Linear Arithmetic

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

24 March 2025

Outline

Linear Arithmetic (KS Ch 5)

- Boolean combinations of linear constraints of the form:

$$a_1 x_1 + \cdots + a_n x_n \leq b_1$$

- Quantifier-Free fragment of $FO(+, -, <, 0, 1)$
- Interpretation of $+, -, <, 0, 1$ fixed; Domain is $\mathbb{R}$, $\mathbb{Q}$, or $\mathbb{Z}$.

Linear Arithmetic syntax

(Formula) $\varphi ::= \mathit{Atom} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi$
(Atom) $\mathit{Atom} ::= \mathit{Term} < \mathit{Term} \mid \mathit{Term} = \mathit{Term}$
(Term) $\mathit{Term} ::= \mathit{Var} \mid \mathit{Const} \mid \mathit{Term} + \mathit{Term} \mid \mathit{Term} - \mathit{Term}$

Examples

Example formula φ_1

$$
\begin{aligned}
x = 19 \land \neg(x \leq 20)\ \lor \\
x \leq 20 \land x \geq 10 \land z = -1 \land x' = x + z \land \neg(x' \leq 20)\ \lor \\
x \leq 20 \land y = 15 \land \neg(x \geq 10) \land \neg(y \geq x')
\end{aligned}
$$

Example conjunctive formula φ_2

$$
\begin{aligned}
x + y < 1\ \land \\
0 < x\ \land \\
0 < y
\end{aligned}
$$

Question we want to answer: Satisfiability.

Importance of Linear Arithmetic

Many practical applications. In Verification:

- Loop invariants, polyhedral data-flow analysis of programs
- Compiler Optimization
- Analysis/Model-Checking of timed, hybrid, dynamical systems.
- Symbolic Execution/Simulation (representation of reachable states).
- Winning Strategies in 2-player Games, Controller Synthesis.

Example: Loop optimization (loop hoisting)

```
for (i = 1; i <= 10; i++)
  a[j+i] := a[j];
```

```
// R1 has i, R2 has j
// loop body
  1. R4 := mem[a+R2];
  2. R5 := R2 + R1;
  3. mem[a+R5] := R4;
  4. R1 := R1 + 1;
```

Statement 1 can be hoisted out of loop if foll constraint is unsat:

$$1 \leq i \wedge i \leq 10 \wedge i + j = j$$

Loop Parallelization

Program:

```
for i = 1 to 100 do
  for j = 1 to 100 do
    A[i,j+i] := A[100,j];
```

Constraints on writes (i_1', j_1'):

$$\begin{aligned} 0 &\leq i_1 \leq 100 \\ 0 &\leq j_1 \leq 100 \\ i_1' &= i_1 \\ j_1' &= j_1 + i_1 \end{aligned}$$

Constraints on reads (i_2', j_2'):

$$\begin{aligned} 0 &\leq i_2 \leq 100 \\ 0 &\leq j_2 \leq 100 \\ i_2' &= 100 \\ j_2' &= j_2 \end{aligned}$$

Check overlap:

$$\begin{aligned} i_1' &= i_2' \\ j_1' &= j_2' \end{aligned}$$

If constraints are UNSAT then we can parallelize the loop.

Checking Verification Conditions

Floyd-Hoare style verification of programs:

```
int x = 19;
int y = 15;
// inv: x <= 20
while (x >= 10) {
  z = -1;
  x = x + z;
}
assert(y >= x);
```

Is the formula: $\forall x, \forall y, \forall z, \forall x'$:

$$\begin{aligned}(x = 19 \land y = 15) &\implies x \leq 20 \land \\ (x \leq 20 \land x \geq 10 \land z' = -1 \land x' = x + z') &\implies x' \leq 20 \land \\ (x \leq 20 \land \neg(x \geq 10)) &\implies y \geq x\end{aligned}$$

valid?

Fourier-Motzkin Elimination (KS Sec 5.4, Schrijver Sec 12.2)

- Fourier 1827, Dines 1917, Motzkin 1936.
- Works for $\mathbb{R}$ and $\mathbb{Q}$ domains.
- Consider conjunctions of linear constraints
- Can check satisfiability, find a solution, eliminate variables (geometric projection, $\exists$-elimination)

General form

Suppose we want to eliminate x_1 from the system of ineqs (1):

$$a_{11}x_1 + \cdots + a_{1n}x_n \leq b_1$$
$$a_{21}x_1 + \cdots + a_{2n}x_n \leq b_2$$
$$\cdots$$
$$a_{m1}x_1 + \cdots + a_{mn}x_n \leq b_m$$

1. Make coeffs of x_1 1, -1, or 0, by scaling by a pos constant to get Ineq (2).
2. Write Ineq (2) as Ineq (3):

$$x_1 \leq b'_1 - (a'_{11}x_2 + \cdots + a'_{1n}x_n) \; (m' \text{ ineqs}) \quad (1)$$
$$-x_1 \leq b'_{m'+1} - (a'_{m'+1,1}x_2 + \cdots + a'_{m'+1,n}x_n) \; (m'' - m' \quad (2)$$
$$a_{m''+1,2}x_2 + \cdots + a_{m''+1,n}x_n \leq b_{m''+1} \; (m - m'' \text{ ineqs}) \quad (3)$$

Fourier-Motzkin contd.

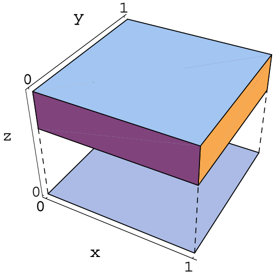
3. Remove constraints of type (1) and (2). Note that constraints of type (3) are retained.
4. Add all combinations of -RHS(2) $\leq$ RHS(1) constraints.
5. Let Ineq (4) be obtained thus.

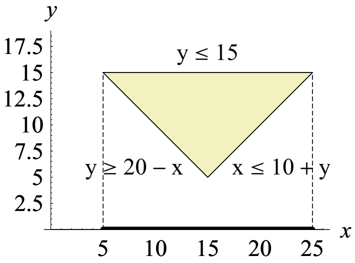
Claim: Ineq (4) represents the projection of the solution set of Ineq (1) to the dimensions $x_2, \ldots, x_n$.

6. Repeat till we get constraints in single variable x_n. Check if the constraints are satisfiable (lower bounds $\leq$ upper bounds). If sat, output SAT; else output UNSAT.

   As a corner case, we may get an empty set of contraints after eliminating a variable. In this case the conjunction of the (empty set of) constraints is *true*. Return SAT.

Examples illustrating projection





$$0 \leq x \leq 1$$
$$0 \leq y \leq 1$$
$$0.75 \leq z \leq 1$$

$$y \leq 15$$
$$y \geq 20 - x$$
$$x \leq 10 + y$$

Example

Given system of ineq:

$$y \leq 15$$
$$y \geq 20 - x$$
$$x \leq 10 + y$$

Rewrite in general form: (Ineq (1))

$$y \leq 15$$
$$-x - y \leq -20$$
$$x - y \leq 10$$

Rewrite: (Ineq (2))

$$y \leq 15$$
$$-x + 20 \leq y$$
$$x - 10 \leq y$$

Eliminate y: (Ineq (3))

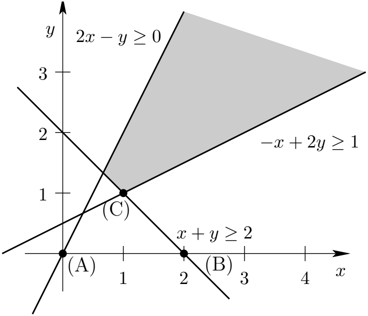
$$-x + 20 \leq 15$$
$$x - 10 \leq 15$$

That is: $5 \leq x \leq 25$. Hence original system of ineqs is satisfiable.

One solution is $x \mapsto 10, y \mapsto 12$.

Exercise

Eliminate x from the system of inequalities:



Correctness claims

The projection of a set S of n-dimension vectors to dimensions 2 to n is defined to be

$$\{(a_2, \ldots, a_n) \mid \exists a_1 \text{ such that } (a_1, a_2, \ldots, a_n) \in S\}.$$

- Ineq (4) represents the projection of the solution set of Ineq (1).
- If Algo reports SAT, then the solution set to Ineq (1) is non-empty; else it is empty.

Some observations on FM Elimination

- Finding a solution: substitute backwards.
- Complexity
  - Number of constraints can blow up from m to m^2 in one iteration.
  - Number of constraints can be exponential in n (See Schrijver p156)
- Linear real arithmetic admits quantifier- elimination.
  - Given formula $\exists x \varphi$, there exists a formula φ' such that

$$\exists x \varphi \equiv \varphi' \text{ (modulo } (\mathbb{R}, +, -, <, 0, 1) \text{ structure)}$$

- Gives us a decision procedure for $Th(\mathbb{R}, +, -, <, 0, 1)$. Why?

Integer Linear Arithmetic

Given a system of linear inequalities Ineq (1):

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &\le b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n &\le b_2 \\ &\cdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &\le b_m \end{aligned}$$

Let us also allow equality ("=") constraints explicitly. Is there an integer-valued solution to Ineq (1)?

Integer Linear Arithmetic

Given a system of linear inequalities Ineq (1):

$$
\begin{aligned}
a_{11}x_1 + \cdots + a_{1n}x_n &\leq b_1 \\
a_{21}x_1 + \cdots + a_{2n}x_n &\leq b_2 \\
&\cdots \\
a_{m1}x_1 + \cdots + a_{mn}x_n &\leq b_m
\end{aligned}
$$

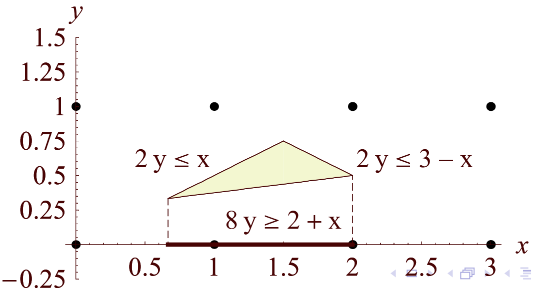
Let us also allow equality ("=") constraints explicitly. Is there an integer-valued solution to Ineq (1)?

How do we answer this?
Is the problem decidable (brute-force procedure)?

Example 1

$$2y \leq x$$
$$8y \geq 2 + x$$
$$2y \leq 3 - x$$

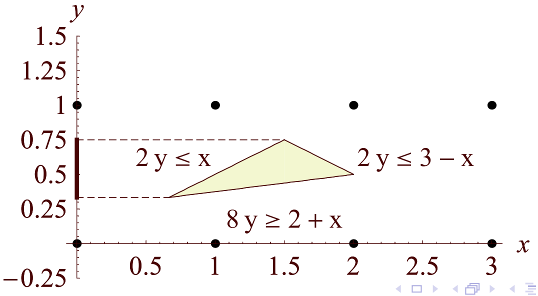
Eliminate y:



Example 1

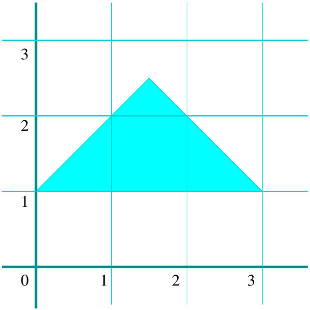
$$2y \leq x$$
$$8y \geq 2 + x$$
$$2y \leq 3 - x$$

Eliminate x:



Another example: All projections non-empty

$$y < x + 1$$
$$y > 1$$
$$y < 4 - x$$



... but no integer solution!

Overall Idea of Omega Test (Pugh 1991)

- Given a system of linear constraints C, the Omega Test algorithm is recursive.
- Adaptation of Fourier-Motzkin Elimination for integer solutions.
- Handle equality constraints separately (use them to eliminate variables as long as equality constraints remain).

Basic idea:

1. (Base case) If C has only one variable, check it for integer solutions and return "Yes" / "No".
2. Eliminate equality constraints and variables along with them.
3. (Recursive step) Reduce question of integer solution to C with n variables, to question of integer solution to C' with $n-1$ variables.

Eliminating equality constraints

Consider following proposal:

If we have a constraint

$$a_1 x_1 + \cdots + a_n x_n = b \quad (a_1 \neq 0)$$

Substitute

$$x_1 = \frac{1}{a_1}(b - a_2 x_2 - \cdots - a_n x_n)$$

in remaining constraints to get projection to $x_2, \ldots, x_n$.

Eliminating equality constraints

Consider example

Problem with equality elimination

$$x = y/2 \quad (1)$$

$$0 < x < y < 2 \quad (2)$$

Substitution of $x = y/2$ in (1) gives

$$0 < y/2 < y < 2$$

which has an integer solution $y \mapsto 1$, but gives us $x \mapsto 0.5$.

Preprocessing the constraints

- Make coefficients (including b_i's) integral, by multiplying by lcm of denominators of rational coefficients.
- Normalize by dividing by gcd of variable coefficients.
- If any equality constraint RHS is fractional, return UNSAT.
- For inequalities with fractional RHS, replace RHS by $\lfloor RHS \rfloor$.

All coefficients and RHS's are integral now, and we will maintain this property.

Eliminating Equality Constraints

Suppose we are given:

$$a_1x_1 + \cdots + a_nx_n = b \tag{1}$$

$$\boxed{\text{(other constraints)}} \tag{2}$$

1. If some x_i has coeff 1 or -1 in (1), substitute for x_i in (2) and discard (1). [Projection of solutions is preserved]
2. If not, choose x_i with least absolute value of coefficient (say x_1), and add constraint with new variable α, where $m = |a_1| + 1$:

$$m\alpha = (a_1 \, mod \, m)x_1 + \cdots + (a_n \, mod \, m)x_n - (b \, mod \, m) \tag{3}$$

3. Coeff of x_1 will be 1 or -1. Eliminate by substituting. Coefficients of other x_i's reduce by $\frac{5}{6}$ at least.
4. Go back to Step 1.

Correctness

Claim

Projection of solutions to (1,2,3) is solutions to (1,2).

Use fact that

$$\frac{a}{m} = \lfloor \frac{a}{m} \rfloor + \frac{(a \; mod \; m)}{m}.$$

Suppose $d_1, \ldots, d_n$ is an integer solution to (1).

$$a_1 d_1 + \cdots + a_n d_n - b = 0$$

$$m \cdot [(\lfloor \frac{a_1}{m} \rfloor d_1 + \cdots + \lfloor \frac{a_n}{m} \rfloor d_n - \lfloor \frac{b}{m} \rfloor)]$$

$$+(a_1 \; mod \; m) d_1 + \cdots + (a_n \; mod \; m) d_n - (b \; mod \; m) = 0$$

Therefore $e, d_1, \ldots, d_n$ is an integer solution to:

$$m\alpha = (a_1 \; mod \; m) x_1 + \cdots + (a_n \; mod \; m) x_n - (b \; mod \; m) \qquad (3)$$

Note on $\widehat{mod}$

Usual notion of "*mod*": For integers a and b, find integers q and r such that $a = b \cdot q + r$ and $0 \leq r < |b|$.

Thus $11 \; mod \; 5$ is 1 and $-11 \; mod \; 5$ is 4.

In Omega Test we use $\widehat{mod}$:

$$a \,\widehat{mod}\, b \quad = \quad \begin{cases} (a \; mod \; b) & \text{if } (a \; mod \; b) < b/2 \\ (a \; mod \; b) - b & \text{otherwise.} \end{cases}$$

Thus

- $11 \,\widehat{mod}\, 5$ is 1
- $13 \,\widehat{mod}\, 5$ is -2
- $-11 \,\widehat{mod}\, 5$ is -1.

Example[1]

$$7x + 12y + 31z = 17$$
$$3x + 5y + 14z = 7$$

substitution	resulting constraints
$x = -8\alpha - 4y - z - 1$	$-7\alpha - 2y + 3z = 3$ $-24\alpha - 7y + 11z = 10$
$y = \alpha + 3\beta$	$-3\alpha - 2\beta + z = 1$ $-31\alpha - 21\beta + 11z = 10$
$z = 3\alpha + 2\beta + 1$	$2\alpha + \beta = -1$
$\beta = -2\alpha - 1$	

[1] from [Pugh 1991]

Omega Test

OmegaTest(C):
  If (C is over single var)
    Return SAT/UNSAT accordingly.
  $C_R = Elim(C, v)$;
  If ($OmegaTest(C_R)$ = UNSAT)
    Return UNSAT;
  $C_D = DarkShadow(C, v)$;
  If ($OmegaTest(C_D)$ = SAT)
    Return SAT;
  $C_G^1, \ldots, C_G^k = GreyShadow(C, v)$;
  If ($OmegaTest(C_G^i)$ = SAT for any i)
    Return SAT;
  Return UNSAT;

Revisiting Fourier-Motzkin

$$y \leq x + 1 \quad (1)$$
$$y \leq -x + 5 \quad (2)$$
$$3y \geq -x + 7 \quad (3)$$

Rewriting (to eliminate y):

$$\begin{array}{rcl} y & \leq & x + 1 \\ y & \leq & -x + 5 \\ -\frac{x}{3} + \frac{7}{3} & \leq & y \end{array}$$

Rewriting (after eliminating y):

$$\begin{array}{rcl} -\frac{x}{3} + \frac{7}{3} & \leq & x + 1 \\ -\frac{x}{3} + \frac{7}{3} & \leq & -x + 5 \end{array}$$

Consider solutions to C' and y on numberline.

Illustrating shadow regions

$$y \leq x + 1 \quad (1)$$
$$y \leq -x + 5 \quad (2)$$
$$3y \geq -x + 7 \quad (3)$$

Real shadow (Eliminate y):

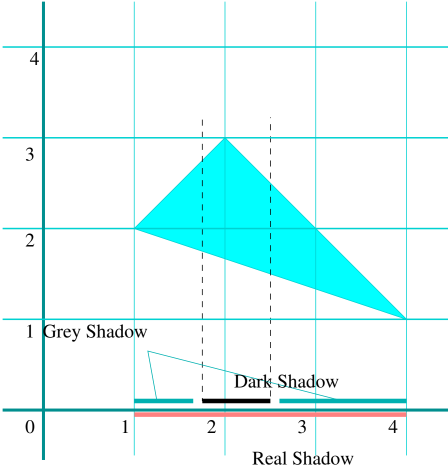
$$\frac{1}{3}(7 - x) \leq x + 1$$
$$\frac{1}{3}(7 - x) \leq -x + 5$$

This gives us $1 \leq x \leq 4$.
Dark shadow:

$$\frac{1}{3}(7 - x) + 1 \leq x + 1$$
$$\frac{1}{3}(7 - x) + 1 \leq -x + 5$$

This gives us $1.75 \leq x \leq 2.5$.

Grey shadow (real - dark):



Checking the Grey Shadow

Suppose the variable we are trying to eliminate is z. Consider any "lower bound" constraint c on z, say:

$$ax + by + d \leq cz.$$

Look for a solution in which the value of z is within a distance of 1 from the lower bound:

$$ax + by + d \leq cz < ax + by + d + c \tag{4}$$

Replace above constraint by each of the equality constraints:

$$ax + by + d = cz \tag{1}$$

$$ax + by + d + 1 = cz \tag{2}$$

$$\ldots$$

$$ax + by + d + (c - 1) = cz \tag{3}$$

Call the resulting system of constraints $C_G^0, \ldots, C_G^{c-1}$. Check each one of them separately for integer solutions. Note that z now has an equality constraint, and we can use equality elimination to eliminate z.

Do this for each lower bound constraint till a solution is found.

Observations

- Checking the grey shadow for integer solutions is a complete test on its own.
- What about projection of integer solutions, for the purpose of quantifier elimination?

Example[2]

$$3 \leq 11x + 13y \leq 21$$
$$-8 \leq 7x - 9y \leq 6$$

$$3 - 13y \leq 11x \leq 21 - 13y$$
$$9y - 8 \leq 7x \leq 9y + 6$$

P'

lower bound	upper bound	unnormalized combination
$33 - 143y \leq 121x$	$121x \leq 231 - 143y$	$198 \geq 0$
$21 - 91y \leq 77x$	$77x \leq 99y + 66$	$190y + 45 \geq 0$
$63y - 56 \leq 49x$	$49x \leq 63y + 42$	$98 \geq 0$
$99y - 88 \leq 77x$	$77x \leq 147 - 91y$	$235 \geq 190y$

P''

lower bound	upper bound	unnormalized combination
$(33 - 143y) + 100 \leq 121x$	$121x \leq 231 - 143y$	$98 \geq 0$
$(21 - 91y) + 60 \leq 77x$	$77x \leq 99y + 66$	$190y \geq 15$
$(63y - 56) + 36 \leq 49x$	$49x \leq 63y + 42$	$62 \geq 0$
$(99y - 88) + 60 \leq 77x$	$77x \leq 147 - 91y$	$175 \geq 190y$

[2] from [Pugh 1991]