Little Engines of Proof: Arithmetic Inequalities

N. Shankar, L. de Moura, H. Ruess, A. Tiwari

shankar@csl.sri.com

URL: http://www.csl.sri.com/~shankar/LEP.html

Computer Science Laboratory
SRI International
Menlo Park, CA

Overview

We have studied decision procedures for equality on constants, pure terms, interpreted terms, and combinations of these.

Next to equality, inequalities constitute the most common kind of constraint.

Inequalities have applications in scheduling, optimization, resource allocation, extended typechecking, compilers, constraint programming, timed/hybrid systems.

Solving arithmetic inequalities is a vast topic that has been heavily studied. We restrict ourselves to a few basic inference procedures for inequality.

Simple Orderings

We already saw that the uniform word problem for transitive and reflexive ordering relations could be captured by computing the transitive closure or the reflexive-transitive closure.

For example, $\models a < b, a < c, b < d, d < c \Rightarrow b < c$.

The following rules constitute an inference system for reflexive/transitive orderings.

$$\frac{\Gamma, a < b, b < c}{\Gamma, a < b, b < c, a < c} \text{ if } a < c \notin \Gamma$$

$$\frac{\Gamma, a < b, a \not< b}{\bot}$$

$$\frac{\Gamma, a \not< a}{\bot}$$

Partial Orders

If we add anti-symmetry to obtain partial orders, then we also have to detect cycles in order to obtain equalities.

Then in addition to the union-find rules for constants, we have

$$\frac{\Gamma, a \leq b, b \leq a}{\Gamma, a \leq b, b \leq a, a = b} \text{ if } a = b \notin \Gamma$$

$$\frac{\Gamma, a \leq b, b = c}{\Gamma, a \leq b, b = c, a \leq c} \text{ if } a \leq c \notin \Gamma$$

$$\frac{\Gamma, a \leq b, a = c}{\Gamma, a \leq b, a = c, c \leq b} \text{ if } c \leq b \notin \Gamma$$

Arithmetic Inequalities

Inequalities get quite a bit more interesting as an arithmetic ordering relation between numerical quantities.

These can be axiomatized by the *ordered field* axioms:

1. $+$ yields a commutative group with identity 0 and inverse $-x$.
2. $*$ yields a commutative group with identity 1, inverse x^{-1} for $x \neq 0$. $*$ distributes over $+$.
3. (a) $(x \not< x)$

   (b) $x < y \Rightarrow y < z \Rightarrow x < z$

   (c) $x < y \Rightarrow x + z < y + z$

   (d) $x, y > 0 \Rightarrow x * y > 0$

   (e) $x > 0 \vee x = 0 \vee 0 > x$

Linear Inequalities

Drop multiplication and allow constants ranging over rational numbers.

Ex: Show that all equalities have the form $c_n * x_n + \ldots c_1 * x_1 + c_0 = 0$, where c_i range over the rationals. Show that such an equation always admits rational solutions.

Ex: Show that all inequalities $a > b$ can be expressed as $c_n * x_n + \ldots c_1 * x_1 + c_0 > 0$, where c_i range over rationals.

$c_n * x_n + \ldots c_1 * x_1 + c_0$ is a linear polynomial.

A linear equality over one variable represents a rational magnitude.

A linear equality (inequality) in two variables is a line (half-space) in 2-space.

Interval Predicates

We will restrict ourselves to lax inequalities ($a \leq b$). The algorithms can be easily extended to strict inequalities ($a > b$).

The simplest case of inequalities in one variable: upper bounds $x \leq c$ and lower bounds $-x \leq c$.

Deciding this fragment is simply a matter of checking if for each variable x, the lower bound is below the upper bound.

$$\frac{\Gamma, x \leq c_1, x \leq c_2}{\Gamma, x \leq c_1} \text{ if } c_1 < c_2$$

$$\frac{\Gamma, x \leq c_1, -x \leq c_2}{\bot} \text{ if } c_1 + c_2 < 0$$

Separation Predicates

These are inequalities of the form $x - y \leq c$.

There is a special variable x_0 representing 0 so interval constraints are written as $x - x_0 \leq c$ or $x_0 - x \leq c$.

Difference Bounded Matrices (DBMs) are a popular representation for separation constraints with many applications.

For n variables, a DBM A is an $n \times n$ matrix such that A_{ij} is the best bound on the separation $x_i - x_j$.

Ex: If the constants are integers, then a conjunction of separation literals is satisfiable in the integers if it is satisfiable in the rationals.

Inference System for Separation Predicates [Pratt]

$\dfrac{\Gamma, x - y \leq c_1, y - x \leq c_2}{\bot}$	if $c_1 + c_2 < 0$

$\dfrac{(\Gamma' \equiv)\Gamma, x - y \leq c_1, y - z \leq c_2}{\Gamma', x - z \leq c_1 + c_2}$	$x \not\equiv z$
	for all $x - y \leq c_1' \in \Gamma, c_1' > c_1$
	for all $y - z \leq c_2' \in \Gamma, c_2' > c_2$

Ex: Prove correctness.

Ex: Show that the above system simulates the transitive closure computation on DBMs.

Shostak's Loop Residue Method

As before, the variable x_0 represents 0.

Two-variable inequality constraints have the form $a * x + b * y \leq c$, where $x \succ y$ in the variable ordering, or of the form $a * x_0 \leq c$, with $a \neq 0, b \neq 0$.

A one-variable inequality $a * x \leq c$ for $x \not\equiv x_0$ is expressed as $a * x + x_0 \leq c$.

$\dfrac{(\Gamma' \equiv)\Gamma, a * x + b * y \leq c, -a' * x + b' * z \leq c'}{\Gamma', a' * b * y + a * b' * z \leq a' * c + a * c'}$	if $x \succ y$, and $a, a' > 0$
$\dfrac{(\Gamma' \equiv)\Gamma, a * x + b * y \leq c, -a' * x + b' * y \leq c'}{\Gamma', (a' * b + a * b') * y + x_0 \leq a' * c + a * c'}$	if $a, a' > 0$
$\dfrac{\Gamma, a * x_0 \leq c}{\bot}$	if $c < 0$

Loop Residue Example

$$\begin{array}{c}
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -3 \\ \hline
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -3, y \leq 3 \\ \hline
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -3, y \leq 3, z \leq 0 \\ \hline
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -3, y \leq 3, z \leq 0, x_0 \leq -1 \\ \hline
\bot
\end{array}$$

$$\begin{array}{c}
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -2 \\ \hline
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -2, y \leq 3 \\ \hline
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -2, y \leq 3, z \leq 1 \\ \hline
x + y \leq 5, -x \leq -2, -z \leq -1, -y + z \leq -2, y \leq 3, z \leq 1, x_0 \leq 0 \\ \hline
x_0 = 0, z = 1, y = 3, x = 2
\end{array}$$

Correctness

Termination: Number of inference steps is bounded by $2^{|\Gamma|}$.

Model Preservation: Easy exercise.

Completeness: Construct $\mathcal{M}$ iteratively so that $\mathcal{M}_0(x_0) = 0$. Pick the minimum unassigned variable x_{i+1}. The constraints $a * x_{i+1} + b * y \leq c$ where x_{i+1} is maximal are evaluated under $\mathcal{M}_i$ to yield $a * x_{i+1} \leq c'$ in x_{i+1}. If $a < 0$, we get the lower bound $x_{i+1} \geq c'/a$. Otherwise, we get upper bound $x_{i+1} \leq c'/a$. Pick any arbitrary value $\mathcal{M}_{i+1}(x_{i+1})$ that satisfies all the bounds. If there is no satisfying choice, then there is an upper bound that is below the lower bound. Resolving the corresponding inequalities yields an inequality that should already be falsified in $\mathcal{M}_i$. A contradiction.

Extending Loop Residue

1. Ex: Extend the loop residue inference system to include strict inequalities.
2. Ex: Extend the loop residue inference system to allow arbitrary inequalities beyond the 2-variable case.
3. Ex: Extend the loop residue method to detect equalities.
4. Ex: Show an example of exponential behavior for loop residue.
5. Ex: Construct a polynomial algorithm for the 2-variable case.

Lemma Generation

When combining SAT solving with inequality solving, loop residue can be used to generate lemmas for the SAT solver.

Given a CNF formula ϕ, build a graph consisting of all the inequalities in ϕ.

Collect all the inequalities $A_1, \ldots, A_m$ in negative weight cycle and add a lemma clause L of the form $\neg A_1 \vee \ldots \vee \neg A_m$.

Let ϕ' be $\phi \wedge \bigwedge_i L_i$ for lemmas $L_1, \ldots, L_n$.

Replace each inequality atom A_i in ϕ' by a fresh propositional constant q_i to get $\overline{\phi'}$.

Check the satisfiability of $\overline{\phi'}$ with a SAT solver.

Fourier Elimination

Inequalities are represented in the form $x \leq u$ or $x \geq l$ where x is the maximal variable in the inequality.

$$\frac{(\Gamma' \equiv)\Gamma, x \geq l, x \leq u}{\Gamma', u \geq l}$$

$$\frac{\Gamma, x_0 \leq c}{\bot} \text{ if } c < 0$$

Ex: Prove correctness.

Ex: Given $m \times n$ matrix A and n-vector $\vec{b}$, prove that either $\exists \vec{x} : A\vec{x} \leq \vec{b} \vee \exists \vec{u} \geq 0 : \vec{u}^T A = 0, \vec{u}^T \vec{b} < 0$.

Ex: Show that any quantified formula whose atomic propositions are linear inequalities, can be reduced to a quantifier-free form.