CS 433 Automated Reasoning 2025

Lecture 18: Solving for QF_LIA

Instructor: Ashutosh Gupta

IITB India

Compile date: 2025-04-01

Methods for LIA

We will see methods for LIA that are

- Simplex + Cuts (Gomery cut, and Branch and bound)
- Cooper's method (extension of Fourier-Motzkin)
- Omega test method (another extension of Fourier-Motzkin (not covered in detail))

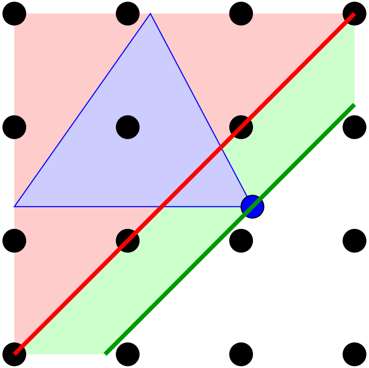
Topic 18.1

Gomery cut

Cuts

Cut: a constraint that chips away non-integral solutions

- In simplex, if current assignment is non-integral,
- find a cut that separates the assignment and integral solutions



Simplex for integers

Recall our normal form for the input problem

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i.$$

l_i and u_i are $+\infty$ and $-\infty$ if there is no lower and upper bound, respectively.

In the following presentation of Gomery cut, we assume that

- at least one bound is finite for each variable and
- all finite bounds are integral.

Exercise 18.1

How can we ensure that introduced slack variables have integral bounds? They are not required to be integers.

Simplex+Gomery cut

Gomery cut chips away non-integer parts of the solution space.

The algorithm proceeds as follows

1. Run simplex as if all variables are rationals and find an assignment v
2. if v is integral, **return** v
3. if for some $i \in B$, $v(x_i)$ is not integer then add a constraint to eliminate the neighbouring non-integer space.

Consider the row k_i of A, $x_i = \sum_{j \in NB} a_{k_i j} x_j$.

An integral solution must satisfy the equality

Wlog, we assume all upper bounds are active for the nonbasic variables.

$$v(x_i) := \sum_{j \in NB} a_{k_i j} u_j$$

After subtracting the two equations,

$$v(x_i) = x_i + \sum_{j \in NB} a_{k_i j}(u_j - x_j).$$

Simplex+Gomery cut (II)

$\{\delta\} = \delta - \lfloor \delta \rfloor$

Consider inequality: $\{v(x_i)\} \leq \sum_{j \in NB} \{a_{k_ij}\}(u_j - x_j)$

Claim: v does not satisfy the above.

1. Since $v(x_i)$ is not an integer, $\{v(x_i)\}$ is positive.
2. Under v the rhs is 0.(Why?)

Claim: An integral solution of input satisfies the above.

1. Any integral solution x satisfies $v(x_i) = x_i + \sum_{j \in NB} a_{k_ij}(u_j - x_j)$.
2. $\sum_{j \in NB} \{a_{k_ij}\}(u_j - x_j) \geq 0$(Why?)
3. For integral x, $\{v(x_i)\} = \{\sum_{j \in NB} \{a_{k_ij}\}(u_j - x_j)\}$
4. Due to 2 and 3, $\{v(x_i)\} \leq \sum_{j \in NB} \{a_{k_ij}\}(u_j - x_j)$

Therefore, the inequality separates v from the integral solutions. We add the above inequality in simplex and run it again.

Exercise 18.2

If v is active at some active lower bounds or no bounds, how the above will change?

Commentary: There are many ways to formulate Gomery cut. In Decision Procedure 2nd Ed. section 5.3.1, you may find another scheme for Gomery cut. Here is another cut scheme: Cuts from Proofs, Dillig et. el., CAV 2009

Branch and bound: Unbounded cases

Let us suppose there is a nonbasic variable that has no bounds.

We can not apply Gomery cut. We may need to case split.

We generate two simplex problems with the following two inequalities respectively.

- $x_i \leq \lfloor v(x_i) \rfloor$
- $x_i \geq \lceil v(x_i) \rceil$

We have removed space $\lceil v(x_i) \rceil > x_i > \lfloor v(x_i) \rfloor$

Solve the two problems separately.

The splits are called branch and bound method.

Example : Gomery cut

Example 18.1

Consider the following simplex state

$$x = 1.5y + z \qquad \text{Bounds } 0 \leq y \leq 3^*,\ 0 \leq z \leq 1^*,\ 0 \leq x \leq 6$$

Current solution: $5.5 = 1.5 * 3 + 1 * 1$

Subtracting equations

$$5.5 - x = 1.5(3 - y) + 1 * (1 - z)$$

Gomery cut

$$.5 \leq .5(3 - y)$$

After simplification : $y \leq 2$

Example:

After updating the bounds

$$x = 1.5y + z \qquad \text{Bounds } 0 \leq y \leq 2^*,\ 0 \leq z \leq 1^*,\ 0 \leq x \leq 6$$

It is not necessary that we get bound on one variable. However, here we are lucky. If not, then we may need to introduce slack variable.

If we had chosen, $0 \leq .5(3 - y)$ it would have satisfied the current corner.

Topic 18.2

Cooper's method

Cooper's method

Cooper's method is one of the well known decision procedure for Presburger arithmetic.

This method proceeds by quantifier elimination.

However, the arithmetic does not allow quantifier elimination as it is.

Example 18.2

The following formula states that y is odd.

$$\exists x. 2x + 1 = y$$

This can not be stated in the arithmetic.

Adding $|$ operator to enable quantifier elimination

We need to introduce modulo operator $|$ that expresses divisibility.

$$k|y \text{ means } k \text{ divides } y, \text{ where } k \in \mathbb{Z}^+$$

$\mathcal{T}'_{\mathbb{Z}}$ We also need to add the following axiom about $|$ in the theory.

$$\forall y.\ k|y \Leftrightarrow \exists x.\ kx = y$$

Example 18.3

Now we can eliminate existential quantifier

$$(\exists x.2x + 1 = y) \equiv 2|(y + 1)$$

We may not write the parenthesis

Exercise 18.3

Give an x that satisfies $2|x + 1 \wedge 3|x + 5 \wedge \neg 5|x - 2$

Cooper's method

Input: $F_1 := \exists x. A_1 \land ... \land A_n$, where A_i is a literal.

The method proceeds in fours steps

- Normalize literals
- Separate out x
- Scale up coefficients of x
- Replace x with x' such that no coefficient to x'
- Eliminate x'

In some notation, we will use formulas like F_1 as set of literals.

Cooper's method : Normalize literals

The literals must be in one of the following forms

- $s < t$
- $k|t$
- $\neg(k|t)$

We may normalize literals as follows and obtain F_2

- $s = t \equiv s < t + 1 \wedge t < s + 1$
- $s \neq t \equiv s < t \vee t < s$
- $\neg s < t \equiv t < s + 1$

Example 18.4

Consider $F_1 := 3x = 6y + 3$

After normalization we obtain, $F_2 = 3x < 6y + 3 + 1 \wedge 3x + 1 > 6y + 3$

Cooper's method : separate out x

For $h \in \mathbb{Z}^+$ and t does not contain x, rewrite terms in literals of F_2 until they are in one of the following forms.

- $hx < t$
- $t < hx$
- $k|hx + t$
- $\neg(k|hx + t)$

We obtain F_3 after this transformation.

Example 18.5

Consider $F_2 := 2x + 3y < 6 \wedge -2x + 3y < 6 \wedge 3| - 5x + 2$.

$$F_3 := 2x < 6 - 3y \wedge -6 + 3y < 2x \wedge 3|5x - 2.$$

Cooper's method : scale up coefficients of x

Let

$$\lambda = lcm\{h | h \text{ is coefficient of } x \text{ in some literal }\}$$

We scale up all literals in F_3 as follows and obtain F_4.

- $hx < t \equiv \lambda x < \lambda' t$
- $t < hx \equiv \lambda' t < \lambda x$
- $k | hx + t \equiv \lambda' k | \lambda x + \lambda' t$
- $\neg(k | hx + t) \equiv \neg(\lambda' k | \lambda x + \lambda' t)$

where $\lambda' h = \lambda$.

Example 18.6

Consider $F_3 = 2x < z + 1 \wedge y - 3 < 3x \wedge 4 | 5x + 1$.

$\lambda = lcm\{2, 3, 5\} = 30$.

Therefore, $F_4 = 30x < 15z + 15 \wedge 10y - 30 < 30x \wedge 24 | 30x + 6$.

Cooper's method : replace x to remove coefficient

We aim to remove coefficients of x.

We substitute λx by x' in the formula

We also need to say that x' is divisible by λ.

We obtain

$$F_5 := F_4[\lambda x \mapsto x'] \wedge \lambda | x'.$$

Example 18.7

$F_4 = 30x < 15z + 15 \wedge 10y - 30 < 30x \wedge 24 | 30x + 6$.

After replacement:
$F_5 = x' < 15z + 15 \wedge 10y - 30 < x' \wedge 24 | x' + 6 \wedge 30 | x'$.

Cooper's method : eliminate x'

- $M := \{A \in F_5 |\ A = (k|t) \text{ or } A = \neg(k|t)\}$.
- $UB := \{x' < t | x' < t \in F_5\}$,
- $LB := \{t < x' | t < x' \in F_5\}$, and
- $\delta := lcm\{k|\ (k|t) \text{ or } \neg(k|t) \text{ in } M\}$.

Now we have two cases.

- $LB = \emptyset$
- $LB \neq \emptyset$

case $LB = \emptyset$

- $\delta := lcm\{k |\ (k|t) \text{ or } \neg(k|t) \text{ in } M\}$,

Since there are no lower bounds in F_5, there is some x' that satisfies the upper bounds in F_5.

We only need to check that the mod literals are mutually satisfiable.

In every δ interval there must be a satisfying assignment.

Therefore, the following is an equivalent and quantifier-free formula.

$$F_6 := \bigvee_{i=1}^{\delta} M[x' \mapsto i]$$

Example : $LB = \emptyset$

Example 18.8

Consider the following formula with no lower bound: $F_5 = x' < 15z + 15 \land 6|x' - y + 6 \land 9|x'$.

Since we can always choose small enough x' to satisfy $x' < 15z + 15$, we can ignore the literal.

$\delta = lcm\{6, 9\} = 18$.
In every interval of 18, one value of x' must satisfy the mod literals.

Therefore, the following is an equivalent and quantifier-free formula.

$$\bigvee_{i=1}^{18} 6|i - y + 6 \land 9|i$$

Exercise 18.4

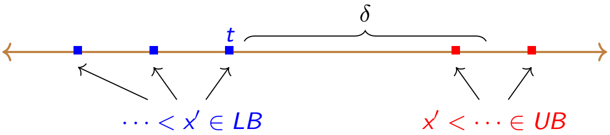
Simplify the above formula

case $LB \neq \emptyset$

Let us suppose $x' = m$ satisfies F_5. So, m is greater than the largest lower bound.

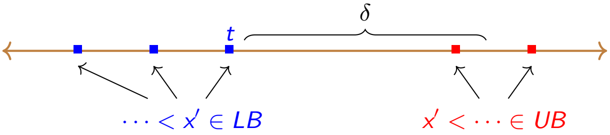
Let $t < x' \in LB$ be the largest lower bound.

- $LB[x' \mapsto t + 1]$ is true
- Since there is a satisfying assignment, $UB[x' \mapsto t + 1]$ is true. Furthermore, there is b such that
  - $UB[x' \mapsto t + i]$ is true for $1 \leq i \leq b$
  - $UB[x' \mapsto t + i]$ is false for $i > b$
- One of $M[x' \mapsto t + 1], .., M[x' \mapsto t + \delta]$ must be true (divisibility argument again)



Exercise 18.5
Where is b in the above drawing?

case $LB \neq \emptyset$ II



However, we do not know which lower bound is maximum!

Therefore, one of the disjuncts in the following formula must be true.

$$F_6 := \bigvee_{t < x' \in LB} \bigvee_{i=1}^{\delta} F_5[t + i]$$

Example : $LB \neq \emptyset$

Example 18.9

Consider the following formula with lower bounds:
$F_5 = x' < 15z + 15 \land 10y - 30 < x' \land 24 | x' + 6 \land 30 | x'$.

$\delta = lcm(24, 30) = 120$

Since $LB := \{10y - 30 < x'\}$

$$F_6 := \bigvee_{i=1}^{120} 10y - 30 + i < 15z + 15 \land 10y - 30 < 10y - 30 + i \land 24 | 10y - 30 + i + 6 \\ \land 30 | 10y - 30 + i$$

After simplification, $F_6 := \bigvee_{i=1}^{120} 10y - 45 + i < 15z \land 24 | 10y - 24 + i \land 30 | 10y - 30 + i$

Exercise: UB vs LB

Topic 18.3

Omega test method

Omega test method

This method is another twist on Fourier-Motzkin to solve for integers

The key issue remains the same. We need a gap between lower bounds and upper bounds such that we can choose appropriate x' in δ.

Commentary: In Decision Procedure 2nd Ed. section 5, you may find detail description pf omega test.

End of Lecture 18