Satisfiability Checking

Gauß and Fourier-Motzkin Variable Elimination for Linear Real Arithmetic

Prof. Dr. Erika Ábrahám

RWTH Aachen University
Informatik 2
LuFG Theory of Hybrid Systems

WS 19/20

The Xmas problem

There are three types of Xmas presents Santa Claus can make.

- Santa Claus wants to reduce the overhead by making only two types.
- He needs at least 100 presents.
- He needs at least 5 of either type 1 or type 2.
- He needs at least 10 of the third type.
- Each present of type 1, 2, and 3 need 1, 2, resp. 5 minutes to make.
- Santa Claus is late, and he has only 3 hours left.
- Each present of type 1, 2, and 3 costs 3, 2, resp. 1 EUR.
- He has 300 EUR for presents in total.

The Xmas problem

There are three types of Xmas presents Santa Claus can make.

- Santa Claus wants to reduce the overhead by making only two types.
- He needs at least 100 presents.
- He needs at least 5 of either type 1 or type 2.
- He needs at least 10 of the third type.
- Each present of type 1, 2, and 3 need 1, 2, resp. 5 minutes to make.
- Santa Claus is late, and he has only 3 hours left.
- Each present of type 1, 2, and 3 costs 3, 2, resp. 1 EUR.
- He has 300 EUR for presents in total.

$$(p_1 = 0 \vee p_2 = 0 \vee p_3 = 0) \wedge p_1 + p_2 + p_3 \geq 100 \wedge$$
$$(p_1 \geq 5 \vee p_2 \geq 5) \wedge p_3 \geq 10 \wedge p_1 + 2p_2 + 5p_3 \leq 180 \wedge$$
$$3p_1 + 2p_2 + p_3 \leq 300$$

Quantifier-free linear real arithmetic (QFLRA)

Quantifier-free linear real arithmetic (QFLRA)

Linear real arithmetic is the first-order theory with signature $\{0, 1, +, <\}$ and the domain being the reals $\mathbb{R}$.

Quantifier-free linear real arithmetic (QFLRA)

Linear real arithmetic is the first-order theory with signature $\{0, 1, +, <\}$ and the domain being the reals $\mathbb{R}$.

Syntax of linear real arithmetic

Terms:	t	$::=$	0	$\mid$	1	$\mid$	x	$\mid$ $t + t$
Constraints:	c	$::=$	$t < t$					
Formulas:	φ	$::=$	c	$\mid$	$\neg\varphi$	$\mid$	$\varphi \wedge \varphi$	$\mid$ $\exists x.\ \varphi$

where x stays for a variable.

Quantifier-free linear real arithmetic (QFLRA)

Linear real arithmetic is the first-order theory with signature $\{0, 1, +, <\}$ and the domain being the reals $\mathbb{R}$.

Syntax of linear real arithmetic

Terms:	t	$::=$	0	$\mid$	1	$\mid$	x	$\mid$ $t + t$
Constraints:	c	$::=$	$t < t$					
Formulas:	φ	$::=$	c	$\mid$	$\neg\varphi$	$\mid$	$\varphi \wedge \varphi$	$\mid$ $\exists x.\ \varphi$

where x stays for a variable.

- Syntactic sugar for constraints: $t_1 \leq t_2$, $t_1 = t_2$, $t_1 \neq t_2$.

Quantifier-free linear real arithmetic (QFLRA)

Linear real arithmetic is the first-order theory with signature $\{0, 1, +, <\}$ and the domain being the reals $\mathbb{R}$.

Syntax of linear real arithmetic

Terms:	t	$::=$	0	$\mid$	1	$\mid$	x	$\mid$ $t + t$
Constraints:	c	$::=$	$t < t$					
Formulas:	φ	$::=$	c	$\mid$	$\neg\varphi$	$\mid$	$\varphi \wedge \varphi$	$\mid$ $\exists x.\ \varphi$

where x stays for a variable.

- Syntactic sugar for constraints: $t_1 \leq t_2$, $t_1 = t_2$, $t_1 \neq t_2$.
- The semantics is standard.

Quantifier-free linear real arithmetic (QFLRA)

Linear real arithmetic is the first-order theory with signature $\{0, 1, +, <\}$ and the domain being the reals $\mathbb{R}$.

Syntax of linear real arithmetic

Terms:	t	$::=$	0	$\mid$	1	$\mid$	x	$\mid$ $t + t$
Constraints:	c	$::=$	$t < t$					
Formulas:	φ	$::=$	c	$\mid$	$\neg\varphi$	$\mid$	$\varphi \wedge \varphi$	$\mid$ $\exists x.\ \varphi$

where x stays for a variable.

- Syntactic sugar for constraints: $t_1 \leq t_2$, $t_1 = t_2$, $t_1 \neq t_2$.
- The semantics is standard.
- Linear real arithmetic is also called linear real algebra.

Quantifier-free linear real arithmetic (QFLRA)

Linear real arithmetic is the first-order theory with signature $\{0, 1, +, <\}$ and the domain being the reals $\mathbb{R}$.

Syntax of linear real arithmetic

Terms:	t	$::=$	0	$\mid$	1	$\mid$	x	$\mid$ $t + t$
Constraints:	c	$::=$	$t < t$					
Formulas:	φ	$::=$	c	$\mid$	$\neg\varphi$	$\mid$	$\varphi \wedge \varphi$	$\mid$ $\exists x.\ \varphi$

where x stays for a variable.

- Syntactic sugar for constraints: $t_1 \leq t_2$, $t_1 = t_2$, $t_1 \neq t_2$.
- The semantics is standard.
- Linear real arithmetic is also called linear real algebra.
- We consider the satisfiability problem for the quantifier-free fragment QFLRA (or equivalently the existential fragment, i.e., no universal quantifiers and no negation of expressions containing existential quantifiers).

Gauß variable elimination (based on equations)

Reminder: In an SMT solver for QFLRA, the theory solver needs to check the satisfiability of sets of constraints $\sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i$, where $a_{i,k}$ and b_i are integer (or rational) constants, x_k are real-valued variables, and $\sim_i \in \{=, \leq, <\}$ for $k=1, \ldots, N$ and $i=1, \ldots, M$. (Note: $t > b$ is equivalent to $-t < -b$ and similarly for $\geq$.)

Gauß variable elimination (based on equations)

Reminder: In an SMT solver for QFLRA, the theory solver needs to check the satisfiability of sets of constraints $\sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i$, where $a_{i,k}$ and b_i are integer (or rational) constants, x_k are real-valued variables, and $\sim_i \in \{=, \leq, <\}$ for $k=1, \ldots, N$ and $i=1, \ldots, M$. (Note: $t > b$ is equivalent to $-t < -b$ and similarly for $\geq$.)

- Assume that the ith constraint is an equation with $a_{ij} \neq 0$ for some $1 \leq j \leq N$:

$$\sum_{k=1}^{N} a_{ik} \cdot x_k = b_i \quad (a_{i,k}, b_i\text{: integer/rational constants, } x_k\text{: variables})$$

Gauß variable elimination (based on equations)

Reminder: In an SMT solver for QFLRA, the theory solver needs to check the satisfiability of sets of constraints $\sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i$, where $a_{i,k}$ and b_i are integer (or rational) constants, x_k are real-valued variables, and $\sim_i \in \{=, \leq, <\}$ for $k=1, \ldots, N$ and $i=1, \ldots, M$. (Note: $t > b$ is equivalent to $-t < -b$ and similarly for $\geq$.)

- Assume that the ith constraint is an equation with $a_{ij} \neq 0$ for some $1 \leq j \leq N$:

$$\sum_{k=1}^{N} a_{ik} \cdot x_k = b_i \quad (a_{i,k}, b_i\text{: integer/rational constants, } x_k\text{: variables})$$

$$\Rightarrow \quad a_{ij} \cdot x_j \;=\; b_i - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} a_{ik} \cdot x_k$$

Gauß variable elimination (based on equations)

Reminder: In an SMT solver for QFLRA, the theory solver needs to check the satisfiability of sets of constraints $\sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i$, where $a_{i,k}$ and b_i are integer (or rational) constants, x_k are real-valued variables, and $\sim_i \in \{=, \leq, <\}$ for $k=1, \ldots, N$ and $i=1, \ldots, M$. (Note: $t > b$ is equivalent to $-t < -b$ and similarly for $\geq$.)

- Assume that the ith constraint is an equation with $a_{ij} \neq 0$ for some $1 \leq j \leq N$:

$$\sum_{k=1}^{N} a_{ik} \cdot x_k = b_i \quad (a_{i,k}, b_i: \textit{ integer/rational constants, } x_k: \textit{ variables})$$

$$\Rightarrow \quad a_{ij} \cdot x_j = b_i - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} a_{ik} \cdot x_k$$

$$\Rightarrow \quad x_j = \frac{b_i}{a_{ij}} - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} \frac{a_{ik}}{a_{ij}} \cdot x_k := \beta_j$$

Gauß variable elimination (based on equations)

Reminder: In an SMT solver for QFLRA, the theory solver needs to check the satisfiability of sets of constraints $\sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i$, where $a_{i,k}$ and b_i are integer (or rational) constants, x_k are real-valued variables, and $\sim_i \in \{=, \leq, <\}$ for $k=1, \ldots, N$ and $i=1, \ldots, M$. (Note: $t > b$ is equivalent to $-t < -b$ and similarly for $\geq$.)

- Assume that the ith constraint is an equation with $a_{ij} \neq 0$ for some $1 \leq j \leq N$:

$$\sum_{k=1}^{N} a_{ik} \cdot x_k = b_i \quad (a_{i,k}, b_i\text{: integer/rational constants, } x_k\text{: variables})$$

$$\Rightarrow \quad a_{ij} \cdot x_j = b_i - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} a_{ik} \cdot x_k$$

$$\Rightarrow \quad x_j = \frac{b_i}{a_{ij}} - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} \frac{a_{ik}}{a_{ij}} \cdot x_k := \beta_j$$

- Replace x_j by β_j in all constraints (and multiply the involved constraints by a_{ij} if integer coefficients are wanted).

Gauß variable elimination (based on equations)

Reminder: In an SMT solver for QFLRA, the theory solver needs to check the satisfiability of sets of constraints $\sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i$, where $a_{i,k}$ and b_i are integer (or rational) constants, x_k are real-valued variables, and $\sim_i \in \{=, \leq, <\}$ for $k=1, \ldots, N$ and $i=1, \ldots, M$. (Note: $t > b$ is equivalent to $-t < -b$ and similarly for $\geq$.)

- Assume that the ith constraint is an equation with $a_{ij} \neq 0$ for some $1 \leq j \leq N$:

$$\sum_{k=1}^{N} a_{ik} \cdot x_k = b_i \quad (a_{i,k}, b_i: \textit{ integer/rational constants, } x_k: \textit{ variables})$$

$$\Rightarrow \quad a_{ij} \cdot x_j = b_i - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} a_{ik} \cdot x_k$$

$$\Rightarrow \quad x_j = \frac{b_i}{a_{ij}} - \sum_{k \in \{1,\ldots,j-1,j+1,\ldots,N\}} \frac{a_{ik}}{a_{ij}} \cdot x_k := \beta_j$$

- Replace x_j by β_j in all constraints (and multiply the involved constraints by a_{ij} if integer coefficients are wanted).
- After removing tautologies, this substitutiton leads to an equisatisfiable problem with (at most) $M-1$ constraints in (at most) $N-1$ variables (at least the ith constraint and x_j are eliminated).

What remains to be solved

- Let us assume first that after applying Gauß variable elimination as long as possible, m non-strict inequalities in n variables are left (i.e., there are no strict inequalities):

$$\bigwedge_{1 \leq i \leq m} \sum_{1 \leq j \leq n} a_{ij} x_j \leq b_i$$

What remains to be solved

- Let us assume first that after applying Gauß variable elimination as long as possible, m non-strict inequalities in n variables are left (i.e., there are no strict inequalities):

$$\bigwedge_{1 \leq i \leq m} \sum_{1 \leq j \leq n} a_{ij} x_j \leq b_i$$

- Input in matrix form: $A\overline{x} \leq \overline{b}$

m constraints

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ a_{m1} & a_{22} & \cdots & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{pmatrix}$$

n variables

Fourier-Motzkin variable elimination

- Earliest method for solving linear inequality systems:
  discovered in 1826 by Fourier, re-discovered by Motzkin in 1936

Fourier-Motzkin variable elimination

- Earliest method for solving linear inequality systems:

  discovered in 1826 by Fourier, re-discovered by Motzkin in 1936

- Basic idea of variable elimination:
  - Pick a variable and eliminate it, yielding an equisatisfiable formula that does not refer to the eliminated variable any more.
  - Continue until all variables are eliminated.

Fourier-Motzkin variable elimination

- Earliest method for solving linear inequality systems:

  discovered in 1826 by Fourier, re-discovered by Motzkin in 1936

- Basic idea of variable elimination:
  - Pick a variable and eliminate it, yielding an equisatisfiable formula that does not refer to the eliminated variable any more.
  - Continue until all variables are eliminated.

- Fourier-Motzkin: Collect requirements on the lower an upper bounds on the variable we want to eliminate.

Variable bounds

- For a variable x_n, we can partition the constraints according to the coefficients of x_n:
  - $a_{in} = 0$: constraint i puts no bound on x_n
  - $a_{in} > 0$: constraint i puts an upper bound on x_n
  - $a_{in} < 0$: constraint i puts a lower bound on x_n

Variable bounds

- For a variable x_n, we can partition the constraints according to the coefficients of x_n:
  - $a_{in} = 0$: constraint i puts no bound on x_n
  - $a_{in} > 0$: constraint i puts an upper bound on x_n
  - $a_{in} < 0$: constraint i puts a lower bound on x_n

$$\sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i$$

Variable bounds

- For a variable x_n, we can partition the constraints according to the coefficients of x_n:
  - $a_{in} = 0$: constraint i puts no bound on x_n
  - $a_{in} > 0$: constraint i puts an upper bound on x_n
  - $a_{in} < 0$: constraint i puts a lower bound on x_n

$$\sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i$$

$$\Rightarrow \quad a_{in} \cdot x_n \;\leq\; b_i - \sum_{j=1}^{n-1} a_{ij} \cdot x_j$$

Variable bounds

- For a variable x_n, we can partition the constraints according to the coefficients of x_n:
  - $a_{in} = 0$: constraint i puts no bound on x_n
  - $a_{in} > 0$: constraint i puts an upper bound on x_n
  - $a_{in} < 0$: constraint i puts a lower bound on x_n

$$\sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i$$

$$\Rightarrow \quad a_{in} \cdot x_n \leq b_i - \sum_{j=1}^{n-1} a_{ij} \cdot x_j$$

$$(a) \quad \overset{a_{in}>0}{\Rightarrow} \quad x_n \leq \frac{b_i}{a_{in}} - \sum_{j=1}^{n-1} \frac{a_{ij}}{a_{in}} \cdot x_j \qquad \text{upper bound}$$

$$(b) \quad \overset{a_{in}<0}{\Rightarrow} \quad x_n \geq \frac{b_i}{a_{in}} - \sum_{j=1}^{n-1} \frac{a_{ij}}{a_{in}} \cdot x_j \qquad \text{lower bound}$$

Example for upper and lower bounds

Category for x_1?

$$
\begin{aligned}
&(1) \quad x_1 - x_2 \leq 0 \\
&(2) \quad x_1 - x_3 \leq 0 \\
&(3) \quad -x_1 + x_2 + 2x_3 \leq 0 \\
&(4) \quad -x_3 \leq -1
\end{aligned}
$$

Example for upper and lower bounds

$$
\begin{aligned}
&(1) \quad x_1 - x_2 \leq 0 \\
&(2) \quad x_1 - x_3 \leq 0 \\
&(3) \quad -x_1 + x_2 + 2x_3 \leq 0 \\
&(4) \quad -x_3 \leq -1
\end{aligned}
$$

Category for x_1?

Upper bound

Example for upper and lower bounds

		Category for x_1?
(1)	$x_1 - x_2 \leq 0$	Upper bound
(2)	$x_1 - x_3 \leq 0$	Upper bound
(3)	$-x_1 + x_2 + 2x_3 \leq 0$	
(4)	$-x_3 \leq -1$	

Example for upper and lower bounds

		Category for x_1?
(1)	$x_1 - x_2 \leq 0$	Upper bound
(2)	$x_1 - x_3 \leq 0$	Upper bound
(3)	$-x_1 + x_2 + 2x_3 \leq 0$	Lower bound
(4)	$-x_3 \leq -1$	

Example for upper and lower bounds

		Category for x_1?
(1)	$x_1 - x_2 \leq 0$	Upper bound
(2)	$x_1 - x_3 \leq 0$	Upper bound
(3)	$-x_1 + x_2 + 2x_3 \leq 0$	Lower bound
(4)	$-x_3 \leq -1$	No bound

Eliminating unbounded variables

- Iteratively remove variables that are not bounded in both ways (and all the constraints that use them).
- The new problem has a solution iff the old problem has one!

$$\begin{aligned} -8x + 7y &\leq 0 \\ -x &\leq -3 \\ -y + z &\leq 0 \\ -z &\leq -10 \\ z &\leq 20 \end{aligned}$$

Eliminating unbounded variables

- Iteratively remove variables that are not bounded in both ways (and all the constraints that use them).
- The new problem has a solution iff the old problem has one!

$$
\begin{array}{rcl}
\cancel{-8x + 7y} & \cancel{\leq} & \cancel{0} \\
\cancel{x} & \cancel{\leq} & \cancel{3} \\
-y + z & \leq & 0 \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
$$

Eliminating unbounded variables

- Iteratively remove variables that are not bounded in both ways (and all the constraints that use them).
- The new problem has a solution iff the old problem has one!

$$
\begin{array}{rcl}
\cancel{-8x + 7y} & \cancel{\leq} & \cancel{0} \\
\cancel{x} & \cancel{\leq} & \cancel{3} \\
-y + z & \leq & 0 \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
\quad \longrightarrow \quad
\begin{array}{rcl}
-y + z & \leq & 0 \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
$$

Eliminating unbounded variables

- Iteratively remove variables that are not bounded in both ways (and all the constraints that use them).
- The new problem has a solution iff the old problem has one!

$$
\begin{array}{rcl}
\cancel{-8x + 7y} & \cancel{\leq} & \cancel{0} \\
\cancel{x} & \cancel{\leq} & \cancel{3} \\
-y + z & \leq & 0 \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
\quad \longrightarrow \quad
\begin{array}{rcl}
\cancel{y + z} & \cancel{\leq} & \cancel{0} \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
$$

Eliminating unbounded variables

- Iteratively remove variables that are not bounded in both ways (and all the constraints that use them).
- The new problem has a solution iff the old problem has one!

$$
\begin{array}{rcl}
\cancel{-8x + 7y} & \cancel{\leq} & \cancel{0} \\
\cancel{x} & \cancel{\leq} & \cancel{3} \\
-y + z & \leq & 0 \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
\longrightarrow
\begin{array}{rcl}
\cancel{y + z} & \cancel{\leq} & \cancel{0} \\
-z & \leq & -10 \\
z & \leq & 20
\end{array}
\longrightarrow
\begin{array}{rcl}
-z & \leq & -10 \\
z & \leq & 20
\end{array}
$$

Fourier-Motzkin variable elimination

- For each pair of a lower bound β_l and an upper bound β_u, we have

$$\beta_l \quad \leq \quad x_n \quad \leq \quad \beta_u$$

Fourier-Motzkin variable elimination

- For each pair of a lower bound β_l and an upper bound β_u, we have

$$\beta_l \quad \leq \quad x_n \quad \leq \quad \beta_u$$

- For each such pair, add the constraint

$$\beta_l \quad \leq \quad \beta_u$$

Fourier-Motzkin: Example

Category for x_1?

$$
\begin{aligned}
&(1) \quad x_1 - x_2 \leq 0 \\
&(2) \quad x_1 - x_3 \leq 0 \\
&(3) \quad -x_1 + x_2 + 2x_3 \leq 0 \\
&(4) \quad -x_3 \leq -1
\end{aligned}
$$

Fourier-Motzkin: Example

$$
\begin{array}{ll}
(1) & x_1 - x_2 \leq 0 \\
(2) & x_1 - x_3 \leq 0 \\
(3) & -x_1 + x_2 + 2x_3 \leq 0 \\
(4) & -x_3 \leq -1 \\
\hline
\end{array}
$$

Category for x_1?

Upper bound

Upper bound

Lower bound

eliminate x_1

Fourier-Motzkin: Example

		Category for x_1?
(1)	$x_1 - x_2 \leq 0$	Upper bound
(2)	$x_1 - x_3 \leq 0$	Upper bound
(3)	$-x_1 + x_2 + 2x_3 \leq 0$	Lower bound
(4)	$-x_3 \leq -1$	
		eliminate x_1
(5)	$2x_3 \leq 0$ (from 1,3)	

Fourier-Motzkin: Example

			Category for x_1?
(1)	$x_1 - x_2 \leq 0$		Upper bound
(2)	$x_1 - x_3 \leq 0$		Upper bound
(3)	$-x_1 + x_2 + 2x_3 \leq 0$		Lower bound
(4)	$-x_3 \leq -1$		
			eliminate x_1
(5)	$2x_3 \leq 0$	(from 1,3)	
(6)	$x_2 + x_3 \leq 0$	(from 2,3)	

Fourier-Motzkin: Example

Category for x_1?

(1) ~~$x_1 \quad x_2 \leq 0$~~

(2) ~~$x_1 \quad x_3 \leq 0$~~

(3) ~~$x_1 + x_2 + 2x_3 \leq 0$~~

(4) $-x_3 \leq -1$

---

eliminate x_1

(5) $2x_3 \leq 0$ (from 1,3)

(6) $x_2 + x_3 \leq 0$ (from 2,3)

Fourier-Motzkin: Example

Category for x_1?

(1) ~~$x_1 \quad x_2 \leq 0$~~

(2) ~~$x_1 \quad x_3 \leq 0$~~

(3) ~~$x_1 + x_2 + 2x_3 \leq 0$~~

(4) $-x_3 \leq -1$

---

eliminate x_1

(5) $2x_3 \leq 0$ (from 1,3)

(6) $x_2 + x_3 \leq 0$ (from 2,3)

---

we eliminate x_3

Fourier-Motzkin: Example

Category for x_1?

(1) ~~$x_1 \quad x_2 \leq 0$~~

(2) ~~$x_1 \quad x_3 \leq 0$~~

(3) ~~$x_1 + x_2 + 2x_3 \leq 0$~~

(4) $-x_3 \leq -1$ — Lower bound

---

eliminate x_1

(5) $2x_3 \leq 0$ (from 1,3) — Upper bound

(6) $x_2 + x_3 \leq 0$ (from 2,3) — Upper bound

---

we eliminate x_3

Fourier-Motzkin: Example

Category for x_1?

~~(1)~~	~~$x_1 \quad x_2 \leq 0$~~		
~~(2)~~	~~$x_1 \quad x_3 \leq 0$~~		
~~(3)~~	~~$x_1 + x_2 + 2x_3 \leq 0$~~		
(4)	$-x_3 \leq -1$		Lower bound
			eliminate x_1
(5)	$2x_3 \leq 0$	(from 1,3)	Upper bound
(6)	$x_2 + x_3 \leq 0$	(from 2,3)	Upper bound
			we eliminate x_3
(7)	$1 \leq 0$	(from 4,5)	

$\rightarrow$ Contradiction (the system is UNSAT)

Algorithm: satCheck (only non-strict inequalities, full lazy)

Input: Set C of non-strict linear real arithmetic inequalities over variables $x_1, \ldots, x_n$

Output: Satisfiability of $\bigwedge_{c \in C} c$

Algorithm: satCheck (only non-strict inequalities, full lazy)

Input: Set C of non-strict linear real arithmetic inequalities over variables $x_1, \ldots, x_n$

Output: Satisfiability of $\bigwedge_{c \in C} c$

1. $k := 1$;
2. if $k > n$ then goto 8
3. $C_u := \{(\sum_{j=1}^{n} a_j x_j \leq b) \in C \mid a_k > 0\}$;
4. $C_l := \{(\sum_{j=1}^{n} a_j x_j \leq b) \in C \mid a_k < 0\}$;
5. $C := C \setminus (C_u \cup C_l)$;
6. $C := C \cup \{\frac{b_l}{a_{lk}} - \sum_{j=k+1}^{n} \frac{a_{lj}}{a_{ln}} \cdot x_j \leq \frac{b_u}{a_{uk}} - \sum_{j=k+1}^{n} \frac{a_{uj}}{a_{un}} \cdot x_j \mid$
   $\sum_{j=1}^{n} a_{lj} x_j \leq b_l \in C_l \wedge \sum_{j=1}^{n} a_{uj} x_j \leq b_u \in C_u\}$;
7. goto 2
8. if $\{(l \leq u) \in C \mid l > u\} = \emptyset$ `then return SAT`;
9. else `return UNSAT`;

Strict inequalities

The approach works also if we have both non-strict and strict inequalities. All we need to change is that

Strict inequalities

The approach works also if we have both non-strict and strict inequalities. All we need to change is that

- we distinguish between strict and non-strict lower and upper bounds (defined by strict respectively non-strict inequalities), and
- for each pair of lower and upper bounds, if any of them is strict then we add the constraint

$$\beta_l \quad < \quad \beta_u$$

instead of

$$\beta_l \quad \leq \quad \beta_u \ .$$

Linear integer arithmetic, non-linear real arithmetic

- Question: Does this method work also for linear integer arithmetic, i.e., if the variables range over the integers (instead of the reals)?

Linear integer arithmetic, non-linear real arithmetic

- Question: Does this method work also for linear integer arithmetic, i.e., if the variables range over the integers (instead of the reals)?
- Answer: No.

Linear integer arithmetic, non-linear real arithmetic

- Question: Does this method work also for linear integer arithmetic, i.e., if the variables range over the integers (instead of the reals)?
- Answer: No.
- Reason: The integer domain is not dense.

Linear integer arithmetic, non-linear real arithmetic

- Question: Does this method work also for linear integer arithmetic, i.e., if the variables range over the integers (instead of the reals)?
- Answer: No.
- Reason: The integer domain is not dense.

- Question: Does this method work also for non-linear real arithmetic, i.e., if the variables range over the reals but also multiplication is allowed?

Linear integer arithmetic, non-linear real arithmetic

- Question: Does this method work also for linear integer arithmetic, i.e., if the variables range over the integers (instead of the reals)?
- Answer: No.
- Reason: The integer domain is not dense.

- Question: Does this method work also for non-linear real arithmetic, i.e., if the variables range over the reals but also multiplication is allowed?
- Answer: No.

Linear integer arithmetic, non-linear real arithmetic

- Question: Does this method work also for linear integer arithmetic, i.e., if the variables range over the integers (instead of the reals)?
- Answer: No.
- Reason: The integer domain is not dense.

- Question: Does this method work also for non-linear real arithmetic, i.e., if the variables range over the reals but also multiplication is allowed?
- Answer: No.
- Reason: in general it is not possible to transform constraints containing non-linear polynomial expressions such that we have a single variable on the left-hand-side and a real-arithmetic expression on the right-hand side (we would need complicated case distinctions, fractions and roots).

Complexity

- Worst-case complexity:

$$m \quad \rightarrow$$

Complexity

- Worst-case complexity:

$$m \quad \rightarrow \quad \left(\frac{m}{2}\right)^2 = \frac{m^2}{4}$$

Complexity

- Worst-case complexity:

$$
\begin{aligned}
m &\rightarrow \left(\frac{m}{2}\right)^2 = \frac{m^2}{4} \\
&\rightarrow \left(\frac{\frac{m^2}{4}}{2}\right)^2 = \frac{m^4}{4^3}
\end{aligned}
$$

Complexity

- Worst-case complexity:

$$
\begin{aligned}
m &\rightarrow \left(\frac{m}{2}\right)^2 = \frac{m^2}{4} \\
&\rightarrow \left(\frac{\frac{m^2}{4}}{2}\right)^2 = \frac{m^4}{4^3} \\
&\rightarrow \left(\frac{\frac{m^4}{4^3}}{2}\right)^2 = \frac{m^8}{4^7}
\end{aligned}
$$

Complexity

- Worst-case complexity:

$$
\begin{aligned}
m &\rightarrow \left(\frac{m}{2}\right)^2 = \frac{m^2}{4} \\
&\rightarrow \left(\frac{\frac{m^2}{4}}{2}\right)^2 = \frac{m^4}{4^3} \\
&\rightarrow \left(\frac{\frac{m^4}{4^3}}{2}\right)^2 = \frac{m^8}{4^7} \\
&\rightarrow \ldots
\end{aligned}
$$

Complexity

- Worst-case complexity:

$$
\begin{aligned}
m &\rightarrow \left(\frac{m}{2}\right)^2 = \frac{m^2}{4} \\
&\rightarrow \left(\frac{\frac{m^2}{4}}{2}\right)^2 = \frac{m^4}{4^3} \\
&\rightarrow \left(\frac{\frac{m^4}{4^3}}{2}\right)^2 = \frac{m^8}{4^7} \\
&\rightarrow \cdots \\
&\rightarrow 4 \cdot \left(\frac{m}{4}\right)^{2^d}
\end{aligned}
$$

Complexity

- Worst-case complexity:

$$
\begin{aligned}
m &\rightarrow \left(\frac{m}{2}\right)^2 = \frac{m^2}{4} \\
&\rightarrow \left(\frac{\frac{m^2}{4}}{2}\right)^2 = \frac{m^4}{4^3} \\
&\rightarrow \left(\frac{\frac{m^4}{4^3}}{2}\right)^2 = \frac{m^8}{4^7} \\
&\rightarrow \cdots \\
&\rightarrow 4 \cdot \left(\frac{m}{4}\right)^{2^d}
\end{aligned}
$$

- Heavy!

Complexity

- Worst-case complexity:

$$
\begin{aligned}
m &\rightarrow \left(\frac{m}{2}\right)^2 = \frac{m^2}{4} \\
&\rightarrow \left(\frac{\frac{m^2}{4}}{2}\right)^2 = \frac{m^4}{4^3} \\
&\rightarrow \left(\frac{\frac{m^4}{4^3}}{2}\right)^2 = \frac{m^8}{4^7} \\
&\rightarrow \cdots \\
&\rightarrow 4 \cdot \left(\frac{m}{4}\right)^{2^d}
\end{aligned}
$$

- Heavy!

- The bottleneck: case-splitting

Requirements on theory solver in the SMT context

1. Incrementality?
2. Minimal infeasible subsets?
3. Backtracking?