

Seminar Decision Procedures

Linear Arithmetic

Matthias Kohler

Definition (linear arithmetic)

The syntax of a formula in linear arithmetic is defined by the following rules

formula : *formula* \wedge *formula* | (*formula*) | *atom*

atom : *sum* *op* *sum*

op : = | \leq

sum : *term* | *sum* + *term*

term : *identifier* | *constant* | *constant identifier*

- *constant* is a rational number
- *identifier* is the name of a variable
- *atoms* are also called *constraints*

Example: $3x_1 + 2x_2 \leq 5x_3 \wedge 2x_1 - 2x_2 = 0$

(Decision Problem)

Given a formula in linear arithmetic

- *check if there is a satisfying assignment to the variables over the reals*
 - ▶ *Simplex Algorithm (worst case exponential time)*
 - ▶ *Fourier Motzkin Elimination*
 - ▶ *Ellipsoid Method (polynomial time)*
 - ▶ *Interior Point Methods*
- *check if there is a satisfying assignment to the variables over the integers*
 - ▶ *NP-hard*
 - ▶ *Brute Force*
 - ▶ *Branch and Bound*
 - ▶ *Cutting-Planes*
 - ▶ *Omega Test*

Dealing with equality constraints

Fourier Motzkin and the Omega Test work only with inequality constraints. There are different methods for eliminating equality constraints.

- For a constraint of the form $a = b$ introduce the two constraints

$$a \leq b$$

$$b \leq a$$

- Solve a equality constraint for some variable x so it is of the form

$$x = A$$

Plug in A for every occurrence of x in every other constraint (equality / inequality).

Fourier-Motzkin Elimination

Eliminate variables one by one until there are only constraints left, that are either trivially satisfiable or trivially unsatisfiable.

But first eliminate equality constraints:

- Solve every equality constraint for some variable x_j .
- Plug the equality into all occurrences of x_j in the other inequalities and equalities.
- Drop the equality constraint.
- Repeat this process, until we have only inequality constraints or a trivially unsatisfiable or trivially satisfiable system.

Fourier-Motzkin Elimination

- Rearrange constraints that contain x_i such that all constraint are in one of the two forms

$$\begin{aligned}x_i &\leq R_i \\L_i &\leq x_i ,\end{aligned}$$

where L_i and R_i do not contain x_i .

- Drop all constraints that contain x_i .
- Introduce the constraints

$$L_i \leq R_j \quad \forall L_i \text{ and } \forall R_j .$$

- Variable x_i is eliminated.
- The number of constraints increased from m to at most $m^2/4$

Fourier-Motzkin Elimination

Theorem

A system of inequalities is satisfiable before eliminating x_i iff it is satisfiable after eliminating x_i .

Proof.

" \implies "

- Consider a satisfiable system already rearranged for x_i .
- Satisfiability implies, that there is an assignment such that

$$\begin{aligned}x_i &\leq R_i \\L_i &\leq x_i\end{aligned}$$

for all L_i and R_i and all other constraints that do not contain x_i are also satisfied.

- This implies

$$L_i \leq x_i \leq R_i$$

Fourier-Motzkin Elimination

Proof.

- This implies

$$L_i \leq R_i$$

for all L_i and R_i .

- Hence the system where x_i is eliminated is satisfiable.

" \Leftarrow "

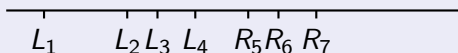
- Consider a system that after eliminating x_i is satisfiable.
- Satisfiability implies that there is an assignment such that

$$L_i \leq R_i$$

for all L_i and R_i and all other constraints that did not contain x_i are also satisfied.

Proof.

- After plugging in the assignment into all L_i and R_i we have the following situation.



- Choose x_i as some number in $[\max_i L_i, \min_i R_i]$.
- Then we have a satisfying assignment for the system

$$x_i \leq R_i$$

$$L_i \leq x_i$$

- Hence the original system is satisfiable.



Fourier-Motzkin Elimination Example

$$\begin{array}{rcl} x_1 - x_2 & \leq 0 & x_1 \leq x_2 \\ x_1 & -x_3 \leq 0 & x_1 \leq x_3 \\ -x_1 + x_2 + 2x_3 & \leq 0 & x_2 + 2x_3 \leq x_1 \\ & -x_3 \leq -1 & -x_3 \leq -1 \end{array} \quad \text{rearrange}$$

drop old constraints, introduce new constraints, eliminate x_1

$$\begin{array}{rcl} x_2 + 2x_3 & \leq x_2 & 2x_3 \leq 0 \\ x_2 + 2x_3 & \leq x_3 & x_2 + x_3 \leq 0 \\ & -x_3 \leq -1 & -x_3 \leq -1 \end{array} \quad \text{simplify}$$

Fourier-Motzkin Elimination Example

Remark: If all constraints that contain x_i after rearranging are of the form $x_i \leq R_i$ (or $L_i \leq x_i$), no new constraints are introduced.

$$\begin{array}{ccc} 2x_3 \leq 0 & & 2x_3 \leq 0 \\ x_2 + x_3 \leq 0 & \text{rearrange} & x_2 \leq -x_3 \\ -x_3 \leq -1 & & -x_3 \leq -1 \end{array}$$

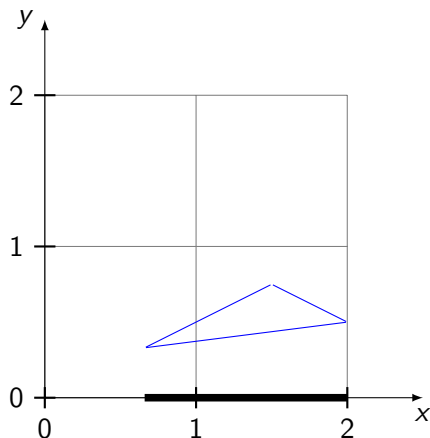
drop old constraints, introduce new constraints, eliminate x_2

$$\begin{array}{l} 2x_3 \leq 0 \\ -x_3 \leq -1 \end{array}$$

This is trivially satisfiable.

Fourier Motzkin Geometric Interpretation

Consider the following picture. We want to eliminate y .



Omega Test

Linear Arithmetic over the integers.

Idea / Overview:

- Eliminate a variable, similar to Fourier Motzkin
- Check if the new system (real shadow) is satisfiable over the integers.
 - ▶ If not, then stop, system is unsatisfiable.
 - ▶ Else proceed
- Check an overapproximation of the system (dark shadow) for an integer solution recursively with the Omega test.
 - ▶ If a solution is found then stop, the system is satisfiable.
 - ▶ Else proceed, an integer solution might still exist.
- Check for an integer solution in the system with something slightly better than brute force (grey shadow).

Require: A conjunction of linear constraints C

```
1: function OMEGA TEST( $C$ )
2:   if  $C$  only contains one variable then
3:     Solve and return result
4:   Choose a variable  $v$  to eliminate
5:    $C_R = \text{RealShadow}(C, v)$ 
6:   if OmegaTest( $C_R$ ) = Unsatisfiable then
7:     return Unsatisfiable
8:    $C_D = \text{DarkShadow}(C, v)$ 
9:   if OmegaTest( $C_D$ ) = Satisfiable then
10:    return Satisfiable
11:    $C_1^G, \dots, C_n^G = \text{GreyShadow}(C, v)$ 
12:   for all  $i \in \{1, \dots, n\}$  do
13:     if OmegaTest( $C_G^i$ ) = Satisfiable then
14:       return Satisfiable
15:   return Unsatisfiable
```

Real Shadow

Let z be the variable that should be eliminated.

Rearrange all constraints that contain z so that all constraints are of the form

$$\beta \leq bz \quad \text{or} \quad cz \leq \gamma .$$

Put every left hand side β and every right hand side γ together to the following constraints.

$$c\beta \leq cbz \leq b\gamma .$$

Drop all constraints that contain z and instead introduce for every left hand side β and every right hand side γ the following constraints.

$$c\beta \leq b\gamma .$$

Keep all constraints that do not contain z . If the original system is satisfiable, so is the new system.

Real Shadow

If the real shadow is satisfiable the original system might not be satisfiable.
Consider the following constraints:

$$\begin{aligned}y &\leq x/2 \\(2 + x)/8 &\leq y \\y &\leq (3 - x)/2\end{aligned}$$

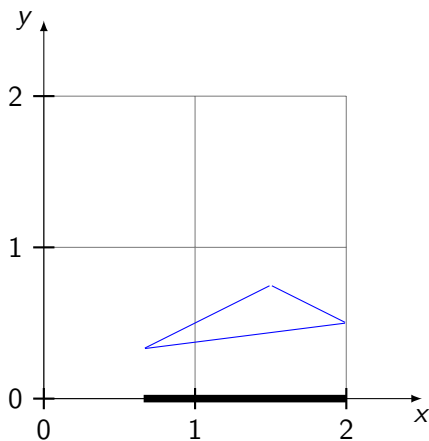
Eliminate y :

$$\begin{aligned}(2 + x)/8 &\leq x/2 \\(2 + x)/8 &\leq (3 - x)/2\end{aligned}$$

Simplify

$$\begin{aligned}2/3 &\leq x \\x &\leq 10\end{aligned}$$

As can be seen in the following plot, the original system (blue triangle) is unsatisfiable over the integers, but the new system (black line) is satisfiable over the integers.



Dark Shadow

Let z be the variable that should be eliminated.

We know that the real shadow is satisfiable, there exists an assignment to variables (except z) so that $c\beta \leq b\gamma$. We want to find out if there also exists a z such that

$$c\beta \leq cbz \leq b\gamma .$$

Equivalent to

$$\exists z \in \mathbb{Z} : \frac{\beta}{b} \leq z \leq \frac{\gamma}{c}$$

Informally we want to find out for every pair of lower and upper bounds (β, γ) if there fits a integer in between that is divisible by bc .

If such an integer can be found for every pair, we have our z and the system is satisfiable.

Idea: Prove the existence of z by contradiction.

Consider one particular left hand side right hand side pair and assume no integer fits between $\frac{\beta}{b}$ and $\frac{\gamma}{c}$.

- because of non strict inequalities $\frac{\beta}{b}$ and $\frac{\gamma}{c}$ can not be integer.
- $\frac{\beta}{b}$ and $\frac{\gamma}{c}$ must be strictly between two consecutive integers.

Thus

$$\left\lfloor \frac{\beta}{b} \right\rfloor < \frac{\beta}{b} \leq \frac{\gamma}{c} < \left\lfloor \frac{\beta}{b} \right\rfloor + 1$$

From this inequality

$$\begin{aligned} \frac{\beta}{b} - \left\lfloor \frac{\beta}{b} \right\rfloor &\geq \frac{1}{b} \\ \left\lfloor \frac{\beta}{b} \right\rfloor + 1 - \frac{\gamma}{c} &\geq \frac{1}{c} \end{aligned}$$

can be derived. (Proof is Homework)

Summing the last two inequalities gives

$$\frac{\beta}{b} + 1 - \frac{\gamma}{c} \geq \frac{1}{b} + \frac{1}{c} .$$

Rearrange

$$\frac{\beta}{b} - \frac{\gamma}{c} \geq -1 + \frac{1}{b} + \frac{1}{c} .$$

Multiply both sides with $-cb$ gives

$$b\gamma - c\beta \leq cb - c - b .$$

Since we want to prove the existence of z by contradiction, we have to prove the negation of the last inequality, that is

$$b\gamma - c\beta > cb - c - b .$$

Because all terms are integers this is equivalent to

$$b\gamma - c\beta \geq cb - c - b + 1 = (c - 1)(b - 1).$$

All inequalities of this form, for every β and γ and the inequalities that did not contain z form the dark shadow.

- Satisfiability of the dark shadow implies satisfiability of the original system.
- Unsatisfiability of the dark shadow does not imply unsatisfiability of the original system.

Dark Shadow Example

Consider the following constraints.

$$\begin{aligned}2y &\leq z \\8y - 2 &\leq z \\z &\leq -2y + 3\end{aligned}$$

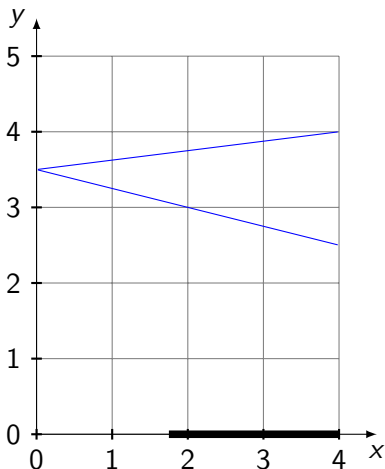
The dark shadow is:

$$\begin{aligned}-2y + 3 - 2y &\geq (1 - 1)(1 - 1) \\-2y + 3 - (8y - 2) &\geq (1 - 1)(1 - 1)\end{aligned}$$

The real shadow is:

$$\begin{aligned}2y &\leq -2y + 3 \\8y - 2 &\leq -2y + 3\end{aligned}$$

Dark Shadow Geometric Interpretation



$$y \leq \frac{1}{8}x + 3.5$$

$$y \geq -\frac{1}{4}x + 0.5$$

The dark shadow when eliminating y
is

$$x \geq 1.75$$

Grey Shadow

If there is an integer solution to the original Problem C , but there is no integer solution in the Dark Shadow, this solution has to satisfy

$$\beta \leq bz \quad (\text{For all } \beta)$$

$$cz \leq \gamma \quad (\text{For all } \gamma)$$

The constraints from C that do not contain z

$$b\gamma - c\beta \leq cb - c - b \quad (\text{For one } \beta \text{ and one } \gamma)$$

The last constraint separates the integer solution from the dark shadow.

Grey Shadow

For one β and one γ we have in addition to all other constraints

$$c\beta \leq cbz \leq b\gamma \text{ and } b\gamma - c\beta \leq cb - c - b .$$

Rearrange

$$c\beta \leq cbz \leq b\gamma \text{ and } b\gamma \leq cb - c - b + c\beta .$$

Equivalent to

$$c\beta \leq cbz \leq b\gamma \leq cb - c - b + c\beta .$$

This implies

$$c\beta \leq cbz \leq cb - c - b + c\beta .$$

Grey Shadow

An integer solution has to satisfy

$$c\beta \leq cbz \leq cb - c - b + c\beta ,$$

in addition to all other constraints. Divide by c .

$$\beta \leq bz \leq (cb - c - b)/c + \beta .$$

- For every integer i in $\{0, \dots, \lfloor (cb - c - b)/c \rfloor\}$ we add the constraint $bz = \beta + i$.
- For every i one grey shadow is formed.

Grey Shadow

We do not know which constraint (which β and γ) separates the solution from the dark shadow.

- Form one grey shadow for each β , each γ and each i .

Adding the constraint $bz = \beta + i$ does not eliminate z .

- For each occurrence of z in the constraints, plug in $(\beta + i)/b$ and multiply the equation by some number so that all coefficients are integer again.

Grey Shadow Example

Consider the following constraints.

$$\begin{aligned}2y &\leq 2z \\8y - 2 &\leq 3z \\5z &\leq -2y + 3\end{aligned}$$

The dark shadows for the constraint pair $2y \leq 2z$, $5z \leq -2y + 3$ are:

$$\begin{aligned}2y &\leq 2z \\8y - 2 &\leq 3z \\5z &\leq -2y + 3 \\10z &= i + 2y\end{aligned}$$

for all integers i in $\{0, \dots, \lfloor (10 - 2 - 5)/5 \rfloor\} = \{0\}$.

Grey Shadow Example

z is not eliminated yet. Let $i = 0$, for every occurrence of z plug in $2y/10$.

$$\begin{aligned}2y &\leq 2\frac{2y}{10} \\8y - 2 &\leq 3\frac{2y}{10} \\5\frac{2y}{10} &\leq -2y + 3\end{aligned}$$

Multiply the inequalities with 10 to get integer coefficients.

$$\begin{aligned}20y &\leq 4y \\80y - 20 &\leq 6y \\10z &\leq -20y + 30\end{aligned}$$

Applications of Linear Arithmetic

Optimization and satisfiability checking can be reduced to each other.

Linear optimization problem

$$\begin{aligned} \max \quad & c^T x \\ & Ax \leq b \\ & x \geq 0 \end{aligned}$$

Satisfiability problem

$$\begin{aligned} & Ax \leq b \\ & x \geq 0 \end{aligned}$$

Satisfiability to Optimization: Set $c = 0$.

Optimization to Satisfiability: Binary Search.

- Find lower and upper bound for the optimal solution.
- Do binary search for the optimal solution between the bounds, by adding taking the constraints of the optimization problem and adding the constraint $c^T x \geq z$.

Linear Arithmetic Hardness

SAT can be coded as a Linear Arithmetic formula.

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_1 \vee x_4)$$

In Linear Arithmetic:

$$(x_1 + x_2 + (1 - x_3) \geq 1) \wedge (x_2 + (1 - x_1) + x_4 \geq 1) \wedge x_1, x_2, x_3, x_4 \in \{0, 1\}$$

 Stephen Boyd and Lieven Vandenberghe.

Convex Optimization.

Cambridge University Press, New York, NY, USA, 2004.

 Daniel Kroening and Ofer Strichman.

Decision Procedures: An Algorithmic Point of View, chapter Linear Arithmetic, pages 111–147.

Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

 William Pugh.

The omega test: a fast and practical integer programming algorithm for dependence analysis.

Communications of the ACM, 8:4–13, 1992.