Satisfiability Checking

13 Gauß and Fourier-Motzkin variable elimination for linear real arithmetic

only "+"

Prof. Dr. Erika Ábrahám

RWTH Aachen University
Informatik 2
LuFG Theory of Hybrid Systems

WS 23/24

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.

$$
\begin{array}{r}
(p_1 = 0 \vee p_2 = 0 \vee p_3 = 0) \wedge p_1 + p_2 + p_3 \geq 100 \wedge \\
(p_1 \geq 5 \vee p_2 \geq 5) \wedge p_3 \geq 10 \wedge p_1 + 2p_2 + 5p_3 \leq 180 \wedge \\
3p_1 + 2p_2 + p_3 \leq 300
\end{array}
$$

Quantifier-free linear real arithmetic (QFLRA)

signature 为一阶理论的概念，一阶理论为一阶逻辑的受限形式，signature为所允许出现的非逻辑符号

Linear real arithmetic is a first-order theory whose signature contains integer constants *const*, real-valued variables x and addition $+$ to build (theory) terms, and strict comparison $<$ to build (atomic) constraints:

term为一阶逻辑概念 小于号<为predicate

Syntax of linear real arithmetic

Predicate(terms)
terms: const, variable 和他们构成的函数

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

- Syntactic sugar: $t_1 = t_2$, $t_1 \leq t_2$, $5 \cdot x + \frac{2}{3} \cdot y < 4$

  乘常数 → 可以转化为 + (如×3)

  都是 linear，只要没有变量·变量就行

- The semantics is standard.

  $\exists x, \varphi(x)$ true $\Longleftrightarrow$ $\forall x, \varphi(x)$ unsat

- 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).

  Quantifier-free Linear real arithmetic

Reduced theory checks for lazy SMT

- Reminder: In (full/less) lazy SMT solving, the theory solver needs to check the satisfiability of sets of constraints (instead of arbitrary Boolean combinations).
- For QFLRA, we convert the input CNF not to negate any constraint by transforming
  - $\neg(t < t')$ to $t \geq t'$,
  - $\neg(t \leq t')$ to $t > t'$,
  - $\neg(t \geq t')$ to $t < t'$,
  - $\neg(t > t')$ to $t \leq t'$ and
  - $\neg(t = t')$ to $(t < t' \vee t > t')$.
- Reduced theory checks: If the input CNF contains no negated constraints then, for each Boolean solution, it is sufficient to check theory consistency for the **true** constraints.

Correctness of reduced theory checks

If the input CNF contains no negated constraints then, for each Boolean solution, it is sufficient to check theory consistency for the **true** constraints.

Proof: Assume QFLRA CNF formula φ with constraints $\mathcal{C}$, none negated. Let φ_{abs} abstract φ by replacing constraints $c \in \mathcal{C}$ by propositions b_c.

First we show that if φ_{abs} has no solution then φ is also unsatisfiable (i.e. UNSAT answers are correct), even if we restrict all theory checks to those constraints that are true under the current Boolean solution.

1. The statement holds for the initial Boolean abstraction φ_{abs}.
2. The abstraction φ_{abs} is changed only when the constraints sent to the theory solver are inconsistent; the returned infeasible subset E is used for the abstraction refinement to $\varphi_{abs} \wedge (\bigvee_{e \in E} \neg b_e)$. Since E is infeasible, the refinement is still conservative (i.e. unsatisfiability of φ_{abs} implies unsatisfiability of φ).

Correctness of reduced theory checks

Second we show that if φ_{abs} has a solution α and the true constraints $C_T = \{c \in \mathcal{C} \mid \alpha(b_c) = true\}$ are consistent then φ has a solution (i.e. SAT answers are correct). So assume a φ_{abs}-solution α with consistent $C := C_T$. We generate a solution for φ by stepwise considering each constraint $c \in \mathcal{C} \setminus C$ as follows:

1. If $C \cup \{\neg c\}$ is consistent then set $C := C \cup \{\neg c\}$.
2. Otherwise, set $C := C \cup \{c\}$ and modify α to assign *true* to c. Note that since φ is constraint-negation-free, c appears only positively in φ, therefore the updated assignment still satisfies φ_{abs}.

After completing the above procedure for each constraint, we get a satisfying assignment α for φ_{abs} such that the true constraints and the negated false constraints build a consistent set, i.e. φ is satisfiable.

Problem statement

Thus the theory solver needs to consider sets $\mathcal{C} = \{c_1, \ldots, c_M\}$ of constraints of the form

$$c_i : \quad \sum_{k=1}^{N} a_{ik} \cdot x_k \sim_i b_i \ ,$$

where

- a_{ik} and b_i are integer (or rational) constants,
- x_k are real-valued variables and
- $\sim_i \in \{=, \leq, <\}$ are comparison operators.

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)

处理等式

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

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

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

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

- 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 weak inequalities in n variables are left (i.e., there are no strict inequalities):

weak inequality: 可以包含等号, 即 ⩽ , ⩾

strong inequality: 不包含等号, 即 <, >

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

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

$$m \text{ constraints} \quad \underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ a_{m1} & a_{22} & \cdots & \cdots & a_{mn} \end{pmatrix}}_{n \text{ variables}} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{pmatrix}$$

Fourier-Motzkin variable elimination

处理高斯线性不等式

- 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 repeats iteratively for each variable x_i, $i = n, \ldots, 1$:
  - For each inequality, bring x_i to stay alone at one side.
  - Collect all lower bounds $\beta_{\ell\,1}, \ldots, \beta_{\ell\,L}$ from inequalities $\beta_{\ell\,j} \leq x_i$.
  - Collect all upper bounds $\beta_{u\,1}, \ldots, \beta_{u\,U}$ from inequalities $x_i \leq \beta_{u\,j}$.
  - $\mathbb{R}$ is dense $\rightsquigarrow$ Constraint set is solvable iff the interval between the highest lower bound and the lowest upper bound is not empty.
  - BUT: bounds are symbolic terms, i.e. no order is given!
  - Solution: Require each lower bound to be less or equal each upper bound.

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

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

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

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

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

$1 \cdot x_1 + (-1) \cdot x_2 + \cancel{0 \cdot x_3} \leq 0$

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

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!

如果只有 upper bound / lower bound，暂时先把式子忽略，回头再将其他变量的结果代入

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

忽略

忽略

将z代入

Fourier-Motzkin variable elimination

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

$$\beta_\ell \quad \leq \quad x_n \quad \leq \quad \beta_u$$

- For each such pair, add the constraint

$$\beta_\ell \quad \leq \quad \beta_u$$

此时变量x被消去

Fourier-Motzkin: Example

			Category for x_1?
$\cancel{(1)}$	$\cancel{x_1 - x_2 \leq 0}$		Upper bound
$\cancel{(2)}$	$\cancel{x_1 - x_3 \leq 0}$		Upper bound
$\cancel{(3)}$	$\cancel{-x_1 + x_2 + 2x_3 \leq 0}$		Lower bound
(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 weak inequalities, full lazy)

Input: Set C of weak linear real arithmetic inequalities over variables $x_1, \ldots, x_n$
Output: Satisfiability of $\bigwedge_{c \in C} c$

1 $k := 0$;

2 if $k = n$ then goto 8 else $k := k + 1$

3 $C_\ell := \{(\sum_{j=1}^{n} a_j x_j \leq b) \in C \mid a_k < 0\}$;

4 $C_u := \{(\sum_{j=1}^{n} a_j x_j \leq b) \in C \mid a_k > 0\}$;

5 $C := C \setminus (C_\ell \cup C_u)$; //case for $a_k = 0$

each lower bound each upper bound

6 $C := C \cup \{ \frac{b_\ell}{a_{\ell k}} - \sum_{j=k+1}^{n} \frac{a_{\ell j}}{a_{\ell k}} \cdot x_j \leq \frac{b_u}{a_{uk}} - \sum_{j=k+1}^{n} \frac{a_{uj}}{a_{uk}} \cdot x_j \mid$
$\sum_{j=1}^{n} a_{\ell j} x_j \leq b_\ell \in C_\ell \wedge \sum_{j=1}^{n} a_{uj} x_j \leq b_u \in C_u \}$;

7 goto 2

8 if $\ell \leq u$ for all $(\ell \leq u) \in C$ `then return` `SAT`; //here, ℓ and u are constants!

9 else `return UNSAT`;

Strict inequalities

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

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

$$\beta_\ell \quad < \quad \beta_u \quad \Leftarrow \quad \beta_l < x \ \wedge \ x \leq \beta_u$$

instead of

$$\beta_\ell \quad \leq \quad \beta_u \ . \quad \Leftarrow \quad \beta_l \leq x \ \wedge \ x \leq \beta_u$$

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)?
  因为给定区间内可能没有整数. e.g. $2.5 \le x \le 2.7$
- 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?
  e.g. $x \cdot y \le 2$: $y=0$: $0 \le 2$; $y \neq 0$: $y>0$: $x \le \frac{2}{y}$; $y<0$: $x \ge \frac{2}{y}$
  not work in general
- Answer: Sound but incomplete.
- 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).
  但是，如果对于 non-Linear，变量是以 Linear 的形式出现：
  e.g. 对于 x: $x + y \cdot z \le 10 \Rightarrow x \le 10 - y \cdot z$ 可以转到一边 ✓

Bonus exercise 19 (8 minutes)

Assume the following constraint set:

$$\{\overset{(1)}{2x - y \le 8},\ \overset{(3)}{2x + y \le -8},\ \overset{(2)}{-2x - y \le 8},\ \overset{(4)}{-2x + y \le 8}\}$$

Eliminating first y and then x using the method of Fourier-Motzkin will result in a single constraint. Which is it (without applying any simplification)?

1. $4 \le 0$
2. $-4 \le 4$
3. $0 \le 4$
4. $0 \le 0$
5. $-4 \le 0$

(1) $2x - 8 \le y$ (3) $y \le -2x - 8$

(2) $-2x - 8 \le y$ (4) $y \le 2x + 8$

(1)+(3) $2x - 8 \le -2x - 8$ $4x \le 0$ $x \le 0$

(1)+(4) $2x - 8 \le 2x + 8$ $(0 \le 16)$

(2)+(3) $-2x - 8 \le -2x - 8$ $(0 \le 0)$

(2)+(4) $-2x - 8 \le 2x + 8$ $-16 \le 4x$ $-4 \le x$

$-4 \le 0$

Complexity

最差情况：一半的上界($\frac{m}{2}$)，一半的下界($\frac{m}{2}$)

$\hookrightarrow (\frac{m}{2}) \times (\frac{m}{2}) = \frac{m^2}{4}$

- Worst-case complexity:

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

dimension，即变量的数量

- Heavy!
- The bottleneck: case-splitting

Requirements on theory solver in the SMT context

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

Learning target

- How does the Gauß method eliminate a variable from a set of QFLRA constraints using an equality?
- How does the Fourier-Motzkin method eliminate a variable from a set of QFLRA constraints?
- What is the complexity of the methods?