in P}\}$$

$$M \cdot A = \pi(\{(y, x) : Q \cdot \binom{y}{x} = b'\})$$

$$Q = \begin{pmatrix} -I & M \\ I & -M \\ 0 & A \end{pmatrix}, b' = \begin{pmatrix} 0 \\ 0 \\ b \end{pmatrix}$$

H_representation(x_1,...,x_k)

Berechne die konvexe Hülle von $x_1, \ldots, x_k$.

$$A = \begin{pmatrix} | & & | \\ x_1 & \ldots & x_k \\ | & & | \end{pmatrix} \in \mathbb{R}^{n\times k},\ P = \{x : Qx = b\}$$

$$Q = \begin{pmatrix} 1 & \ldots & 0 \\ 0 & \ddots & 0 \\ 0 & \ldots & 1 \\ 1 & \ldots & 1 \\ -1 & \ldots & -1 \end{pmatrix}, b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ -1 \end{pmatrix}$$

Die konvexe Hülle ist dann $A \cdot P$.

compute_x_or_y

Fall n=1: Sei $A = \mathbb{R}^{m\times 1}$. Gibt es ein $x \in \mathbb{R}$, sodass $Ax \geq b$?

Alle Einträge von $A = (a_1, \ldots, a_m)$ mit $a_i > 0$ legen eine ***untere Schranke*** für x fest.

Alle Einträge von $A = (a_1, \ldots, a_m)$ mit $a_i < 0$ legen eine ***obere Schranke*** für x fest.

Falls die untere Schranke größer als die untere Schranke ist, so gibt es keine Lösung.

Dann findet man ein $y \in \mathbb{R}^m$ mit $y^T A = 0$ und $y^T b > 0$. Setze dazu

$$y_l = 1, y_u = -\frac{a_l}{a_u}$$

und alle anderen Komponenten $y_i = 0$.

compute_x_or_y

Fall n>1: Um das y zu konstruieren, berechne $y = M^T y_{old}$.

Um das $x \in \mathbb{R}^{n+1}$ zu konstruieren, setze das alte $x' \in \mathbb{R}^n$ in $A \begin{pmatrix} x' \\ z \end{pmatrix} \geq b$ ein und finde für z eine passende Lösung (siehe Fall 1).