

Interprocedurally Analyzing Linear Inequality Relations

Andrea Flexeder, Michael Petter, Helmut Seidl



TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK



Nordic workshop on programming theory 2006

Research group



Helmut Seidl



Andrea Flexeder



Michael Petter

Center of interest: program analysis by abstract interpretation

⇒ Focus on efficient interprocedural analysis of numerical properties:

$$\begin{array}{r|l} 2x^2 + xy - 5 = 0 & x^3 - 5xy^2 + 1 \geq 0 \\ \uparrow & \uparrow \\ x - 2y + 5 = 0 & 3x + y + 1 \geq 0 \\ \uparrow & \uparrow \\ x - 5 = 0 & y + 1 \geq 0 \end{array}$$

Research group



Helmut Seidl



Andrea Flexeder



Michael Petter

Center of interest: program analysis by abstract interpretation

⇒ Focus on efficient interprocedural analysis of numerical properties:

$$\begin{array}{l|l} 2x^2 + xy - 5 = 0 & x^3 - 5xy^2 + 1 \geq 0 \\ \quad \quad \quad \uparrow & \quad \quad \quad \uparrow \\ x - 2y + 5 = 0 & 3x + y + 1 \geq 0 \\ \quad \quad \quad \uparrow & \quad \quad \quad \uparrow \\ x - 5 = 0 & y + 1 \geq 0 \end{array}$$

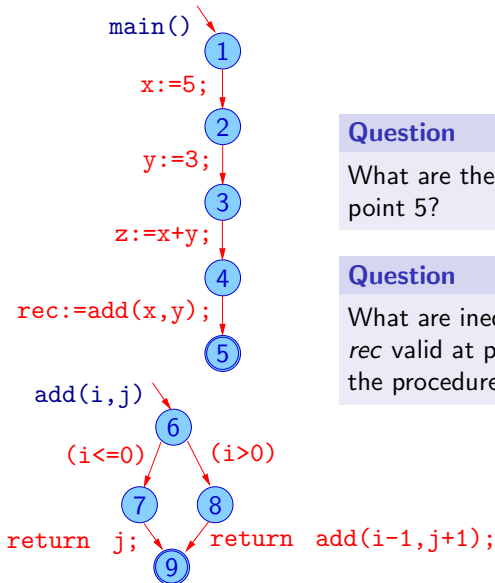
Outline

- **Problem:** Interprocedural analysis of linear inequalities
- 5 **Ideas** to implement an analysis
 - 1 Convex abstraction
 - 2 Abstraction of effects
 - 3 Guard handling
 - 4 Polyhedral analysis
 - 5 Simplicial analysis
- Experimental **Results** and **Conclusion**





Problem: Interprocedural Linear Inequalities



Question

What are the inequalities, valid at program point 5?

Question

What are inequality relations between `z` and `rec` valid at program point 5, considering the procedure call `add(x,y)`?

⇒ *Linear inequalities*

Background

Intraprocedural

Karr	Affine Relationships Among Variables of a Program	1976
Halbwachs, Cousot	Automatic Discovery of Linear Restraints among Variables of a Program	1978
Miné	The Octagon abstract domain	2001
Simon, King, Howe	Two Variables per Linear Inequality as an Abstract Domain	2002

Interprocedural

Seidl, Müller-Olm	Program Analysis through Linear Algebra	2004
Seidl, Müller-Olm	A Generic Framework for Interprocedural Analysis of Numerical Properties	2005
Seidl, M.-O., Petter	Interprocedurally Analyzing Polynomial Identities	2006
Seidl, Flexeder, P.	Interprocedurally Analyzing Linear Inequality Relations	2007

Idea 1: Convex abstraction

$$X \subseteq \{1\} \times \mathbb{F}^n \quad \text{i.e.} \quad (1, \mathbf{x}, \mathbf{y}, \mathbf{z}, \text{rec})^T$$

$$\langle X \rangle = \left\{ \sum_{i=1}^n \lambda_i x_i \mid n \in \mathbb{N} \wedge 0 \leq \lambda_i \wedge \sum_{i=1}^n \lambda_i = 1 \wedge x_i \in X \right\}$$

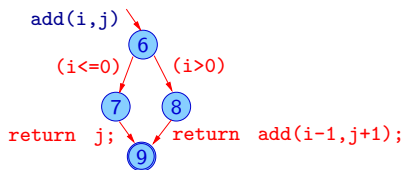
Abstract interpretation system

- linear inequality relations \equiv half spaces
- conjunctive combination of half spaces \equiv convex sets
- program state as vector of values: $(1, 5, 3, 8, 8)^T$
- assignments as linear transformations
- guarded statements as intersections of convex sets
- inspired by intraprocedural analysis by *Cousot & Halbwachs*

Open problem: procedure calls

- standard approach with transition invariants
- information loss after procedure call/guards

Idea 2: Convex abstraction of effects



For $(1, i, j, \text{ret})^T$:
Convex set of effects of `add()`:

$$\left\{ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ -\lambda & 1 & 0 & 0 \\ \lambda & 0 & 1 & 0 \\ \lambda & 0 & 1 & 0 \end{array} \right) \mid \lambda \geq 0 \right\}$$

- Linear transformation functions \equiv multiplication with matrices
- Matrices \equiv vectors with n^2 components
- Subsequent transition matrices are composable

Effect of a procedure \Rightarrow Convex set of transition matrices

- Element-wise multiplication \equiv procedure call
- In absence of guards precise convex hull of *collecting semantics*

Open problem: Conditional branches handled only non-deterministically

Linear closure of transition matrices in

Idea 3: Guard handling

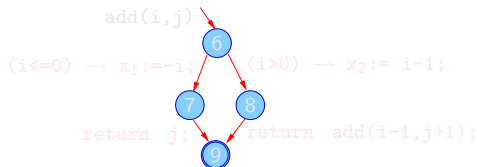
Intraprocedural: handling guards via intersection of convex sets



Interprocedural:

Intersections cannot be expressed using transition matrices

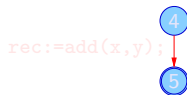
Idea: Store conditionals in auxiliary variables and postpone evaluation to intraprocedural part



Perform intersection after each procedure call:

$$\cap \{x \mid x \in \{1\} \times \mathbb{F}^n \wedge x_i \geq 0\}$$

⇒ *Safe over-approximation* of exact guard handling



Linear guards intraprocedurally in

Idea 3: Guard handling

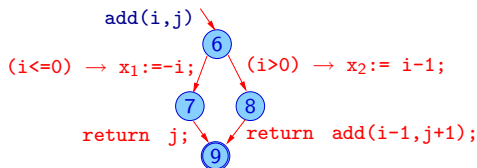
Intraprocedural: handling guards via intersection of convex sets



Interprocedural:

Intersections cannot be expressed using transition matrices

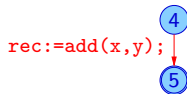
Idea: Store conditionals in auxiliary variables and postpone evaluation to intraprocedural part



Perform intersection after each procedure call:

$$\cap \{x \mid x \in \{1\} \times \mathbb{F}^n \wedge x_i \geq 0\}$$

\Rightarrow *Safe over-approximation* of exact guard handling



Linear guards intraprocedurally in

Status – Roundup

- We can specify a constraint system, whose smallest solution characterizes the *convex hull* of the valid program states.
- In absence of guarded statements, it is even exact



But for an **effective algorithm** several other assumptions have to hold:

- Effective representation of convex sets
- Effective algorithms for
 - set subsumption and union
 - effect composition and application

Status – Roundup

- We can specify a constraint system, whose smallest solution characterizes the *convex hull* of the valid program states.
- In absence of guarded statements, it is even exact



But for an **effective algorithm** several other assumptions have to hold:

- Effective representation of convex sets
- Effective algorithms for
 - set subsumption and union
 - effect composition and application

Status – Roundup

- We can specify a constraint system, whose smallest solution characterizes the *convex hull* of the valid program states.
- In absence of guarded statements, it is even exact



But for an **effective algorithm** several other assumptions have to hold:

- Effective representation of convex sets
- Effective algorithms for
 - set subsumption and union
 - effect composition and application

Status – Roundup

- We can specify a constraint system, whose smallest solution characterizes the *convex hull* of the valid program states.
- In absence of guarded statements, it is even exact

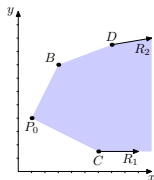


But for an **effective algorithm** several other assumptions have to hold:

- Effective representation of convex sets
- Effective algorithms for
 - set subsumption and union
 - effect composition and application

Idea 4: Polyhedral analysis

- Approximation of convex sets via convex polyhedra.
- Representation as sets of *points* \mathbf{P} , *rays* \mathbf{R} and *lines* \mathbf{L}
- Subsumption test \equiv linear programming problem
- Composition implementable approximately as multiplication of frame elements



$$\langle \mathbf{P}, \mathbf{R}, \mathbf{L} \rangle = \left\{ \sum_{i=0}^q \lambda_i P_i + \sum_{i=1}^r \mu_i R_i + \sum_{i=1}^s \eta_i L_i \mid q, r, s \geq 0 \wedge \lambda_i, \mu_i \geq 0 \wedge \sum_i \lambda_i = 1 \right\}$$

\Rightarrow Effective intraprocedural approach

\Rightarrow Even effective interprocedural extension

Problem: Polyhedral operations are rather expensive!

Linear programming for convex polyhedra in

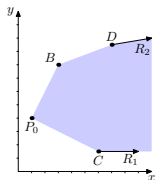
Halbwachs, Cousot

Automatic Discovery of Linear Restraints among Program Variables

1978

Idea 4: Polyhedral analysis

- Approximation of convex sets via convex polyhedra.
- Representation as sets of *points* \mathbf{P} , *rays* \mathbf{R} and *lines* \mathbf{L}
- Subsumption test \equiv linear programming problem
- Composition implementable approximately as multiplication of frame elements



$$\langle \mathbf{P}, \mathbf{R}, \mathbf{L} \rangle = \left\{ \sum_{i=0}^q \lambda_i P_i + \sum_{i=1}^r \mu_i R_i + \sum_{i=1}^s \eta_i L_i \mid q, r, s \geq 0 \wedge \lambda_i, \mu_i \geq 0 \wedge \sum_i \lambda_i = 1 \right\}$$

\Rightarrow Effective intraprocedural approach

\Rightarrow Even effective interprocedural extension

Problem: Polyhedral operations are rather expensive!

Linear programming for convex polyhedra in

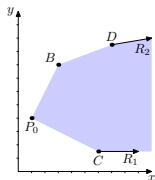
Halbwachs, Cousot

Automatic Discovery of Linear Restraints among Program Variables

1978

Idea 4: Polyhedral analysis

- Approximation of convex sets via convex polyhedra.
- Representation as sets of *points* \mathbf{P} , *rays* \mathbf{R} and *lines* \mathbf{L}
- Subsumption test \equiv linear programming problem
- Composition implementable approximately as multiplication of frame elements



$$\langle \mathbf{P}, \mathbf{R}, \mathbf{L} \rangle = \left\{ \sum_{i=0}^q \lambda_i P_i + \sum_{i=1}^r \mu_i R_i + \sum_{i=1}^s \eta_i L_i \mid q, r, s \geq 0 \wedge \lambda_i, \mu_i \geq 0 \wedge \sum_i \lambda_i = 1 \right\}$$

\Rightarrow Effective intraprocedural approach

\Rightarrow Even effective interprocedural extension

Problem: Polyhedral operations are rather expensive!

Linear programming for convex polyhedra in

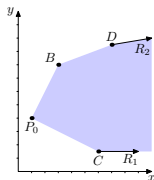
Halbwachs, Cousot

Automatic Discovery of Linear Restraints among Program Variables

1978

Idea 4: Polyhedral analysis

- Approximation of convex sets via convex polyhedra.
- Representation as sets of *points* \mathbf{P} , *rays* \mathbf{R} and *lines* \mathbf{L}
- Subsumption test \equiv linear programming problem
- Composition implementable approximately as multiplication of frame elements



$$\langle \mathbf{P}, \mathbf{R}, \mathbf{L} \rangle = \left\{ \sum_{i=0}^q \lambda_i P_i + \sum_{i=1}^r \mu_i R_i + \sum_{i=1}^s \eta_i L_i \mid q, r, s \geq 0 \wedge \lambda_i, \mu_i \geq 0 \wedge \sum_i \lambda_i = 1 \right\}$$

\Rightarrow Effective intraprocedural approach

\Rightarrow Even effective interprocedural extension

Problem: Polyhedral operations are rather expensive!

Linear programming for convex polyhedra in

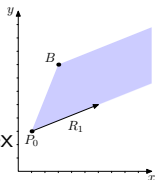
Idea 5: Simplicial analysis

Idea: Approximation of convex polyhedra

Problem: common approaches have exponentially sized frame sets

Solution: Simplices

- Maximal n frame elements and a base point
- linear independency of frame elements
- frame set same size as constraint representation
- approximation of convex polyhedra by enclosing simplex
- subsumption test reduces to linear equation system:



$$P = P_0 + \sum_{i=1}^q \lambda_i (P_i - P_0) + \sum_{i=1}^r \mu_i R_i + \sum_{i=1}^s \eta_i L_i$$

Efficient operations by solving systems of n linear equations

⇒ complexity: polynomial (≤ 5) in number of variables

Approximations of convex polyhedra in

Miné

Simon, King, Howe

The Octagon abstract domain

Two Variables per Linear Inequality as an Abstract Domain

2001

2002

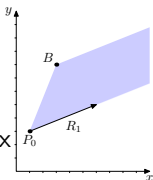
Idea 5: Simplicial analysis

Idea: Approximation of convex polyhedra

Problem: common approaches have exponentially sized frame sets

Solution: Simplices

- Maximal n frame elements and a base point
- linear independency of frame elements
- frame set same size as constraint representation
- approximation of convex polyhedra by enclosing simplex
- subsumption test reduces to linear equation system:



$$P = P_0 + \sum_{i=1}^q \lambda_i (P_i - P_0) + \sum_{i=1}^r \mu_i R_i + \sum_{i=1}^s \eta_i L_i$$

Efficient operations by solving systems of n linear equations

⇒ complexity: polynomial (≤ 5) in number of variables

Approximations of convex polyhedra in

Miné

Simon, King, Howe

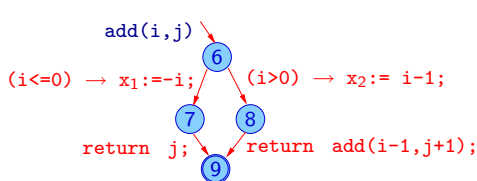
The Octagon abstract domain

Two Variables per Linear Inequality as an Abstract Domain

2001

2002

In the example



$$\begin{pmatrix} 1 \\ \text{ret} \\ i \\ j \\ x_2 \\ x_1 \end{pmatrix}$$

At program point 9, we get for the effect frame $\langle \mathbf{P}, \mathbf{R}, \mathbf{L} \rangle$:

$$\mathbf{P} = \left\{ \left(\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \right\}$$

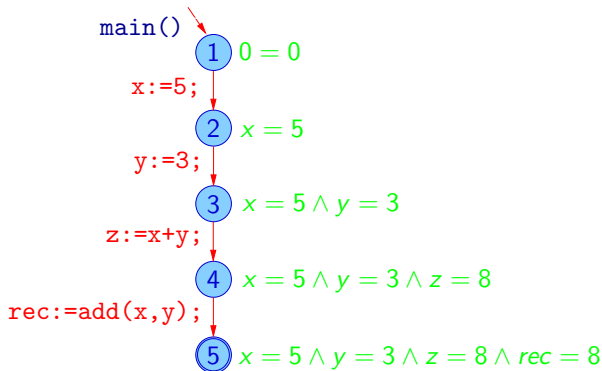
$$\mathbf{R} = \left\{ \left(\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right) \right\}$$

$$\mathbf{L} = \emptyset$$

In the example

Result

Integration of effects into reachability analysis and evaluation of postponed conditions



Experimental results

Implementation of a prototype

- local variables
- advanced enclosing simplex construction

Qualitative comparison of different approaches:

Test program	Convex Polyhedra	Simplices
recursive add	0.214 sec.	0.081 sec.
array bounds	16.70 sec.	0.492 sec.
nested loops	187.4 sec.	2.821 sec.

- simplicial analysis proved to be rather fast
- inferred inequalities quite precise by means of simplices
- direct condition evaluation no advantage to single evaluation

Conclusion:

- simplices as representation of procedure effects
- arbitrary approximations of polyhedra for reachability analysis

Thank You for Your attention!