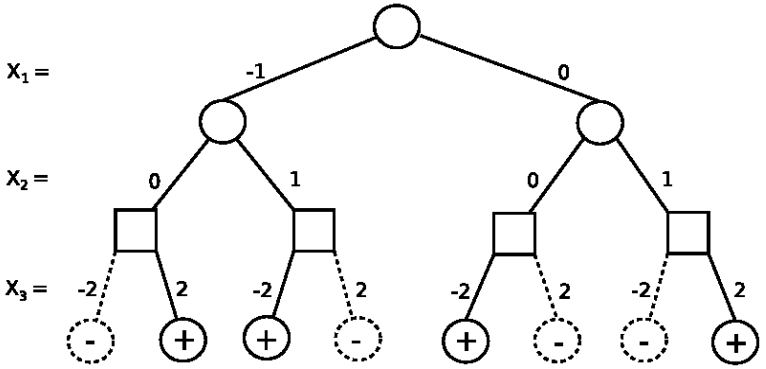
Definition1 (Strategie):



- Regelwerk für den Existenzspieler
- Teilbaum $(V_x \cup V_y, E, L)$ der Tiefe n
- Jede Kante $e_i \in E$ repräsentiert eine Belegung von Variable x_i auf ebene i des Baumes
- $l_i \in L$ repräsentiert den Wert von Variable x_i an Kante e_i
- Existenzknoten haben eine ausgehende Kante
- Allknoten haben zwei ausgehende Kanten

Eine Strategie ist eine **Gewinnstrategie** für den Existenzspieler, wenn alle Pfade von der Wurzel zu den Blättern einen Vektor x repräsentieren, so dass $Ax \leq b$ gilt.

Strategien sind auch Lösungen, wenn der Existenzspieler aus dem kontinuierlichen wählen darf.

Quantifizierte Lineare Programme:

TECHNISCHE UNIVERSITÄT DARMSTADT

Ein Algorithmus um Quantifizierte Lineare Programme zu lösen:

- Gegeben QLP Instanz $I=[Q:Ax\leq b]$
- Ziel: Elimination der letzten Variable x_n
- Wenn Quantifizierer $q_n\in\exists$, dann wende Fourier-Motzkin-Elimination an
- Quantifizierer $q_n\in\forall$, dann setze für jede Zeile einzeln worst-case ein
  - Interpretation als Spiel: Allspieler macht letzten Zug
  - Allspieler möchte $Ax\leq b$ mit diesem Zug verhindern
  - Variablenbelegung x_n muss mindestens zur Verletzung einer Ungleichung führen
  - Einsetzen des Worst-Case Wertes für jede Ungleichung

(Algorithmus wurde entdeckt von Subramani)

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Problemstellung QLP:

$$G : \exists x_1 \in [a_1, b_1]\ \forall y_1 \in [l_1, u_1]\ \exists x_2 \in [a_2, b_2] \ldots \exists x_n \in [a_n, b_n]\ \forall y_n \in [l_n, u_n]$$

$$A' \cdot \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{pmatrix} = \underbrace{A \cdot \begin{pmatrix} x \\ y \end{pmatrix}}_{\text{durch Umordnung}} \leq b, x \geq 0$$

„Entscheide, ob es ein x_1 gibt, so dass für alle y_1 es ein x_2 gibt u.s.w., so dass dass gegebene Ungleichungssystem eine gültige Lösung hat."

Dabei: $A \in \mathbb{Z}^{m \times 2n}$, $x,y \in \mathbb{Q}^n$, $b \in \mathbb{Z}^m$
(Vorsicht: nicht A, $a_1,..,a_n,b_1,...,b_n$, b) durcheinanderbringen. Bedeutung ist aus Zusammenhang entnehmbar.

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Problemstellung QLP:

- Das Paar (A,b) wird als Nebenbedingungssystem bezeichnet
- o.B.d.A sollen die Quantifizierer sich strikt abwechseln. Andernfalls könnte man zusätzliche Dummyvariablen einführen, die in (A,b) nicht vorkommen.
- Die Zeichenkette

$$\exists x_1 \in [a_1, b_1] \, \forall y_1 \in [l_1, u_1] \, \forall x_2 \in [a_2, b_2] \ldots \exists x_n \in [l_n, u_n] \, \forall y_n \in [a_n, b_n]$$

wird Quantifizierer-String Q(x,y) genannt. Die Länge von Q(x,y) wird mit |Q(A,b)| bezeichnet.

- Bedingungen an die ∃–Variablen können in die Matrix A kodiert werden, Bedingungen an die ∀–Variablen nicht.

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

QLP als Spiel:

- Wir können ein QLP als Spiel zwischen einem Spieler X, der die Existenz-variablen setzt und einem Spieler Y, der die All-Variablen setzt, auffassen.

- Die Züge werde alternierend ausgeführt.

- Ein *Spiel* (x,y) ist eine Abfolge von Zügen
  $x = [x_1, x_2,..., x_n]$
  $y = [y_1, y_2,..., y_n]$

- Eine *Strategie* für X bezeichnen wir mit $\underline{x}$ und eine Strategie für Y mit $\underline{y}$.
  $\underline{x} = [x_1, \quad x_2(x_1,y_1), \ldots \quad , x_n(x_1,y_1,x_2 \ldots y_{n-1})]$
  $\underline{y} = [y_1(x_1), y_2(x_1,y_1,x_2), \ldots, y_n(x_1,y_1,x_2 \ldots y_{n-1},x_n)]$
  Es gibt für jedes QLP entweder eine Gewinnstrategie für X oder eine für Y. (XOR)

Diese Art der Strategie wird meist, z.B. im Gebiet der Kontrolltheorie als *Politik* bezeichnet. Strategien sind bei uns eigentlich Bäume.

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Lösungsalgorithmus, QLP-Decide(A,b)

1: $A_{n+1} = A$; $b_{n+1} = b$;
2: for (i = n down to 2) do
3: (A_i,b_i) = Elim-Univ-Variable($A_{i+1},b_{i+1},y_i,l_i,u_i$);
4: (A_i,b_i) = Elim-Exist-Variable(A_i,b_i,x_i);
5: if (not Check-consistency()) then System is infeasable
6: Prune-Constraints()
7: (A_1,b_1) = Elim-Univ-Variable(A_2,b_2,y_1,l_1,u_1);
8: // The System is now reduced to a one-variable-system, that
9: // can easily be checked on consistency. Let the system have
10: // the form $x_1 \leq v$ and $x_1 \geq w$
11: if ($w \leq x_1 \leq v$ and $w \leq v$) then return System is feasible
12: else return System is infeasible

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Elim-Univ-Variable(A,b,y^n,l^n,u^n)

// beachte: Variablenindizierung hier oben statt unten

1: // Elimination der innersten Variable y_n (All-Variable).
2: for all constraints from i = 1 to m do
3: // assume the worst case for the ∃-Player
4: if ($a_{i,2n}l^n \leq a_{i,2n}u^n$) $y_i^n = u^n$
5: else $y_i^n = l^n$
6: build the new A and b by incorporating all constants to b

Beispiel: $y_1 \in [-1/3,1]$

(A,b) =

$3x_1 + 4y_1 \leq 5 \quad \rightarrow y_1 = 1$

$4x_1 - 3y_1 \leq 1 \quad \rightarrow y_1 = -1/3$

Damit sieht das neue System wie folgt aus:

$3x_1 + 4 \leq 5$ → $3x_1 \leq 1$

$4x_1 + 1 \leq 1$ → $4x_1 \leq 0$

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheit

Seien

$$L: \exists x_1 \in [a_1, b_1]\ \forall y_1 \in [l_1, u_1]\ \exists x_2 \in [a_2, b_2] \ldots \exists x_n \in [a_n, b_n]\ \forall y_n \in [l_n, u_n] \quad A \cdot \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{pmatrix} \leq b, x \geq 0$$

und

$$R: \exists x_1 \in [a_1, b_1]\ \forall y_1 \in [l_1, u_1]\ \exists x_2 \in [a_2, b_2] \ldots \exists x_n \in [a_n, b_n] \quad A' \cdot \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ y_{n-1} \\ x_n \end{pmatrix} \leq b', x \geq 0$$

zwei Systeme, wobei R aus L durch
Elim-Univ-Variable(A,b,y_n,l_n,u_n) hervorgegangen sei.

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Intuition

Sei $\underline{x}$ eine Gewinnstrategie für X, d.h., X kann in jedem Schritt x_i so wählen, dass egal wie Y spielt, X gewinnt und also insbesondere alle Nebenbedingungen eingehalten werden. Für alle Spiele, die der Strategie $\underline{x}$ folgen, muss also X seinen Zug x_n machen, bevor Y seinen letzten Zug y_n macht.
Wenn X dem Y eine Ungleichung zu zerstören ermöglicht, gewinnt Y.
Also muss X für jede einzelne Ungleichung sicherstellen, dass Y auch im schlimmsten Fall nicht gewinnt.

Korrektheit

L => R ist klar. X gibt Y keine Möglichkeit zu gewinnen.

R => L ist ebenso klar. Wenn es in R eine Gewinnstrategie für X gibt, ist diese durch unsere Konstruktion entstanden. Verfolgt man die Umformungen zurück, bekommt Y zwar mehr Freiheiten (eine zusätzliche) Aktion, die rechte Seite b wird aber so stark erweitert, dass die zusätzlichen Aktionen gerade nicht ermöglichen, eine Ungleichung in L zu zerstören.

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Elim-Exist-Variable(A,b,x_i)

// entspricht Fourier-Motzkin-Elemination

1: Form the set $L_{\leq}$ of every constraint that can be written in the form $x_i \leq 0$. If $x_i \leq m_j$ is a constraint in (A,b), m_j is added to $L_{\leq}$.

2: Form the set $L_{\geq}$ of every constraint that can be written in the form $x_i \geq 0$. If $x_i \geq n_k$ is a constraint in (A,b), n_k is added to $L_{\leq}$.

3: Form the set $L_{=}$ of every constraint that does not contain x_i

4: $\mathcal{J} = \emptyset$

5: for each constraint $m_j \in L_{\leq}$ do

6: for each constraint $n_k \in L_{\geq}$ do

7: Create new constraint l_{kj} from $n_k \leq m_j$; $\mathcal{J} = \mathcal{J} \cup l_{kj}$

8: Form a new system (A‘,b‘) from the constarints in $\mathcal{J} = \mathcal{J} \cup L_{=}$

9: return (A‘,b‘)

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheit

Seien

$$L: \exists x_1\ \forall y_1 \in [l_1, u_1]\ \exists x_2 \ldots \forall y_{n-1} \in [l_{n-1}, u_{n-1}]\ \exists x_n \qquad A \cdot \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{pmatrix} \leq b, x \geq 0$$

und

$$R: \exists x_1\ \forall y_1 \in [l_1, u_1]\ \exists x_2 \ldots \exists x_{n-1}\ \forall\ y_{n-1} \in [l_{n-1}, u_{n-1}] \quad A' \cdot \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ y_{n-1} \\ x_n \end{pmatrix} \leq b', x \geq 0$$

zwei Systeme, wobei R aus L durch **Elim-Univ-Variable(A,b,x_n)** hervorgegangen sei.

Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheit

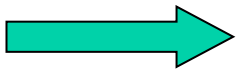
Seien

$\underline{x}_L = [x_1, x_2(x_1,y_1),...,x_n(x_1,y_1,x_2 ...y_{n-1})]$ eine Strategie für L und
$\underline{x}_R = [x_1, x_2(x_1,y_1),..., x_{n-1}(x_1,y_1,x_2 ...y_{n-2})]$ eine Strategie für R.
$\underline{x}_R$ entsteht aus $\underline{x}_L$ durch weglassen der letzten Komponente.

Annahme: $\underline{x}_L$ ist eine Gewinnstrategie für L, aber $\underline{x}_R$ ist keine Gewinnstrategie für R.

Dann gibt es eine Gewinnstrategie $\underline{y}_R(\underline{x}_R)$, $\underline{y}_R$ (möglicherweise) abhängig von $\underline{x}_R$. Betrachte nun das Spiel (x_R,y_R), welches entsteht, indem X und Y die Züge ihrer Strategien $\underline{y}_R(\underline{x}_R)$und $\underline{x}_R$ anwenden.

Da $\underline{y}_R(\underline{x}_R)$ eine Gewinnstrategie ist, gibt es eine Nebenbedingung in $A'\cdot(x_R,y_R) \leq b'$, die verletzt wird. Sei dies in der i-te Zeile und sei $p'_i := A'_{i*} \cdot (x_R,y_R) > b'_i$. Schreibe p'_i auch als $A'^{x}_{i*} \cdot x_R + A'^{y}_{i*} \cdot y_R > b'$, wobei A'^{x}_{i*} und A'^{y}_{i} entsprechende Teil matrizen von A' sind.



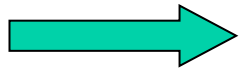
Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheit

1. Fall: p'_i gibt es auch als p_i in L. Dann kommt Variable x_n offenbar in p_i nicht vor. D.h., die gleiche Partie wird von X auch in L verloren.
2. Fall: p'_i wurde in L zusammengesetzt aus $l_i : m_i \leq x_n$ und $l_j : x_n \leq n_j$, um mit *Elim-Exist-Variable* $m_i \leq n_j$ zu erhalten. Wir schreiben
   l_i als $(A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R - b_i) / A_{i,n} \leq x_n$ und
   l_j als $x_n \leq (b_j - A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R) / A_{i,n}$.
   Da $\underline{y}_R(\underline{x}_R)$ eine Gewinnstrategie in R ist, wissen wir aber, dass
   $(A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R - b_i) / A_{i,n} > (b_j - A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R) / A_{i,n}$
   D.h., es kann kein x_n geben, welches zwischen den beiden Termen liegt.
   $\underline{y}_R(\underline{x}_R)$ wäre also auch für L eine Gewinnstrategie für Spieler Y.

Es gilt also L => R. Wir müssen noch die Rückrichtung zeigen.



Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheit

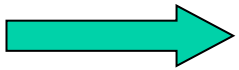
Seien $S_1 = \{m_1 \leq x_n, m_2 \leq x_n,...,m_p \leq x_n\}$ und $S_2 = \{x_n \leq n_1, x_n \leq n_2,...,x_n \leq n_q\}$ die Nebenbedingungen in L, die als $x_n \geq ()$ bzw. als $x_n \leq ()$ geschrieben werden können. $\underline{x}_R$ sei eine Gewinnstrategie für Spieler X in R.

Betrachte nun x_n mit: $\max^p_{i=1} m_i \leq x_n \leq \min^q_{j=1} n_j$. Die Behauptung ist, dass $\underline{x}_L = [\underline{x}_R, x_n]$ eine Gewinnstrategie für L ist.

Annahme: Das Gegenteil ist der Fall, $\underline{x}_L$ ist also keine Gewinnstrategie für L.

Dann gibt es eine Gewinnstrategie $\underline{y}_L(\underline{x}_L)$ für Y. Seien y_L und x_L wieder die Züge des entstehenden Spiels. $\underline{x}_R$, $\underline{y}_R$, x_R und y_R analog.

Da $\underline{y}_L(\underline{x}_L)$ eine Gewinnstrategie von Y in L ist, gibt es eine Nebenbedingung p_i, die im System $A \cdot (x_L, y_L) \leq b$ verletzt wird.



Eliminationsalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheit

1. Fall: p_i taucht in gleicher Form in R auf. Offenbar war x_n dann nicht Teil von p_i. Somit gewinnt Y mit $\underline{y}_R(x_R)$ auch in R.
2. p_i sei von der Form $l_i : m_i \leq x_n \in S_1$. Betrachte nun eine beliebiges Element $l_j : x_n \leq n_j \in S_2$. Dann ist $m_i \leq n_j$ Teil von R.

Wir schreiben l_i als $(A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R - b_i) / A_{i,n} \leq x_n$ und l_j als $x_n \leq (b_j - A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R) / A_{i,n}$. Weil $\underline{x}_R$ eine Strategie für R ist, wissen wir, dass

$$(A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R - b_i) / A_{i,n} \leq (b_j - A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R) / A_{i,n}.$$

Da aber $y_L(x_L)$ eine Gewinnstrategie für Y in L ist, gilt auch:

$(A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R - b_i) / A_{i,n} > x_n$, für alle $x_n \in \mathbb{Q}$.

D.h., X konnte kein $x_n \in \mathbb{Q}$ finden, so dass

$(A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R - b_i) / A_{i,n} \leq x_n$ und

$x_n \leq (b_j - A^x_{i,\{1,...,2n-2\}} \cdot x_R - A^y_{i,\{1,...,2n-1\}} \cdot y_R) / A_{i,n}$.

$\underline{y}_L(\underline{x}_L)$ kann also keine Gewinnstrategie für L sein, da anderenfalls $\underline{y}_R(\underline{x}_R)$ eine Gewinnstrategie von Y in R wäre. (analog für p_i von der Form $l_j : x_n \leq n_j \in S_2$.)