Decision Procedures for Verification

Decision Procedures (3)

14.01.2019

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Fourier-Motzkin Quantifier Elimination

Linear rational arithmetic permits quantifier elimination:
every formula $\exists x F$ or $\forall x F$ in linear rational arithmetic can be converted into an equivalent formula without the variable x.

The method was discovered in 1826 by J. Fourier and re-discovered by T. Motzkin in 1936.

Observation: Every literal over the variables $x, y_1, \ldots, y_n$ can be converted into an ODAG-equivalent atom $x \sim t[\overline{y}]$ or $0 \sim t[\overline{y}]$, where $\sim \in \{<, >, \leq, \geq, \approx, \not\approx\}$ and $t[\overline{y}]$ has the form $\sum_i q_i \cdot y_i + q_0$.

In other words, we can either eliminate x completely or isolate it on one side of the atom.

Moreover, we can convert every $\not\approx$ atom into an ODAG-equivalent disjunction of two $<$ atoms.

Fourier-Motzkin Quantifier Elimination

We first consider existentially quantified conjunctions of atoms.

(1) **If the conjunction contains an equation $x \approx t[\overline{y}]$,**
we can eliminate the quantifier $\exists x$ by substitution:

$$\exists x(x \approx t[\overline{y}] \wedge F)$$

is equivalent to

$$F\sigma, \text{ where } \sigma = [t[\overline{y}]/x]$$

Fourier-Motzkin Quantifier Elimination

We first consider existentially quantified conjunctions of atoms.

(2) **If x occurs only in inequations, then:**

$$\begin{aligned} \exists x \quad & (\textstyle\bigwedge_i x < s_i(\overline{y}) \wedge \bigwedge_j x \leq t_j(\overline{y}) \wedge \\ & \ \textstyle\bigwedge_k x > u_k(\overline{y}) \wedge \bigwedge_l x \geq v_l(\overline{y}) \wedge \\ & F(\overline{y})) \end{aligned}$$

is equivalent to:

$$\begin{aligned} & \textstyle\bigwedge_i \bigwedge_k s_i(\overline{y}) > u_k(\overline{y}) \wedge \bigwedge_j \bigwedge_k t_j(\overline{y}) > u_k(\overline{y}) \wedge \\ & \textstyle\bigwedge_i \bigwedge_l s_i(\overline{y}) > v_l(\overline{y}) \wedge \bigwedge_j \bigwedge_l t_j(\overline{y}) \geq v_l(\overline{y}) \wedge \\ & F(\overline{y}) \end{aligned}$$

Proof: "$\Rightarrow$" follows by transitivity;

"$\Leftarrow$" Take $\frac{1}{2}(\min\{s_i, t_j\} + \max\{u_k, v_l\})$ as a witness.

Fourier-Motzkin Quantifier Elimination

Extension to arbitrary formulas:

- Transform into prenex formula;
- If innermost quantifier is $\exists$:
  transform matrix into DNF and move $\exists$ into disjunction;
- If innermost quantifier is $\forall$: replace $\forall x F$ by $\neg \exists x \neg F$, then eliminate $\exists$.

Consequences:

(1) Every closed formula over the signature of ODAGs is ODAG-equivalent to either $\top$ or $\bot$.

(2) ODAGs are a complete theory, i.e., every closed formula over the signature of ODAGs is either valid or unsatisfiable w.r.t. ODAGs.

(3) Every closed formula over the signature of ODAGs holds either in all ODAGs or in no ODAG.

ODAGs are indistinguishable by first-order formulas over the signature of ODAGs. (These properties do not hold for extended signatures!)

Fourier-Motzkin: Complexity

- **One FM-step for $\exists$:**
  formula size grows quadratically, therefore $O(n^2)$ runtime.
- m **quantifiers $\exists \dots \exists$:**
  naive implementation needs $O(n^{2^m})$ runtime;

  It is unknown whether optimized implementation with simply exponential runtime is possible.
- m **quantifiers $\exists\forall\exists\forall \dots \exists\forall$:**
  CNF/DNF conversion (exponential!) required after each step;
  therefore non-elementary runtime.

Fourier-Motzkin: Complexity

- **One FM-step for $\exists$:**

  formula size grows quadratically, therefore $O(n^2)$ runtime.

- **m quantifiers $\exists \ldots \exists$:**

  naive implementation needs $O(n^{2^m})$ runtime;

  It is unknown whether optimized implementation with simply exponential runtime is possible.

- **m quantifiers $\exists\forall\exists\forall \ldots \exists\forall$:**

  CNF/DNF conversion (exponential!) required after each step;
  therefore non-elementary runtime.

Improvement: Loos-Weispfenning Quantifier Elimination

Loos-Weispfenning Quantifier Elimination

A more efficient way to eliminate quantifiers in linear rational arithmetic was developed by R. Loos and V. Weispfenning (1993).

The method is also known as "test point method" or "virtual substitution method".

For simplicity, we consider only one particular ODAG, namely $\mathbb{Q}$ (as we have seen above, the results are the same for all ODAGs).

Loos-Weispfenning Quantifier Elimination

Let $F(x, \overline{y})$ be a positive boolean combination of linear (in-)equations of the form $\quad x \sim_i s_i(\overline{y}) \quad$ and $\quad 0 \sim_j s_j(\overline{y})$ with $\sim_i, \sim_j \in \{\approx, \not\approx, <, \leq, >, \geq\}$,
(i.e. a formula built from linear (in-) equations, $\vee$ and $\wedge$, but without $\neg$).

Goal: Find a finite set T of "test points" so that

$$\exists x F(x, \overline{y}) \models\!\mid \bigvee_{t \in T} F(x, \overline{y})[t/x].$$

In other words:
We want to replace the infinite disjunction $\exists x$ by a finite disjunction.

Loos-Weispfenning Quantifier Elimination

If we keep the values of the variables $\overline{y}$ fixed, we can regard F as a function

$$F : \mathbb{Q} \to \{0, 1\} \quad \text{defined by } x \mapsto F(x, \overline{y})$$

Remarks:

(1) The value of each of the atoms $s_i(\overline{y}) \sim_i x$ changes only at $s_i(\overline{y})$,
(2) The value of F can only change if the value of one of its atoms changes.

(3) F is a piecewise constant function; more precisely:
the set of all x with $F(x, \overline{y}) = 1$ is a finite union of intervals.

(The union may be empty, the individual intervals may be finite or infinite and open or closed.)

Let $\delta(\overline{y}) = \min\{|s_i(\overline{y}) - s_j(\overline{y})| \mid s_i(\overline{y}) \neq s_j(\overline{y})\}$.

Each of the intervals has either length 0 (i.e., it consists of one point), or its length is at least $\delta(\overline{y})$.

Loos-Weispfenning Quantifier Elimination

If the set of all x for which $F(x, \overline{y})$ is 1 is non-empty, then

(i) $F(x, \overline{y}) = 1$ for all $x \leq r(\overline{y})$ for some $r(\overline{y}) \in \mathbb{Q}$

(ii) or there is some point where the value of $F(x, \overline{y})$ switches from 0 to 1 when we traverse the real axis from $-\infty$ to $+\infty$.

We use this observation to construct a set of test points.

Loos-Weispfenning Quantifier Elimination

We start with a "sufficiently small" test point $r(\overline{y})$ to take care of case (i).

For case (ii), we observe that $F(x, \overline{y})$ can only switch from 0 to 1 if one of the atoms switches from 0 to 1. (We consider only positive boolean combinations of atoms and $\wedge$ and $\vee$ are monotonic w.r.t. truth values.)

- $x \leq s_i(\overline{y})$ and $x < s_i(\overline{y})$ do not switch from 0 to 1 when x grows.
- $x \geq s_i(\overline{y})$ and $x \approx s_i(\overline{y})$ switch from 0 to 1 at $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y})$ is a test point.
- $x > s_i(\overline{y})$ and $x \not\approx s_i(\overline{y})$ switch from 0 to 1 "right after" $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y}) + \epsilon$ (for some $0 < \epsilon < \delta(\overline{y})$) is a test point.

Loos-Weispfenning Quantifier Elimination

We start with a "sufficiently small" test point $r(\overline{y})$ to take care of case (i).

For case (ii), we observe that $F(x, \overline{y})$ can only switch from 0 to 1 if one of the atoms switches from 0 to 1. (We consider only positive boolean combinations of atoms and $\wedge$ and $\vee$ are monotonic w.r.t. truth values.)

- $x \leq s_i(\overline{y})$ and $x < s_i(\overline{y})$ do not switch from 0 to 1 when x grows.
- $x \geq s_i(\overline{y})$ and $x \approx s_i(\overline{y})$ switch from 0 to 1 at $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y})$ is a test point.
- $x > s_i(\overline{y})$ and $x \not\approx s_i(\overline{y})$ switch from 0 to 1 "right after" $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y}) + \epsilon$ (for some $0 < \epsilon < \delta(\overline{y})$) is a test point.

If $r(\overline{y})$ is sufficiently small and $0 < \epsilon < \delta(\overline{y})$, then

$$T := \{r(\overline{y})\} \cup \{s_i(\overline{y}) \mid \sim_i \in \{\geq, \approx\}\} \cup \{s_i(\overline{y}) + \epsilon \mid \sim_i \in \{>, \not\approx\}\}.$$

is a set of test points.

Loos-Weispfenning Quantifier Elimination

Problems:

(1) We don't know how small $r(\overline{y})$ has to be for case (i).

(2) We don't know $\delta(\overline{y})$ for case (ii).

Idea: We consider the limits for $r \to -\infty$ and for $\epsilon \to 0$ (but positive), that is, we redefine

$$T := \{-\infty\} \cup \{s_i(\overline{y}) \mid \sim_i \in \{\geq, \approx\}\} \cup \{s_i(\overline{y}) + \epsilon \mid \sim_i \in \{>, \not\approx\}\}.$$

New problem:

How can we eliminate the infinitesimals $-\infty$ and ϵ when we substitute elements of T for x?

Loos-Weispfenning Quantifier Elimination

Virtual substitution:

$$(x < s(\overline{y}))[-\infty/x] := \lim_{r\to -\infty}(r < s(\overline{y})) = \top$$

$$(x \leq s(\overline{y}))[-\infty/x] := \lim_{r\to -\infty}(r \leq s(\overline{y})) = \top$$

$$(x > s(\overline{y}))[-\infty/x] := \lim_{r\to -\infty}(r > s(\overline{y})) = \perp$$

$$(x \geq s(\overline{y}))[-\infty/x] := \lim_{r\to -\infty}(r \geq s(\overline{y})) = \perp$$

$$(x \approx s(\overline{y}))[-\infty/x] := \lim_{r\to -\infty}(r \approx s(\overline{y})) = \perp$$

$$(x \not\approx s(\overline{y}))[-\infty/x] := \lim_{r\to -\infty}(r \not\approx s(\overline{y})) = \top$$

Loos-Weispfenning Quantifier Elimination

Virtual substitution:

$$(x < s(\overline{y}))[u + \epsilon/x] := \lim_{\substack{\epsilon \to 0 \\ \epsilon > 0}} (u + \epsilon < s(\overline{y})) = (u < s(\overline{y}))$$

$$(x \leq s(\overline{y}))[u + \epsilon/x] := \lim_{\substack{\epsilon \to 0 \\ \epsilon > 0}} (u + \epsilon \leq s(\overline{y})) = (u < s(\overline{y}))$$

$$(x > s(\overline{y}))[u + \epsilon/x] := \lim_{\substack{\epsilon \to 0 \\ \epsilon > 0}} (u + \epsilon > s(\overline{y})) = (u \geq s(\overline{y}))$$

$$(x \geq s(\overline{y}))[u + \epsilon/x] := \lim_{\substack{\epsilon \to 0 \\ \epsilon > 0}} (u + \epsilon \geq s(\overline{y})) = (u \geq s(\overline{y}))$$

$$(x \approx s(\overline{y}))[u + \epsilon/x] := \lim_{\substack{\epsilon \to 0 \\ \epsilon > 0}} (u + \epsilon \approx s(\overline{y})) = \bot$$

$$(x \not\approx s(\overline{y}))[u + \epsilon/x] := \lim_{\substack{\epsilon \to 0 \\ \epsilon > 0}} (u + \epsilon \not\approx s(\overline{y})) = \top$$

We have traversed the real axis from $-\infty$ to $+\infty$.

Loos-Weispfenning Quantifier Elimination

Virtual substitution:

Alternatively, we can traverse it from $+\infty$ to $-\infty$.

In this case, the test points are

$$T' := \{+\infty\} \cup \{s_i(\overline{y}) \mid \sim_i \in \{\leq, \approx\}\} \cup \{s_i(\overline{y}) - \epsilon \mid \sim_i \in \{<, \not\approx\}\}.$$

Infinitesimals are eliminated in a similar way as before.

In practice: Compute both T and T' and take the smaller set.

For a universally quantified formula $\forall x F$, we replace it by $\neg \exists x \neg F$, push inner negation downwards, and then continue as before.

Note that there is no CNF/DNF transformation required.

Loos-Weispfenning quantifier elimination works on arbitrary positive formulas.

Loos-Weispfenning: Complexity

- **One LW-step for $\exists$ or $\forall$:**

  As the number of test points is at most equal to the number of atoms, the formula size grows quadratically; therefore $O(n^2)$ runtime.

- **Multiple quantifiers of the same kind:**

$$
\begin{array}{ll}
 & \exists x_2 \exists x_1 . F(x_1, x_2, \overline{y}) \\
\mapsto & \exists x_2 . \bigvee_{t_1 \in T_1} F(x_1, x_2, \overline{y})[t_1/x_1] \\
\mapsto & \bigvee_{t_1 \in T_1} (\exists x_2 . F(x_1, x_2, \overline{y})[t_1/x_1]) \\
\mapsto & \bigvee_{t_1 \in T_1} \bigvee_{t_2 \in T_2} F(x_1, x_2, \overline{y})[t_1/x_1][t_2/x_2]
\end{array}
$$

- ***m* quantifiers $\exists \ldots \exists$ or $\forall \ldots \forall$:**
  formula size is multiplied by n in each step $\Rightarrow O(n^{m+1})$ runtime.

- ***m* quantifiers $\exists\forall\exists\forall \ldots \forall$:** doubly exponential runtime.

Note: The formula resulting from a LW-step is usually highly redundant. An efficient implementation must make use of simplification techniques.

Until now

Decidable fragments of first-order logic

Decision procedures for single theories

- UIF
- Numeric domains

  Here:

  Difference logic

  Linear arithmetic over $\mathbb{R}$, $\mathbb{Q}$

Next: Reasoning in combinations of theories

Combinations of decision procedures