

Arithmetic

Presburger Arithmetic

- The theory of integers with $+$, $-$, $=$, $>$
 - Quantifiers are omitted
- The most useful in program verification
 - And quite useful for program analysis also
- Example of a satisfiability problem:

$$y > 2x + 1 \wedge y + x > 1 \wedge y < 0$$
- Satisfiability of a system of linear inequalities
 - A polynomial problem (when looking for rational solutions)
 - Some of the algorithms are quite simple
 - If we add the requirement that solutions are in \mathbb{Z} then the problem is NP-complete

Fourier-Motzkin Elimination

- Pick a variable x_1
- Partition inequalities by the sign of x_1

$$(1) \quad x_1 - P_i(x_2, \dots, x_k) \geq 0 \quad i = 1 \dots n$$

$$(2) \quad -x_1 + Q_j(x_2, \dots, x_k) \geq 0 \quad j = 1 \dots m$$

$$R_k(x_2, \dots, x_k) \geq 0$$
 - Coefficients of x_1 normalized
- Add (1) and (2) to get

$$Q_j(x_2, \dots, x_k) - P_i(x_2, \dots, x_k) \geq 0$$

$$R_k(x_2, \dots, x_k) \geq 0$$
- Repeat until all variables eliminated

$$c \geq 0$$

Fourier-Motzkin Elimination: example

- Initial inequalities

$$\begin{array}{l} 3x \geq 2y \quad | \times 2 \\ 3y \geq 4 \quad | \times 3 \\ 3 \geq 2x \quad | \times 3 \end{array} \quad \left. \begin{array}{l} \text{In } \mathbb{Z} \text{ we get:} \\ x \leq 1 \\ y \geq 2 \\ \text{then:} \\ 3x \geq 4 \Rightarrow x \geq 2 \end{array} \right\}$$
- Eliminating x

$$\begin{array}{l} 3y \geq 4 \quad | \times 4 \\ 9 \geq 4y \quad | \times 3 \end{array}$$
- Eliminating y

$$27 \geq 16 \quad \Rightarrow \text{satisfiable (if procedure is sound)...}$$

...but only in \mathbb{Q} , not in \mathbb{Z}

Soundness of Elimination Procedure

Claim: *initial inequalities satisfiable iff satisfiable after elimination of x_1*

Proof:

- \Rightarrow This direction is straightforward
- \Leftarrow Given Ψ satisfies inequalities with x_1 eliminated, let

$$\Psi(z) = \begin{cases} \min_j \Psi(Q_j(x_2, \dots, x_k)) & z \equiv x \\ \Psi(z) & \text{otherwise} \end{cases}$$

- Clearly $\Psi(x_1) \leq \Psi(Q_j(x_2, \dots, x_k))$ for all j
- Since mutated inequalities are satisfied we also get that $\Psi(x_1) \geq \Psi(P_i(x_2, \dots, x_k))$ for all i

Fourier-Motzkin: epilogue

- Incurs exponential growth in number of inequalities
- Not used in practice...

Feasibility Theorem (Farkas lemma)

Theorem: a system of inequalities $A\bar{x} \geq \bar{b}$ is not satisfiable iff exists $\bar{c} \geq \bar{0}$ such that

$$\begin{aligned} \bar{c}^T A &= \bar{0}^T \\ \bar{c}^T \bar{b} &> 0 \end{aligned}$$

is satisfiable.

Feasibility Theorem: proof

Proof:

\Leftarrow given the premises hold, assume that $A\bar{x} \geq \bar{b}$ is satisfiable, then multiplying by \bar{c} should retain satisfiability

\Rightarrow elimination procedure serves as constructive proof:

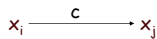
- assume $A\bar{x} \geq \bar{b}$ is unsat
- system remains unsat after each elimination step
- eventually obtain $\bar{c} \geq \bar{0}$ where $\bar{c} \geq \bar{0}$

but every synthesized inequality is a non-negative linear combination of input inequalities

Difference Constraints (Pratt)

- A special case of linear arithmetic
- All constraints of the form:
 - $x_i - x_j \leq c$ or $x_i - 0 \leq c$ or $0 - x_j \leq c$
- The most common form of constraint

- Construct a directed graph with:
 - A node for 0
 - A node for each variable x_i
 - An edge from x_i to x_j of weight c for each $x_i - x_j \leq c$



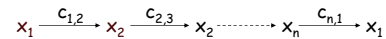
Difference Constraints

Theorem:

A set of difference constraints is satisfiable iff there is no negative weight cycle in the graph

Proof:

\Rightarrow (soundness) assume a negative cycle exists



s.t. $c_{1,2} + c_{2,3} + \dots + c_{n,1} < 0$

Difference Constraints: soundness (2)

But there is an assignment that satisfies

$$\begin{aligned} x_1 - x_2 &\leq c_{1,2} \\ x_2 - x_3 &\leq c_{2,3} \\ &\vdots \\ x_n - x_1 &\leq c_{n,1} \end{aligned}$$

$$0 \leq c_{1,2} + c_{2,3} + \dots + c_{n,1} \Rightarrow \text{contradiction}$$

Difference Constraints: completeness

\Leftarrow Notation: δ_{ij} is length of the shortest path $x_i \rightsquigarrow x_j$

- Set to ∞ if no such path exists
- Well-defined iff no negative weight cycles exist

Lemma: if C is satisfiable then $\delta_{ij} = \max_{\psi \models C} \psi(x_i - x_j)$

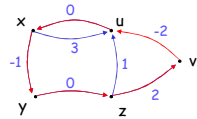
Proof:

- Given a satisfying interpretation ψ , adding satisfied constraints along path from x_i to x_j we get that $\psi(x_i - x_j) \leq \delta_{ij}$
 - Construct assignment ψ by $\psi(x_i) = \psi(x_j) + \delta_{ij}$ for all $\delta_{ij} < \infty$ (other pairs infer no transitive constraints)
- Then for some $x_i - x_m \leq a$ in C , $\psi(x_i - x_m) = \delta_{ij} - \delta_{mj} \leq \delta_{im} \leq a$ holds $\Rightarrow \psi$ satisfies C

Example

Consider a set of inequalities:

$$x \geq u \quad y \leq 0 \quad u + 3 \geq x \quad x + 1 \leq y \quad u + 1 \geq 0 \quad v + 2 \geq 0$$



now try to add $v \leq u - 2$
 \Rightarrow creates negative cycle

More features: incremental verification

Lemma: if C is satisfiable then $C \wedge x_i - x_j \leq a$ is sat. $\Leftrightarrow \delta_{ji} + a \geq 0$

Proof:

\Rightarrow assume $\delta_{ji} + a < 0$, but $C \Rightarrow x_j - x_i \leq a$ and thus $C \wedge x_i - x_j \leq a \Rightarrow 0 \leq \delta_{ji} + a \Rightarrow$ contradiction

\Leftarrow assume $C \wedge x_i - x_j \leq a$ is unsat., then

$C \Rightarrow x_j - x_i > a \Rightarrow x_i - x_j < -a$
 but $\delta_{ji} = \max_{\psi \models C} \psi(x_j - x_i)$ thus $\delta_{ji} < -a \Rightarrow$ contradiction

An easy way to incrementally check satisfiability!

More features: incremental construction

Lemma: if C is sat. (δ) and $C \wedge x_i - x_j \leq a$ is sat. (δ') then $\delta'_{ki} = \min\{\delta_{ki}, \delta_{ki} + a + \delta_{ji}\}$

Proof:

Simple following the shortest-path interp. of δ_{ki}

Easy to recompute δ through incremental steps!

Difference Constraints: conclusions

- Can be solved with Bellman-Ford in $O(n^3)$
 - In practice n is typically quite small
 - In practice we use incremental algorithms (to account for assumptions being pushed and popped)
 - There are $O(n^2)$ algorithms as well
- Algorithm is complete in \mathbb{Z} !
 - given that all $c_i \in \mathbb{Z}$
 - outcome of modeling as accumulative paths in a graph
- Was used successfully in array-bounds checking elimination and induction variable discovery

Extensions of Difference Constraints

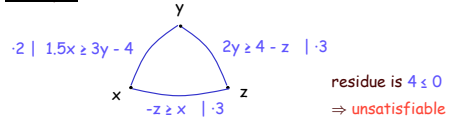
- Shostak extended the algorithm to $ax + by \leq c$
- Construct a graph as before
 - One node for each variable
 - One undirected edge for each constraint
- An admissible loop: any two adjacent edges, " $ax + by \leq c$ " and " $dy + ez \leq f$ ", have $\text{sgn}(b) \neq \text{sgn}(d)$
 - The residue of such adjacent edges is a constraint on x and z
 - Obtained by applying "transitivity" along the path:

$$\begin{array}{r|l} ax + by \leq c & | \cdot |d| \\ dy + ez \leq f & | \cdot |b| \\ \hline a|d|x + e|b|z \leq c|d| + f|b| \end{array}$$
 - Residue for a loop is an inequality without variables " $0 \leq n$ "

Difference Constraints: Shostak (2)

Claim: if a simple loop has a residue " $0 \leq n$ " with $n < 0$ then the set of constraints is unsat.

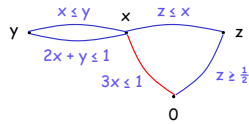
Example:



- However, the converse is not true...

Difference Constraints: Shostak (3)

Example: all admissible loops are feasible...



- But $x \leq y \wedge 2x + y \leq 1 \Rightarrow 3x \leq 1 \Rightarrow 3z \leq 1 \Rightarrow z \leq 1/3$
- Solution: augment graph with residues of all admissible loops
 - for this case add $3x \leq 1$ between x and 0
 - creates an infeasible loop

Difference Constraints: Shostak (4)

Theorem: inequalities are satisfiable iff all residues for simple loops (after augmentation) are feasible

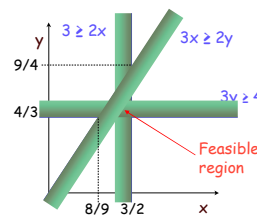
- Some algorithmic issues:
 - complexity induced by the finding of loops \Rightarrow possible exponential worst-case behavior
 - later extensions: avoid considering all cycles \Rightarrow polynomial worst-case time
- Further extensions allow more than two variables
 - pick an order of variables and determine two primary ("smaller") variables for each inequality
 - compute loop residues as before
 - can reduce polynomially to a set with 3 vars. per inequality

SUP-INF (Bledsoe '74, Shostak '77)

- A set of inequality constraints S with n variables forms a convex polyhedron in \mathbb{R}^n
- For each variable x compute
 - $SUP_x(x)$ the maximal value taken by x on all interpretations satisfying S
 - $INF_x(x)$ is the minimal value...
- We get that
 - $[INF(x), SUP(x)]$ is the projection on the x_k axis
 - If $[INF(x)] \times [SUP(x)]$ for some $x \Rightarrow$ no satisfying assignment!
- SUP/INF are computed recursively
 - Partition S into $x_1 \geq A_i(x_2, \dots, x_n) + c_i$ and $x_1 \leq B_j(x_2, \dots, x_n) + d_j$
 - Define: $SUP(x) = SUP(\min(B_j(x_2, \dots, x_n) + d_j))$
 $= \min_j(SUP(B_j(x_2, \dots, x_n) + d_j))$

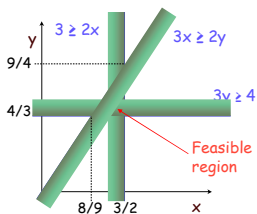
Example

Is this system satisfiable?



- $SUP(x)$
 - Only constraint that bounds x above is $3 \geq 2x$
 - Therefore $SUP(x) = 3/2$
- $INF(x)$
 - Only $3x \geq 2y$ bounds x below
 - $INF(x) = 2/3 y \geq 2/3 INF(y)$
 - Depends on $INF(y)$

Example (2)



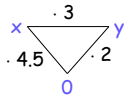
- $INF(y)$
 - Only $3y \geq 4$ bounds y below
 - $INF(y) = 4/3$
- $INF(x)$ reprise
 - Recall: $INF(x) \geq 2/3 INF(y)$
 - Thus $INF(x) \geq 8/9$
- $SUP(y)$
 - Only $3x \geq 2y$ bounds y below
 - $SUP(y) \geq 3/2 SUP(x) = 9/4$

Example (3)

- We get
 - $[INF(x)] = 1 \leq 1 = [SUP(x)] \Rightarrow x \in \{1\}$
 - $[INF(y)] = 2 \leq 2 = [SUP(y)] \Rightarrow y \in \{2\}$
- Seems satisfiable...
 - But it is not!
 - Not complete for \mathbb{Z}
- Shostak's improvement
 - If $[INF(x)] = [SUP(x)]$ replace x by $[INF(x)]$ and re-confront with constraints
 - In our case $3 \geq 2x$ instantiates to $3 \geq 2 \cdot 1 \Rightarrow$ no integer solution!
 - Does not guarantee completeness for \mathbb{Z} , but makes it less likely to hit the incompleteness

How Complete are These Procedures?

- Consider: $3x \geq 2y \wedge 3y \geq 4 \wedge 3 \geq 2x$



Residue is: $13.5 \geq 8 \Rightarrow$ satisfiable
But only in \mathbb{Q} , not in \mathbb{Z}

- The unsat procedure is sound: $\text{unsat } \mathbb{Q} \Rightarrow \text{unsat } \mathbb{Z}$
- But it is incomplete!
- Not a problem in practice
- Or the problem goes away with tricks like this:
Transform " $ax \geq b$ " into " $x \geq \lceil b/a \rceil$ "

Arithmetic. Discussion

- There are many satisfiability algorithms
 - Even for the general case (e.g. Simplex)
 - Except for difference constraints, all are incomplete in \mathbb{Z}
 - But \mathbb{Z} can be handled well with heuristics
- There are no practical satisfiability procedures for (\mathbb{Q}, \times) and the satisfiability of (\mathbb{Z}, \times) is only semi-decidable