Towards Conflict-Driven Learning for Virtual Substitution

SMT Workshop

Marek Košta
(joint work K. Korovin and T. Sturm)

Max Planck Institute for Informatics
Saarbrücken, Germany

July 18, 2014

Motivation

Learning techniques turned out to be useful in different SMT contexts.

Concrete Examples

Algorithm	Learning Technique
DPLL	CDCL
Fourier-Motzkin	Bound Propagation, Conflict Resolution
CAD	model-based decision procedure
Virtual Substitution	this work

Our Goal

Take a virtual substitution-based method for a very special case (feasibility of linear programs over the reals) and devise a learning strategy.

Important Difference

- Fourier-Motzkin is doubly exponential (worst-case): All heuristic algorithms based on it will be doubly exponential as well.
- Virtual substitution is singly exponential.

Motivation

Learning techniques turned out to be useful in different SMT contexts.

Concrete Examples

Algorithm	Learning Technique
DPLL	CDCL
Fourier-Motzkin	Bound Propagation, Conflict Resolution
CAD	model-based decision procedure
Virtual Substitution	this work

Our Goal

Take a virtual substitution-based method for a very special case (feasibility of linear programs over the reals) and devise a learning strategy.

Important Difference

- Fourier-Motzkin is doubly exponential (worst-case): All heuristic algorithms based on it will be doubly exponential as well.
- Virtual substitution is singly exponential.

Motivation

Learning techniques turned out to be useful in different SMT contexts.

Concrete Examples

Algorithm	Learning Technique
DPLL	CDCL
Fourier-Motzkin	Bound Propagation, Conflict Resolution
CAD	model-based decision procedure
Virtual Substitution	this work

Our Goal

Take a virtual substitution-based method for a very special case (feasibility of linear programs over the reals) and devise a learning strategy.

Important Difference

- Fourier-Motzkin is doubly exponential (worst-case): All heuristic algorithms based on it will be doubly exponential as well.
- Virtual substitution is singly exponential.

Virtual Substitution

The Key Equivalence

$$\mathbb{R} \models \exists x \varphi \longleftrightarrow \bigvee_{e \in E(\varphi, x)} \varphi[x \leftarrow e],$$

where $E(\varphi, x)$ is a finite non-empty **elimination set** for formula φ and variable x, e is an **elimination term**, and $[x \leftarrow e]$ is a **virtual substitution**, which maps atomic formulas to quantifier-free formulas.

Ordinary vs. Virtual

Ordinary	Virtual
terms to terms	atomic formulas to quantifier-free formulas
$(x+y)/(x \leftarrow \frac{2}{5})$ yields $\frac{2+5y}{5}$	$(x+y \geqslant 0)[x \leftarrow \frac{2}{5}]$ yields $2+5y \geqslant 0$
$(x+y)/(x \leftarrow \infty)$ yields ???	$(x+y \geqslant 0)[x \leftarrow \infty]$ yields true

Virtual Substitution

The Key Equivalence

$$\mathbb{R} \models \exists x \varphi \longleftrightarrow \bigvee_{e \in E(\varphi, x)} \varphi[x \leftarrow e],$$

where $E(\varphi, x)$ is a finite non-empty **elimination set** for formula φ and variable x, e is an **elimination term**, and $[x \leftarrow e]$ is a **virtual substitution**, which maps atomic formulas to quantifier-free formulas.

Ordinary vs. Virtual

Ordinary	Virtual
terms to terms	atomic formulas to quantifier-free formulas
$(x+y)/(x \leftarrow \frac{2}{5})$ yields $\frac{2+5y}{5}$	$(x+y \geqslant 0)[x \leftarrow \frac{2}{5}]$ yields $2+5y \geqslant 0$
$(x+y)/(x \leftarrow \infty)$ yields ???	$(x+y \geqslant 0)[x \leftarrow \infty]$ yields true

An Example

Consider formula φ

$$x - 3y + 10 \geqslant 0 \wedge -2x - y + 5 \geqslant 0 \wedge x + y + 5 \geqslant 0 \wedge -x + y + 3 \geqslant 0.$$

One possible elimination set $E(\varphi, x)$ is:

$$\left\{ \frac{-y+5}{2}, y+3 \right\}.$$

Since $E(\varphi, x)$ is an elimination set, the key equivalence guarantees that

$$\begin{aligned} \exists x[\varphi] \quad \longleftrightarrow \quad & \varphi\left[x \leftarrow \frac{-y+5}{2}\right] \vee \varphi[x \leftarrow y+3] \\ & = (-7y + 25 \geqslant 0 \wedge 0 \geqslant 0 \wedge y + 15 \geqslant 0 \wedge 3y + 1 \geqslant 0) \vee \\ & \quad (-2y + 13 \geqslant 0 \wedge -3y - 1 \geqslant 0 \wedge 2y + 8 \geqslant 0 \wedge 0 \geqslant 0) \\ \longleftrightarrow \quad & y + 4 \geqslant 0 \wedge -7y + 25 \geqslant 0. \end{aligned}$$

The Main Idea

Input: a system of linear constraints $\varphi(x_1, \ldots, x_n) = l_1 \geqslant 0 \wedge \cdots \wedge l_n \geqslant 0$

Output: true iff the sentence $\exists x_n \ldots \exists x_1 \varphi(x_1, \ldots, x_n)$ is true in $\mathbb{R}$

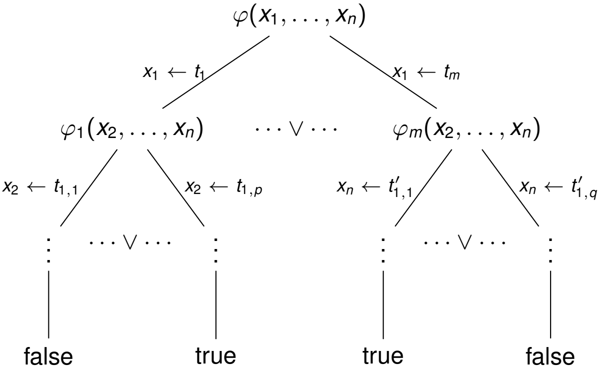
Simulate a virtual substitution-based QE algorithm by a calculus.

The Main Idea

Input: a system of linear constraints $\varphi(x_1, \ldots, x_n) = l_1 \geqslant 0 \wedge \cdots \wedge l_n \geqslant 0$

Output: true iff the sentence $\exists x_n \ldots \exists x_1 \varphi(x_1, \ldots, x_n)$ is true in $\mathbb{R}$

Simulate a virtual substitution-based QE algorithm by a calculus.



States of our Calculus

A state is a triple (F, S, L), where

$$F = l_1 \geqslant 0 \wedge \cdots \wedge l_n \geqslant 0$$

is a system of linear inequalities, S is a **stack**, and L is a **set of lemmas**.

The initial state is $(F, \langle\rangle, \emptyset)$.

Stack S

- is a list of virtual substitutions $\langle x_1 \leftarrow t_1\ (J_1), \ldots, x_n \leftarrow t_n\ (J_n)\rangle$. Here t_i is a formal solution of $J_i = 0$ with respect to x_i, where $J_i \geqslant 0$ is an input constraint.
- can contain $x_i \leftarrow ?$ or $x_i \leftarrow \bot$ at the end of it.
- can be substituted into a formula φ. This is denoted by φ/S.
- example: $S = \langle x_1 \leftarrow \frac{2x_2 - x_3}{47}, x_2 \leftarrow x_3 - 11, x_4 \leftarrow ?\rangle$

A lemma implied by F

- is a disjunction of negated equations of input constraints.
- example: $l_3 \neq 0 \vee l_5 \neq 0$, where $l_3 \geqslant 0$ and $l_5 \geqslant 0$ are input constraints.

States of our Calculus

A state is a triple (F, S, L), where

$$F = l_1 \geqslant 0 \wedge \cdots \wedge l_n \geqslant 0$$

is a system of linear inequalities, S is a **stack**, and L is a **set of lemmas**.

The initial state is $(F, \langle\rangle, \emptyset)$.

Stack S

- is a list of virtual substitutions $\langle x_1 \leftarrow t_1\ (J_1), \ldots, x_n \leftarrow t_n\ (J_n)\rangle$. Here t_i is a formal solution of $J_i = 0$ with respect to x_i, where $J_i \geqslant 0$ is an input constraint.
- can contain $x_i \leftarrow ?$ or $x_i \leftarrow \bot$ at the end of it.
- can be substituted into a formula φ. This is denoted by φ/S.
- example: $S = \langle x_1 \leftarrow \frac{2x_2 - x_3}{47}, x_2 \leftarrow x_3 - 11, x_4 \leftarrow ?\rangle$

A lemma implied by F

- is a disjunction of negated equations of input constraints.
- example: $l_3 \neq 0 \vee l_5 \neq 0$, where $l_3 \geqslant 0$ and $l_5 \geqslant 0$ are input constraints.

States of our Calculus

A state is a triple (F, S, L), where

$$F = I_1 \geqslant 0 \wedge \cdots \wedge I_n \geqslant 0$$

is a system of linear inequalities, S is a **stack**, and L is a **set of lemmas**.

The initial state is $(F, \langle\rangle, \emptyset)$.

Stack S

- is a list of virtual substitutions $\langle x_1 \leftarrow t_1\,(J_1), \ldots, x_n \leftarrow t_n\,(J_n)\rangle$. Here t_i is a formal solution of $J_i = 0$ with respect to x_i, where $J_i \geqslant 0$ is an input constraint.
- can contain $x_i \leftarrow ?$ or $x_i \leftarrow \bot$ at the end of it.
- can be substituted into a formula φ. This is denoted by φ / S.
- example: $S = \langle x_1 \leftarrow \frac{2x_2 - x_3}{47}, x_2 \leftarrow x_3 - 11, x_4 \leftarrow ?\rangle$

A lemma implied by F

- is a disjunction of negated equations of input constraints.
- example: $I_3 \neq 0 \vee I_5 \neq 0$, where $I_3 \geqslant 0$ and $I_5 \geqslant 0$ are input constraints.

Transition Rules (1)

DECIDE:

$$(F, S, L) \vdash (F, S \mid x_{k+1} \leftarrow ?, L)$$

where S does not contain "?" or "$\bot$"

if F/S is not trivially inconsistent, and $x_{k+1} \in \text{var}(F/S)$

SUBSTITUTE:

$$(F, S \mid x_k \leftarrow ?, L) \vdash (F, S \mid x_k \leftarrow \text{eterm}(F, S, L, x_k), L)$$

LEAF CONFLICT:

$$(F, S, L) \vdash (F, S, L \cup \{\bigvee_{i=1}^{k} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_k \leftarrow t_k\,(J_k)\rangle$, $k \geqslant 1$

if F/S is trivially inconsistent, and L/S is not trivially inconsistent

LEAF BACKTRACK:

$$(F, S \mid x_i \leftarrow t_i\,(J_i) \mid \ldots \mid x_k \leftarrow t_k\,(J_k), L) \vdash (F, S \mid x_i \leftarrow ?, L)$$

if $L/S \mid x_i \leftarrow t_i\,(J_i)$ is trivially inconsistent, and L/S is not trivially inconsistent

Transition Rules (1)

DECIDE:

$$(F, S, L) \;\vdash\; (F, S \,|\, x_{k+1} \leftarrow ?, L)$$

where S does not contain "?" or "$\perp$"

if F/S is not trivially inconsistent, and $x_{k+1} \in \text{var}(F/S)$

SUBSTITUTE:

$$(F, S \,|\, x_k \leftarrow ?, L) \;\vdash\; (F, S \,|\, x_k \leftarrow \text{eterm}(F, S, L, x_k), L)$$

LEAF CONFLICT:

$$(F, S, L) \;\vdash\; (F, S, L \cup \{\bigvee_{i=1}^{k} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_k \leftarrow t_k\,(J_k)\rangle$, $k \geqslant 1$

if F/S is trivially inconsistent, and L/S is not trivially inconsistent

LEAF BACKTRACK:

$$(F, S \,|\, x_i \leftarrow t_i\,(J_i) \,|\, \ldots \,|\, x_k \leftarrow t_k\,(J_k), L) \;\vdash\; (F, S \,|\, x_i \leftarrow ?, L)$$

if $L/S \,|\, x_i \leftarrow t_i\,(J_i)$ is trivially inconsistent, and L/S is not trivially inconsistent

Transition Rules (1)

DECIDE:

$$(F, S, L) \vdash (F, S \,|\, x_{k+1} \leftarrow ?, L)$$

where S does not contain "?" or "$\perp$"

if F/S is not trivially inconsistent, and $x_{k+1} \in \mathrm{var}(F/S)$

SUBSTITUTE:

$$(F, S \,|\, x_k \leftarrow ?, L) \vdash (F, S \,|\, x_k \leftarrow \mathrm{eterm}(F, S, L, x_k), L)$$

LEAF CONFLICT:

$$(F, S, L) \vdash \left(F, S, L \cup \left\{\bigvee_{i=1}^{k} J_i \neq 0\right\}\right)$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_k \leftarrow t_k\,(J_k)\rangle$, $k \geqslant 1$

if F/S is trivially inconsistent, and L/S is not trivially inconsistent

LEAF BACKTRACK:

$$(F, S \,|\, x_i \leftarrow t_i\,(J_i) \,|\, \ldots \,|\, x_k \leftarrow t_k\,(J_k), L) \vdash (F, S \,|\, x_i \leftarrow ?, L)$$

if $L/S \,|\, x_i \leftarrow t_i\,(J_i)$ is trivially inconsistent, and L/S is not trivially inconsistent

Transition Rules (1)

DECIDE:

$$(F, S, L) \vdash (F, S \mid x_{k+1} \leftarrow ?, L)$$

where S does not contain "?" or "$\bot$"

if F/S is not trivially inconsistent, and $x_{k+1} \in \mathrm{var}(F/S)$

SUBSTITUTE:

$$(F, S \mid x_k \leftarrow ?, L) \vdash (F, S \mid x_k \leftarrow \mathrm{eterm}(F, S, L, x_k), L)$$

LEAF CONFLICT:

$$(F, S, L) \vdash \left(F, S, L \cup \left\{\bigvee_{i=1}^{k} J_i \neq 0\right\}\right)$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_k \leftarrow t_k\,(J_k)\rangle$, $k \geqslant 1$

if F/S is trivially inconsistent, and L/S is not trivially inconsistent

LEAF BACKTRACK:

$$(F, S \mid x_i \leftarrow t_i\,(J_i) \mid \ldots \mid x_k \leftarrow t_k\,(J_k), L) \vdash (F, S \mid x_i \leftarrow ?, L)$$

if $L/S \mid x_i \leftarrow t_i\,(J_i)$ is trivially inconsistent, and L/S is not trivially inconsistent

Transition Rules (2)

INNER CONFLICT:

$$(F, S \,|\, x_k \leftarrow \bot, L) \;\vdash\; (F, S \,|\, x_k \leftarrow \bot, L \cup \{\textstyle\bigvee_{i=1}^{k-1} J_i \neq 0\})$$
$$\text{where } S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})\rangle$$

if L/S is not trivially inconsistent

INNER BACKTRACK:

$$(F, S \,|\, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1}) \,|\, x_k \leftarrow \bot, L) \;\vdash\; (F, S \,|\, x_{k-1} \leftarrow ?, L)$$

if $L/S \,|\, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})$ is trivially inconsistent

FAIL:

$$(F, \langle x_1 \leftarrow \bot\rangle, L) \;\vdash\; \bot$$

SUCCEED:

$$(F, S, L) \;\vdash\; \top$$

if $\mathrm{var}(F/S) = \emptyset$, and F/S is equivalent to true

Transition Rules (2)

INNER CONFLICT:

$$(F, S \,|\, x_k \leftarrow \bot, L) \;\vdash\; (F, S \,|\, x_k \leftarrow \bot, L \cup \{\bigvee_{i=1}^{k-1} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})\rangle$

if L/S is not trivially inconsistent

INNER BACKTRACK:

$$(F, S \,|\, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1}) \,|\, x_k \leftarrow \bot, L) \;\vdash\; (F, S \,|\, x_{k-1} \leftarrow ?, L)$$

if $L/S \,|\, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})$ is trivially inconsistent

FAIL:

$$(F, \langle x_1 \leftarrow \bot\rangle, L) \;\vdash\; \bot$$

SUCCEED:

$$(F, S, L) \;\vdash\; \top$$

if $\mathrm{var}(F/S) = \emptyset$, and F/S is equivalent to true

Transition Rules (2)

INNER CONFLICT:

$$(F, S \mid x_k \leftarrow \bot, L) \vdash (F, S \mid x_k \leftarrow \bot, L \cup \{\bigvee_{i=1}^{k-1} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})\rangle$

if L/S is not trivially inconsistent

INNER BACKTRACK:

$$(F, S \mid x_{k-1} \leftarrow t_{k-1}\,(J_{k-1}) \mid x_k \leftarrow \bot, L) \vdash (F, S \mid x_{k-1} \leftarrow ?, L)$$

if $L/S \mid x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})$ is trivially inconsistent

FAIL:

$$(F, \langle x_1 \leftarrow \bot\rangle, L) \vdash \bot$$

SUCCEED:

$$(F, S, L) \vdash \top$$

if $\mathrm{var}(F/S) = \emptyset$, and F/S is equivalent to true

Transition Rules (2)

INNER CONFLICT:

$$(F, S \,|\, x_k \leftarrow \bot, L) \;\vdash\; (F, S \,|\, x_k \leftarrow \bot, L \cup \{\textstyle\bigvee_{i=1}^{k-1} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})\rangle$

if L/S is not trivially inconsistent

INNER BACKTRACK:

$$(F, S \,|\, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1}) \,|\, x_k \leftarrow \bot, L) \;\vdash\; (F, S \,|\, x_{k-1} \leftarrow ?, L)$$

if $L/S \,|\, x_{k-1} \leftarrow t_{k-1}\,(J_{k-1})$ is trivially inconsistent

FAIL:

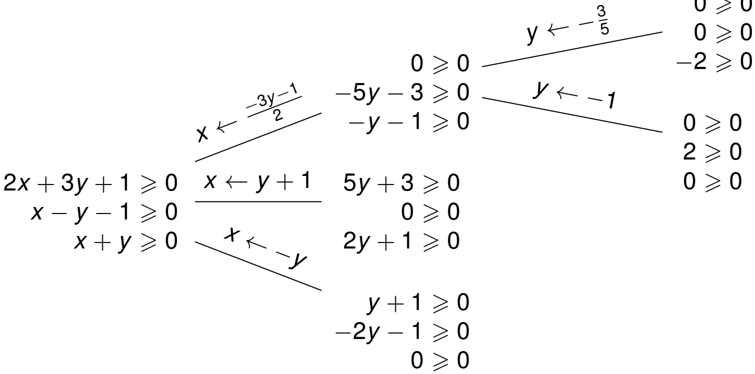
$$(F, \langle x_1 \leftarrow \bot\rangle, L) \;\vdash\; \bot$$

SUCCEED:

$$(F, S, L) \;\vdash\; \top$$

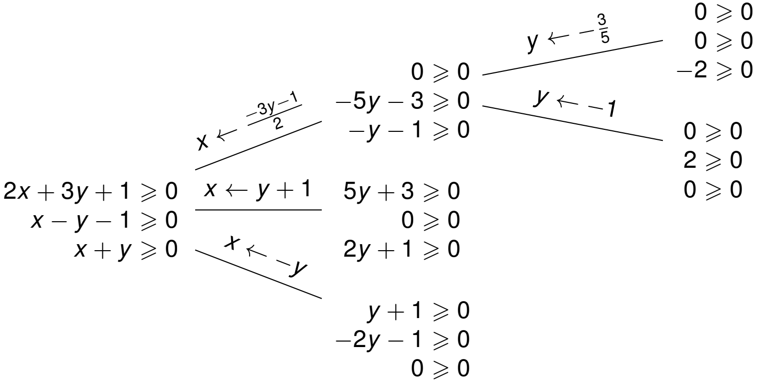
if $\mathrm{var}(F/S) = \emptyset$, and F/S is equivalent to true

An Example



$$(F, \langle\rangle, \emptyset)$$

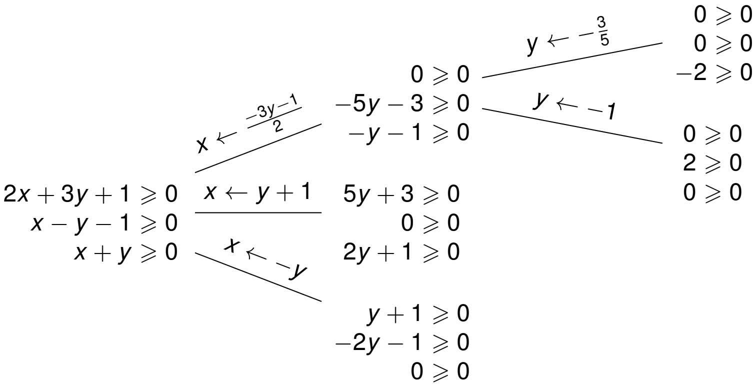
An Example



$$(F, \langle x \leftarrow ?\rangle, \emptyset)$$

by DECIDE

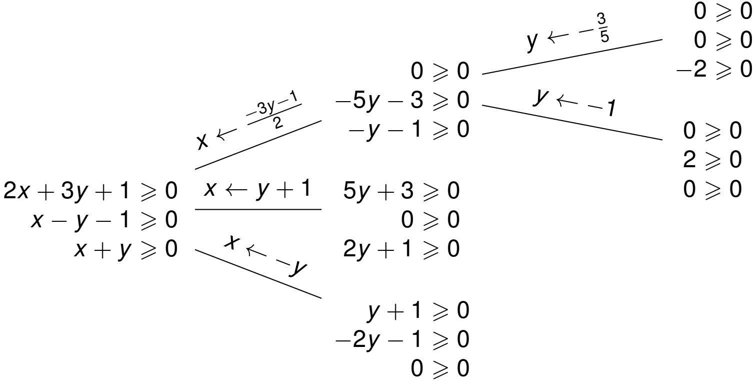
An Example



$$(F, \langle x \leftarrow \frac{-3y-1}{2} \rangle, \emptyset)$$

by SUBSTITUTE

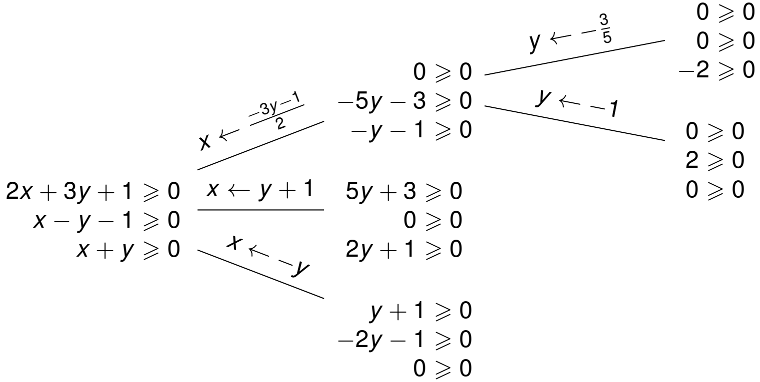
An Example



$$(F, \langle x \leftarrow \tfrac{-3y-1}{2}, y \leftarrow ?\rangle, \emptyset)$$

by DECIDE

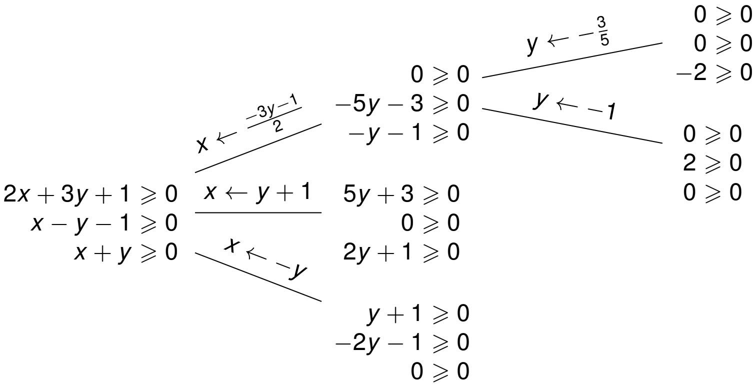
An Example



$$(F, \langle x \leftarrow \frac{-3y-1}{2}, y \leftarrow -\frac{3}{5} \rangle, \emptyset)$$

by SUBSTITUTE

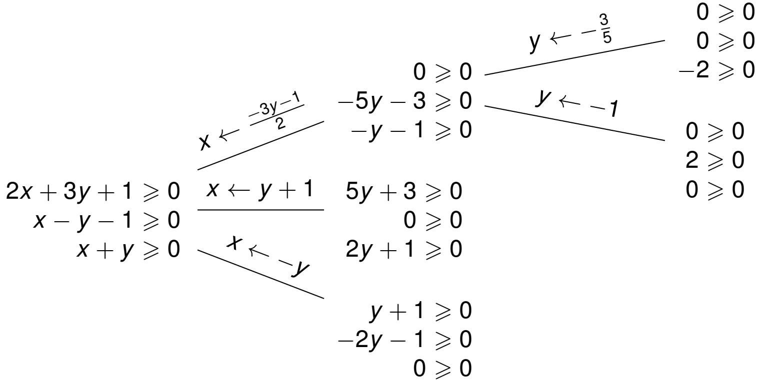
An Example



$$(F, \langle x \leftarrow \frac{-3y-1}{2}, y \leftarrow -\frac{3}{5} \rangle, \{2x + 3y + 1 \neq 0 \vee x - y - 1 \neq 0\})$$

by LEAF CONFLICT

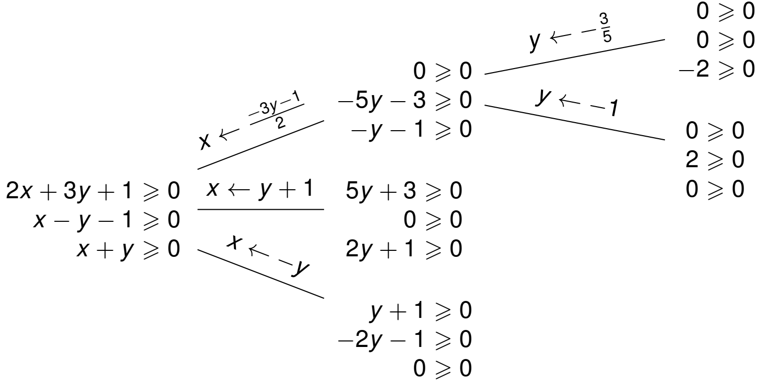
An Example



$$(F, \langle x \leftarrow \frac{-3y-1}{2}, y \leftarrow ?\rangle, \{2x + 3y + 1 \neq 0 \lor x - y - 1 \neq 0\})$$

by LEAF BACKTRACK

An Example



$$(F, \langle x \leftarrow \frac{-3y-1}{2}, y \leftarrow -1\rangle, \{2x + 3y + 1 \neq 0 \vee x - y - 1 \neq 0\})$$

by SUBSTITUTE

An Example

$$\begin{array}{r} 2x+3y+1 \geqslant 0 \\ x-y-1 \geqslant 0 \\ x+y \geqslant 0 \end{array}$$

$x \leftarrow \frac{-3y-1}{2}$:

$$\begin{array}{r} 0 \geqslant 0 \\ -5y-3 \geqslant 0 \\ -y-1 \geqslant 0 \end{array}$$

- $y \leftarrow -\frac{3}{5}$:

$$\begin{array}{r} 0 \geqslant 0 \\ 0 \geqslant 0 \\ -2 \geqslant 0 \end{array}$$

- $y \leftarrow -1$:

$$\begin{array}{r} 0 \geqslant 0 \\ 2 \geqslant 0 \\ 0 \geqslant 0 \end{array}$$

$x \leftarrow y+1$:

$$\begin{array}{r} 5y+3 \geqslant 0 \\ 0 \geqslant 0 \\ 2y+1 \geqslant 0 \end{array}$$

$x \leftarrow -y$:

$$\begin{array}{r} y+1 \geqslant 0 \\ -2y-1 \geqslant 0 \\ 0 \geqslant 0 \end{array}$$

$\top$

by SUCCEED

The Main Theoretical Results

For the first time, virtual substitution was formalized as a calculus.

Lemma (determinism)

For every reachable state there is exactly one rule applicable. The only non-determinism in the calculus is in the selection of the next variable.

Theorem (soundness and completeness)

Given a system F of linear inequalities, every derivation beginning in the initial state $(F, \langle\rangle, \emptyset)$ terminates either in state $\bot$ or in state $\top$. In the first case the system is unsatisfiable, in the latter case it is satisfiable.

Theorem (complexity)

There is a singly exponential upper bound on the number of derivation steps.

From a practical point of view, every reasonable implementation of the calculus is a systematic depth-first search.

The Main Theoretical Results

For the first time, virtual substitution was formalized as a calculus.

Lemma (determinism)

For every reachable state there is exactly one rule applicable. The only non-determinism in the calculus is in the selection of the next variable.

Theorem (soundness and completeness)

Given a system F of linear inequalities, every derivation beginning in the initial state $(F, \langle\rangle, \emptyset)$ terminates either in state $\bot$ or in state $\top$. In the first case the system is unsatisfiable, in the latter case it is satisfiable.

Theorem (complexity)

There is a singly exponential upper bound on the number of derivation steps.

From a practical point of view, every reasonable implementation of the calculus is a systematic depth-first search.

The Main Theoretical Results

For the first time, virtual substitution was formalized as a calculus.

Lemma (determinism)

For every reachable state there is exactly one rule applicable. The only non-determinism in the calculus is in the selection of the next variable.

Theorem (soundness and completeness)

Given a system F of linear inequalities, every derivation beginning in the initial state $(F, \langle\rangle, \emptyset)$ terminates either in state $\bot$ or in state $\top$. In the first case the system is unsatisfiable, in the latter case it is satisfiable.

Theorem (complexity)

There is a singly exponential upper bound on the number of derivation steps.

From a practical point of view, every reasonable implementation of the calculus is a systematic depth-first search.

The Main Theoretical Results

For the first time, virtual substitution was formalized as a calculus.

Lemma (determinism)

For every reachable state there is exactly one rule applicable. The only non-determinism in the calculus is in the selection of the next variable.

Theorem (soundness and completeness)

Given a system F of linear inequalities, every derivation beginning in the initial state $(F, \langle\rangle, \emptyset)$ terminates either in state $\bot$ or in state $\top$. In the first case the system is unsatisfiable, in the latter case it is satisfiable.

Theorem (complexity)

There is a singly exponential upper bound on the number of derivation steps.

From a practical point of view, every reasonable implementation of the calculus is a systematic depth-first search.

Enhancing the Basic Calculus

- The idea is to learn a stronger, i.e., a shorter lemma, which triggers backtrack to a further node.
- To achieve this, we analyze a conflicting situation.

A conflicting situation

- is a state (F, S, L), where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_n \leftarrow t_n\,(J_n)\rangle$, and
- there is a **conflicting inequality** $(K \geqslant 0) \in F$, s.t. $(K \geqslant 0)/S$ yields $\gamma \geqslant 0$, where $0 > \gamma \in \mathbb{Q}$.

In a conflicting situation, the basic calculus learns a lemma

$$B : J_1 \neq 0 \vee \cdots \vee J_n \neq 0.$$

Learning a shorter lemma

$$E : J_{i_1} \neq 0 \vee \cdots \vee J_{i_k} \neq 0,$$

where $(J_{i_j} \neq 0) \in B$, for all $j \in \{1, \ldots, k\}$, would trigger a further backtrack.

Enhancing the Basic Calculus

- The idea is to learn a stronger, i.e., a shorter lemma, which triggers backtrack to a further node.
- To achieve this, we analyze a conflicting situation.

A conflicting situation

- is a state (F, S, L), where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_n \leftarrow t_n\,(J_n)\rangle$, and
- there is a **conflicting inequality** $(K \geqslant 0) \in F$, s.t. $(K \geqslant 0)/S$ yields $\gamma \geqslant 0$, where $0 > \gamma \in \mathbb{Q}$.

In a conflicting situation, the basic calculus learns a lemma

$$B : J_1 \neq 0 \vee \cdots \vee J_n \neq 0.$$

Learning a shorter lemma

$$E : J_{i_1} \neq 0 \vee \cdots \vee J_{i_k} \neq 0,$$

where $(J_{i_j} \neq 0) \in B$, for all $j \in \{1, \ldots, k\}$, would trigger a further backtrack.

Enhancing the Basic Calculus

- The idea is to learn a stronger, i.e., a shorter lemma, which triggers backtrack to a further node.
- To achieve this, we analyze a conflicting situation.

A conflicting situation

- is a state (F, S, L), where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_n \leftarrow t_n\,(J_n)\rangle$, and
- there is a **conflicting inequality** $(K \geqslant 0) \in F$, s.t. $(K \geqslant 0)/S$ yields $\gamma \geqslant 0$, where $0 > \gamma \in \mathbb{Q}$.

In a conflicting situation, the basic calculus learns a lemma

$$B : J_1 \neq 0 \vee \cdots \vee J_n \neq 0.$$

Learning a shorter lemma

$$E : J_{i_1} \neq 0 \vee \cdots \vee J_{i_k} \neq 0,$$

where $(J_{i_j} \neq 0) \in B$, for all $j \in \{1, \ldots, k\}$, would trigger a further backtrack.

Linear Combination

The Key Observation

Result of a substitution is in fact a linear combination of the involved constraints.

An Example

Consider the stack from our previous example:

$$S = \left\langle x \leftarrow \frac{-3y-1}{2}\,(J_1),\, y \leftarrow -\frac{3}{5}\,(J_2) \right\rangle,$$

where $J_1 \geqslant 0$ is $2x + 3y + 1 \geqslant 0$, and $J_2 \geqslant 0$ is $x - y - 1 \geqslant 0$. Then

$$(x + y \geqslant 0)/S \quad \text{yields} \quad -2 \geqslant 0.$$

Given a conflicting inequality $K \geqslant 0$, the left-hand side of K/S can be expressed as a linear combination of $J_i \in S$ and K. In our example we have:

$$10\left(-\frac{2}{5}(2x + 3y + 1) - \frac{1}{5}(x - y - 1)\right) + 10(x + y) = -2.$$

Linear Combination

The Key Observation

Result of a substitution is in fact a linear combination of the involved constraints.

An Example

Consider the stack from our previous example:

$$S \quad = \quad \left\langle x \leftarrow \frac{-3y-1}{2}\,(J_1), y \leftarrow -\frac{3}{5}\,(J_2) \right\rangle,$$

where $J_1 \geqslant 0$ is $2x + 3y + 1 \geqslant 0$, and $J_2 \geqslant 0$ is $x - y - 1 \geqslant 0$. Then

$$(x + y \geqslant 0)/S \quad \text{yields} \quad -2 \geqslant 0.$$

Given a conflicting inequality $K \geqslant 0$, the left-hand side of K/S can be expressed as a linear combination of $J_i \in S$ and K. In our example we have:

$$10\left(-\frac{2}{5}(2x+3y+1) - \frac{1}{5}(x-y-1)\right) + 10(x+y) = -2.$$

Linear Combination

The Key Observation

Result of a substitution is in fact a linear combination of the involved constraints.

An Example

Consider the stack from our previous example:

$$S = \left\langle x \leftarrow \frac{-3y-1}{2}\,(J_1), y \leftarrow -\frac{3}{5}\,(J_2) \right\rangle,$$

where $J_1 \geqslant 0$ is $2x + 3y + 1 \geqslant 0$, and $J_2 \geqslant 0$ is $x - y - 1 \geqslant 0$. Then

$$(x + y \geqslant 0)/S \quad \text{yields} \quad -2 \geqslant 0.$$

Given a conflicting inequality $K \geqslant 0$, the left-hand side of K/S can be expressed as a linear combination of $J_i \in S$ and K. In our example we have:

$$10\left(-\frac{2}{5}(2x+3y+1) - \frac{1}{5}(x-y-1)\right) + 10(x+y) = -2.$$

Linear Combination

The Key Observation

Result of a substitution is in fact a linear combination of the involved constraints.

An Example

Consider the stack from our previous example:

$$S = \left\langle x \leftarrow \frac{-3y-1}{2}\,(J_1), y \leftarrow -\frac{3}{5}\,(J_2) \right\rangle,$$

where $J_1 \geqslant 0$ is $2x + 3y + 1 \geqslant 0$, and $J_2 \geqslant 0$ is $x - y - 1 \geqslant 0$. Then

$$(x + y \geqslant 0)/S \quad \text{yields} \quad -2 \geqslant 0.$$

Given a conflicting inequality $K \geqslant 0$, the left-hand side of K/S can be expressed as a linear combination of $J_i \in S$ and K. In our example we have:

$$10\left(-\frac{2}{5}(2x+3y+1) - \frac{1}{5}(x-y-1)\right) + 10(x+y) = -2.$$

Conflict Analysis

Lemma

Consider a stack $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_k \leftarrow t_k\,(J_k)\rangle$ and a linear term K. Then one can compute $\alpha_1, \ldots, \alpha_k \in \mathbb{Q}$, $b \in \mathbb{N} \setminus \{0\}$ such that

$$K/S = b \sum_{i=1}^{k} \alpha_i J_i + bK.$$

Lemma (Conflict Analysis)

Consider a conflicting situation (F, S, L), where $S = \langle x_1 \leftarrow t_1\,(J_1), \ldots, x_n \leftarrow t_n\,(J_n)\rangle$, and there is a conflicting inequality $(K \geqslant 0) \in F$. Compute $\alpha_1, \ldots, \alpha_k$ from the previous lemma. Then it is sufficient to take those J_i, where $\alpha_i < 0$, i.e.,

$$\mathbb{R} \models F \longrightarrow \bigvee_{\substack{i=1 \\ \alpha_i < 0}}^{k} J_i \neq 0.$$

Conflict Analysis

Lemma

Consider a stack $S = \langle x_1 \leftarrow t_1(J_1), \ldots, x_k \leftarrow t_k(J_k) \rangle$ and a linear term K. Then one can compute $\alpha_1, \ldots, \alpha_k \in \mathbb{Q}$, $b \in \mathbb{N} \setminus \{0\}$ such that

$$K/S = b \sum_{i=1}^{k} \alpha_i J_i + bK.$$

Lemma (Conflict Analysis)

Consider a conflicting situation (F, S, L), where $S = \langle x_1 \leftarrow t_1(J_1), \ldots, x_n \leftarrow t_n(J_n) \rangle$, and there is a conflicting inequality $(K \geqslant 0) \in F$. Compute $\alpha_1, \ldots, \alpha_k$ from the previous lemma. Then it is sufficient to take those J_i, where $\alpha_i < 0$, i.e.,

$$\mathbb{R} \models F \longrightarrow \bigvee_{\substack{i=1 \\ \alpha_i < 0}}^{k} J_i \neq 0.$$

An Enhanced Calculus

Define a function lincomb$(S, K \geqslant 0)$, which computes the coefficients α_i from the Conflict Analysis Lemma.

To get the enhanced calculus, replace the LEAF CONFLICT rule by

ANALYZE CONFLICT :

$$(F, S, L) \vdash (F, S, L \cup \{\bigvee_{i=1, \alpha_i < 0}^{k} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\ (J_1), \ldots, x_k \leftarrow t_k\ (J_k)\rangle$

and $(\alpha_1, \ldots, \alpha_k) = \text{lincomb}(S, K \geqslant 0)$

if F contains a conflicting inequality $K \geqslant 0$ with respect to S.

Properties of the Enhanced Calculus

All our results on the basic calculus carry over to the enhanced calculus.

An Enhanced Calculus

Define a function lincomb($S, K \geqslant 0$), which computes the coefficients α_i from the Conflict Analysis Lemma.

To get the enhanced calculus, replace the LEAF CONFLICT rule by

ANALYZE CONFLICT :

$$(F, S, L) \ \vdash \ (F, S, L \cup \{\bigvee_{i=1,\alpha_i<0}^{k} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\ (J_1), \ldots, x_k \leftarrow t_k\ (J_k)\rangle$
and $(\alpha_1, \ldots, \alpha_k) = \text{lincomb}(S, K \geqslant 0)$

if F contains a conflicting inequality $K \geqslant 0$ with respect to S.

Properties of the Enhanced Calculus

All our results on the basic calculus carry over to the enhanced calculus.

An Enhanced Calculus

Define a function lincomb($S, K \geqslant 0$), which computes the coefficients α_i from the Conflict Analysis Lemma.

To get the enhanced calculus, replace the LEAF CONFLICT rule by

ANALYZE CONFLICT :

$$(F, S, L) \ \vdash \ (F, S, L \cup \{\bigvee_{i=1,\alpha_i<0}^{k} J_i \neq 0\})$$

where $S = \langle x_1 \leftarrow t_1\ (J_1), \ldots, x_k \leftarrow t_k\ (J_k)\rangle$

and $(\alpha_1, \ldots, \alpha_k) = \mathrm{lincomb}(S, K \geqslant 0)$

if F contains a conflicting inequality $K \geqslant 0$ with respect to S.

Properties of the Enhanced Calculus

All our results on the basic calculus carry over to the enhanced calculus.

Implementation & Experiments (1)

- We implemented both calculi in Redlog, which is part of the open-source computer algebra system REDUCE.
- The implementation uses only precise arithmetic, no floats.
- Our aim is NOT to compete with existing specialized and highly efficient tools for linear programming.
- Interesting property: The implementation maps almost one-to-one to the presented formalism, because of the determinism of both calculi.

Implementation & Experiments (2)

number of explored QE-tree nodes by our calculi and by the Redlog's virtual substitution-based quantifier elimination algorithm: The benchmarks were taken from Netlib and formulated as decision problems.

	original		
	basic	enhanced	rlqe
afiro	333,355	136	267
blend	83	83	1,592
kb2	43	43	6,965
sc50a	86	70	7,535
sc50b	48	48	1,561
sc105	n/a	4,450	1,785,226

	feasible			infeasible		
	basic	enhanced	rlqe	basic	enhanced	rlqe
afiro	909,315	183	546	34,382,742	183	574
blend	n/a	n/a	n/a	n/a	n/a	n/a
kb2	n/a	n/a	n/a	n/a	n/a	n/a
sc50a	166,894	600	24,700	15,064,009	568	56,668
sc50b	49	49	602	216,952	49	2,223
sc105	n/a	6,701	n/a	n/a	5,427	n/a

Future Work

1. all predicates, not just $\geqslant$
2. Our implementation already uses $\pm\infty$ internally, add support for $\pm\varepsilon$.
3. Can we extract proofs for unsatisfiable instances?
4. arbitrary Boolean structure
5. What can we do about parameters?
6. Use the calculus point-of-view for the non-linear case, and develop a learning strategy for non-linear virtual substitution.

Summary

1. A brief introduction to virtual substitution.
2. We presented a basic calculus based on virtual substitution.
3. The main results concerning the calculus are: determinism, soundness, and completeness.
4. The main idea of the enhanced calculus is to learn stronger lemmas.
5. To learn stronger lemmas, we use linear algebra techniques.
6. We implemented both calculi in Redlog and presented experiments conducted with our implementation.