Generation of Basic Semi-algebraic Invariants Using Convex Polyhedra

Generation of Invariant Conjunctions of Polynomial Inequalities Using Convex Polyhedra

R. Bagnara [1], E. Rodríguez-Carbonell [2], E. Zaffanella [1]

[1] University of Parma, Italy

[2] Technical University of Catalonia, Spain

Why Care about Polynomial Invariants?

- Linear invariants used to verify many classes of systems:
  - Imperative programs
  - Logic programs
  - Synchronous systems
  - Hybrid systems

Why Care about Polynomial Invariants?

- Linear invariants used to verify many classes of systems:
  - Imperative programs
  - Logic programs
  - Synchronous systems
  - Hybrid systems
- But some applications require polynomial invariants:

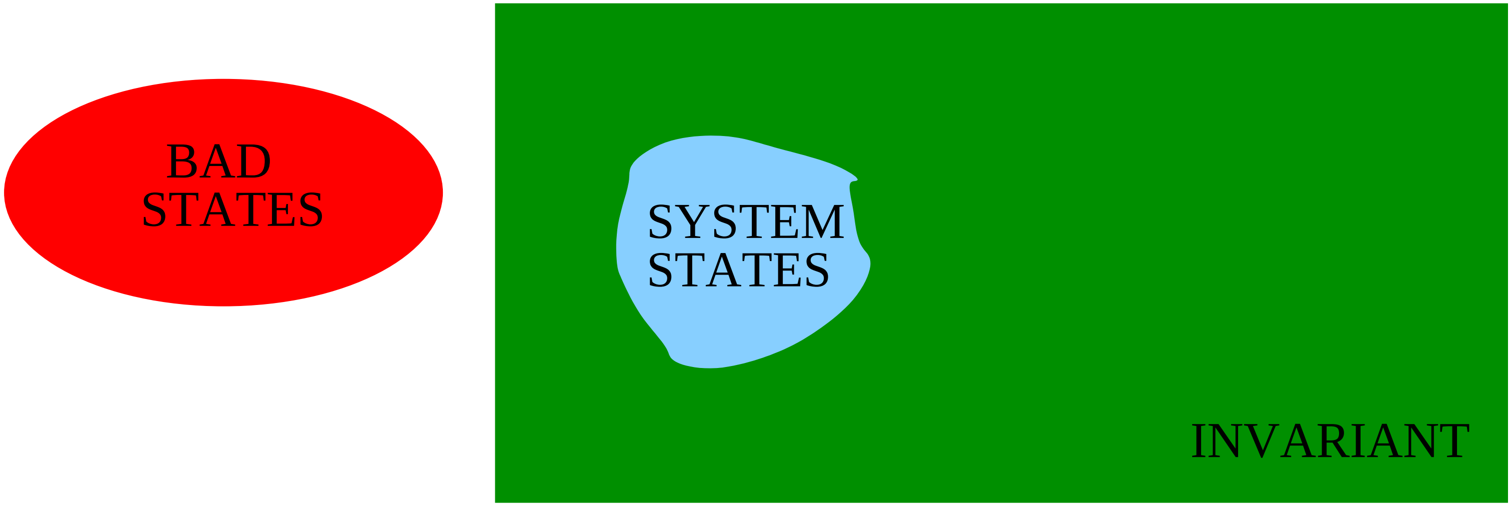
  ASTRÉE employs the ellipsoid abstract domain to verify absence of run-time errors in flight control software

Applying Invariants to Verification

CORRECTNESS OF THE SYSTEM:

$$\text{SYSTEM STATES} \ \cap \ \text{BAD STATES} \ = \ \varnothing$$

Applying Invariants to Verification



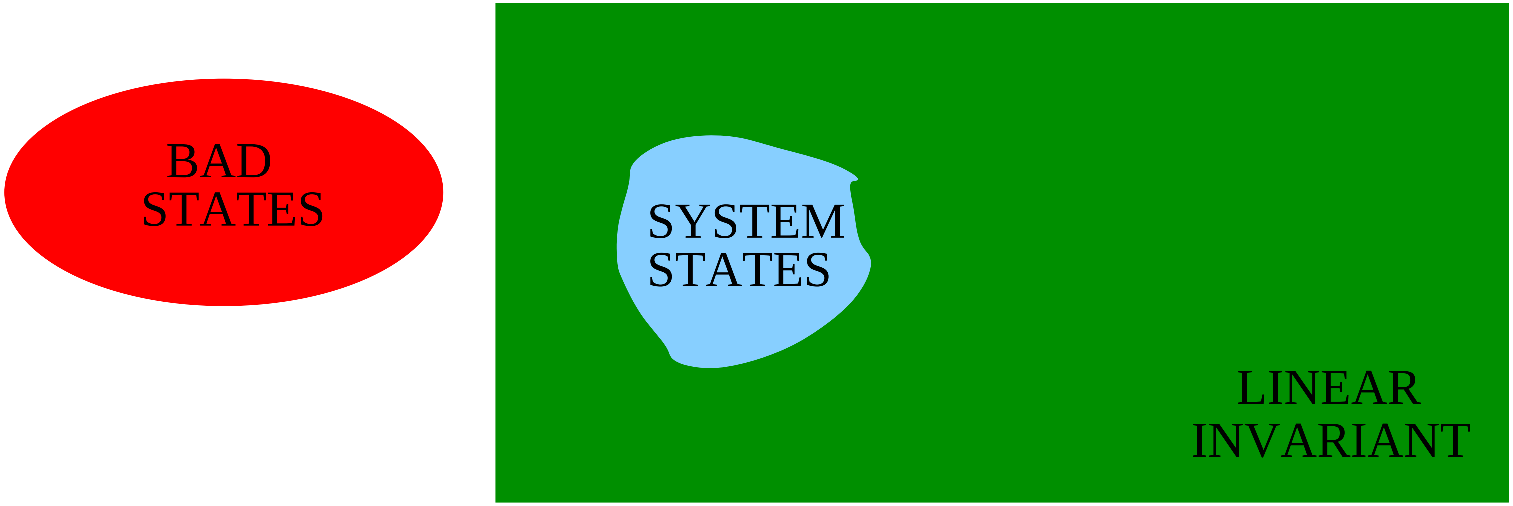
CORRECTNESS OF THE SYSTEM:

$$\text{SYSTEM STATES} \cap \text{BAD STATES} = \emptyset$$

SUFFICIENT CONDITION:

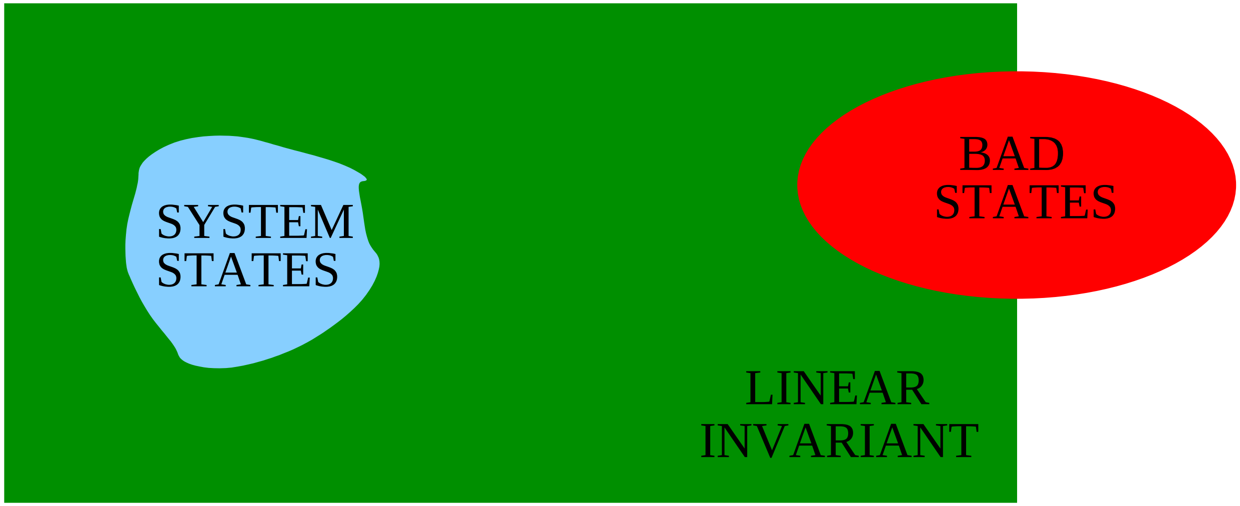
$$\text{INVARIANT} \cap \text{BAD STATES} = \emptyset$$

Applying Invariants to Verification



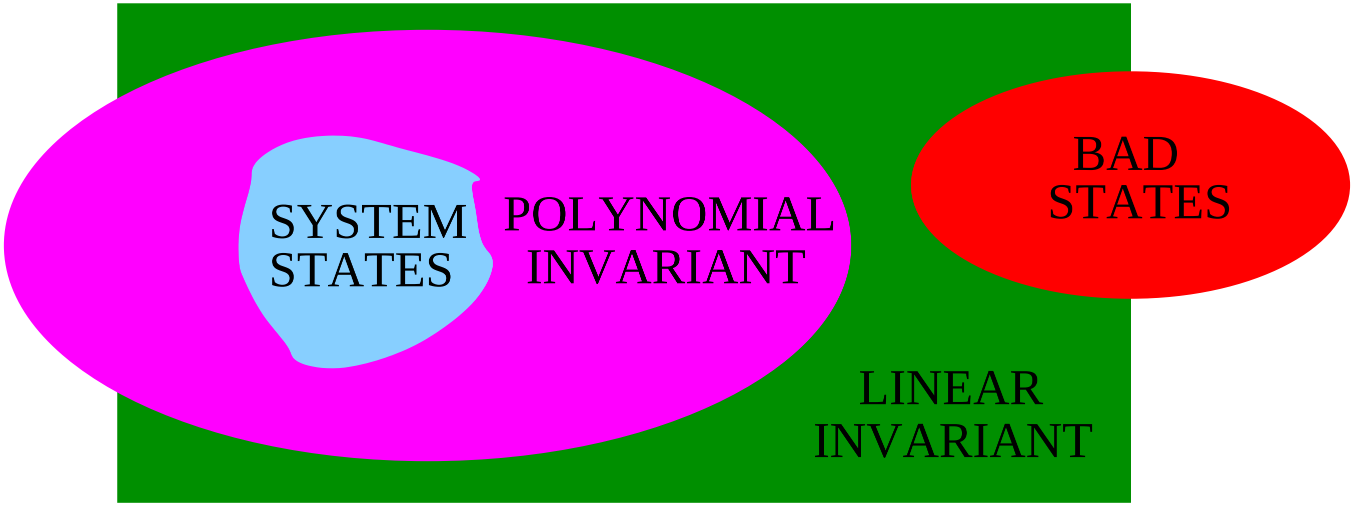
In this case the linear invariant suffices

Applying Invariants to Verification



In this case the linear invariant does not suffice

Applying Invariants to Verification



But a polynomial invariant suffices to prove correctness!

Linear vs. Polynomial Invariants

linear equalities

$$c = 2a + 1$$

[Karr'76]

Linear vs. Polynomial Invariants

linear equalities

linear inequalities

$a \geq 0 \wedge b \geq 0$

[Cousot & Halbwachs'78]

Linear vs. Polynomial Invariants

linear equalities

linear inequalities

polynomial equalities

$b = a^2 \wedge c = 2a + 1$

[Colón'04, Müller-Olm & Seidl'04, Rodríguez-Carbonell & Kapur'04, Sankaranarayanan et al.'04]

Linear vs. Polynomial Invariants

linear equalities

linear inequalities   polynomial equalities

polynomial inequalities

$$a^2 \leq b$$

[Bensalem et al.'00, Blanchet et al.'03, Kapur'04]

Overview of the Talk

- Overview of the Method
- Abstract Domain & Semantics: Polynomial Cones
- Experimental Evaluation
- Future Work & Conclusions

Overview of the Talk

- Overview of the Method
- Abstract Domain & Semantics: Polynomial Cones
- Experimental Evaluation
- Future Work & Conclusions

Drawing a Parallel from Equalities

Linear equalities
[Karr'76]

$\downarrow$

Polynomial equalities
[Colon'04]

Drawing a Parallel from Equalities

Linear equalities
[Karr'76]

↓

Polynomial equalities
[Colon'04]

Linear inequalities
[Cousot & Halbwachs'78]

↓

Polynomial inequalities
[This paper]

From Linear to Polynomial Equalities

$a := 0\ ;$
$b := 0\ ;$
$c := 1\ ;$

while $?$ **do**

$a := a + 1\ ;$
$b := b + c\ ;$
$c := c + 2\ ;$

end while

From Linear to Polynomial Equalities

$a := 0\ ;$
$b := 0\ ;$
$c := 1\ ;$

$\{\ a = 0 \wedge b = 0 \wedge c = 1\ \}$
while $?$ **do**

$a := a + 1\ ;$
$b := b + c\ ;$
$c := c + 2\ ;$

end while

From Linear to Polynomial Equalities

$a := 0\ ;$
$b := 0\ ;$
$c := 1\ ;$

$\{\ a = b \wedge c = 2a + 1\ \}$
while $?$ **do**

$a := a + 1\ ;$
$b := b + c\ ;$
$c := c + 2\ ;$

end while

From Linear to Polynomial Equalities

$a := 0$;
$b := 0$;
$c := 1$;

$\{ \ c = 2a + 1 \ \}$
while $?$ **do**

$a := a + 1$;
$b := b + c$;
$c := c + 2$;

end while

Loop invariant
$\{ \ c = 2a + 1 \ \}$

From Linear to Polynomial Equalities

$a := 0\ ;$
$b := 0\ ;$
$c := 1\ ;$

Introduce new variable s
standing for a^2

while $?$ **do**

$a := a + 1\ ;$
$b := b + c\ ;$
$c := c + 2\ ;$

end while

From Linear to Polynomial Equalities

$a := 0\ ;$
$b := 0\ ;$
$c := 1\ ;$
$s := 0\ ;$ ⟵

while $?$ **do**

$a := a + 1\ ;$
$b := b + c\ ;$
$c := c + 2\ ;$
$s := s + 2a + 1\ ;$ ⟵

end while

Introduce new variable s standing for a^2

Extend program with new variable s

$$a := 0 \quad \rightarrow \quad s := 0$$

$$a := a + 1 \quad \rightarrow \quad s := s + 2a + 1$$

From Linear to Polynomial Equalities

$a := 0\ ;$
$b := 0\ ;$
$c := 1\ ;$
$s := 0\ ;$
$\{\ a = 0 \wedge b = 0 \wedge c = 1 \wedge s = 0\ \}$
while $?$ **do**

$a := a + 1\ ;$
$b := b + c\ ;$
$c := c + 2\ ;$
$s := s + 2a + 1\ ;$

end while

From Linear to Polynomial Equalities

$a := 0\ ;$

$b := 0\ ;$

$c := 1\ ;$

$s := 0\ ;$

$\{\ a = b \wedge b = s \wedge c = 2a + 1\ \}$

while $?$ **do**

$\quad a := a + 1\ ;$

$\quad b := b + c\ ;$

$\quad c := c + 2\ ;$

$\quad s := s + 2a + 1\ ;$

end while

From Linear to Polynomial Equalities

$a := 0$;
$b := 0$;
$c := 1$;
$s := 0$;
$\{ \ b = s \wedge c = 2a + 1 \ \}$
while $?$ **do**

$a := a + 1$;
$b := b + c$;
$c := c + 2$;
$s := s + 2a + 1$;

end while

Loop invariant
$\{ \ b = a^2 \wedge c = 2a + 1 \ \}$
is more precise

From Linear to Polynomial Inequalities

$\{ \ \mathrm{Pre}: \ b \geq 0 \ \}$

$a := 0 \ ;$

while $(a+1)^2 \leq b$ **do**

$\qquad a := a + 1 \ ;$

end while

$\{ \ \mathrm{Post}: \ (a+1)^2 > b \wedge b \geq a^2 \ \}$

From Linear to Polynomial Inequalities

$\{\ \mathrm{Pre}:\ \ b \geq 0\ \}$

$a := 0\ ;$

while $(a+1)^2 \leq b$ **do**

$\quad a := a + 1\ ;$

end while

$\{\ \mathrm{Post}:\ \ (a+1)^2 > b \wedge b \geq a^2\ \}$

Linear analysis cannot deal with the quadratic condition

$$(a+1)^2 \leq b$$

From Linear to Polynomial Inequalities

$\{ \text{ Pre} : \; b \geq 0 \; \}$

$a := 0 \; ;$

$\{ \; a \geq 0 \wedge b \geq 0 \; \}$

while $(a+1)^2 \leq b$ **do**

$a := a + 1 \; ;$

end while

$\{ \text{ Post} : \; (a+1)^2 > b \wedge b \geq a^2 \; \}$

Loop invariant $\{ \; a \geq 0 \wedge b \geq 0 \; \}$

not precise enough

From Linear to Polynomial Inequalities

$\{ \text{ Pre : } \; b \geq 0 \; \}$

$a := 0 \; ;$

$s := 0 \; ;$ ⟵

while $(a+1)^2 \leq b$ **do**

$a := a + 1 \; ;$

$s := s + 2a + 1 \; ;$ ⟵

end while

$\{ \text{ Post : } \; (a+1)^2 > b \wedge b \geq a^2 \; \}$

Introduce new variable s standing for a^2

Extend program with new variable s

$$a := 0 \quad \rightarrow \quad s := 0$$

$$a := a + 1 \quad \rightarrow \quad s := s + 2a + 1$$

From Linear to Polynomial Inequalities

$\{ \text{ Pre} : \; b \geq 0 \ \}$

$a := 0 \ ;$

$s := 0 \ ;$

$\{ \ b \geq s \wedge \cdots \ \}$

while $(a+1)^2 \leq b$ **do**

$a := a + 1 \ ;$

$s := s + 2a + 1 \ ;$

end while

$\{ \text{ Post} : \; (a+1)^2 > b \wedge b \geq a^2 \ \}$

Loop invariant

$\{ \ b \geq a^2 \wedge \cdots \ \}$

enough to prove partial correctness

- Overview of the Method
- Abstract Domain & Semantics: Polynomial Cones
- Experimental Evaluation
- Future Work & Conclusions

Linearization of Polynomial Constraints

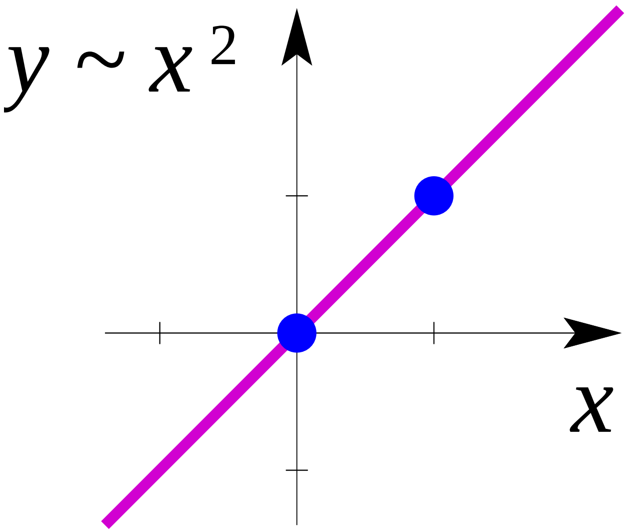
- Abstract values = sets of constraints
- Given a degree bound d, all terms x^α with $\deg(x^\alpha) \leq d$ are considered as different and independent variables

Linearization of Polynomial Constraints

- Abstract values = sets of constraints
- Given a degree bound d, all terms x^α with $\deg(x^\alpha) \leq d$ are considered as different and independent variables

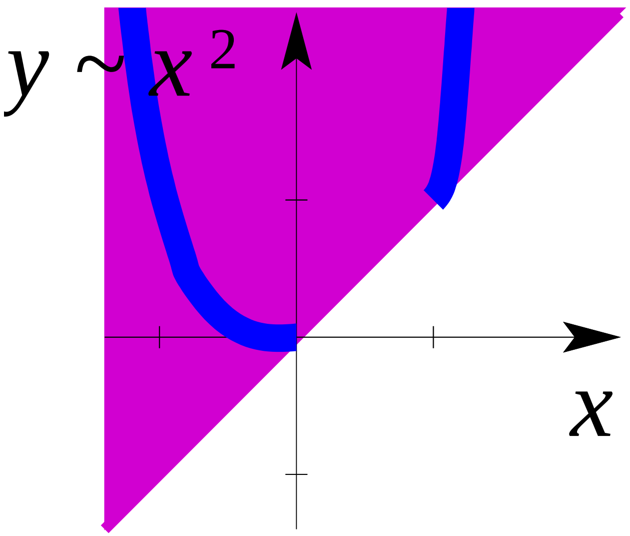
$$x^2 = x$$

$$y = x$$



$$x^2 \geq x$$

$$y \geq x$$



Vector Spaces $\leftrightarrow$ Polynomial Cones

polynomial $= 0$

- $\forall$ polynomial p, $p \sim p = 0$
- Vector space = set of polynomials V s.t.
  - $0 \in V$
  - $\forall p, q \in V$ and $\lambda, \mu \in \mathbb{R}$, $\lambda p + \mu q \in V$

$$\frac{}{0 = 0}$$

$$\frac{p = 0 \quad q = 0 \quad \lambda, \mu \in \mathbb{R}}{\lambda p + \mu q = 0}$$

Vector Spaces $\leftrightarrow$ Polynomial Cones

polynomial $= 0$	polynomial ≥ 0
$\forall$ polynomial p, $p \sim p = 0$	$\forall$ polynomial p, $p \sim p \geq 0$
Vector space = set of polynomials V s.t.	Polynomial cone = set of polynomials C s.t.
$0 \in V$	$1 \in C$
$\forall p, q \in V$ and $\lambda, \mu \in \mathbb{R}$, $\lambda p + \mu q \in V$	$\forall p, q \in C$ and $\lambda, \mu \in \mathbb{R}_+$, $\lambda p + \mu q \in C$
$\dfrac{}{0 = 0}$	$\dfrac{}{1 \geq 0}$
$\dfrac{p = 0 \quad q = 0 \quad \lambda, \mu \in \mathbb{R}}{\lambda p + \mu q = 0}$	$\dfrac{p \geq 0 \quad q \geq 0 \quad \lambda, \mu \in \mathbb{R}_+}{\lambda p + \mu q \geq 0}$

Explicitly Adding Other Inference Rules

polynomial $= 0$

$$\frac{p = 0 \quad \deg(pq) \leq d}{pq = 0}$$

Closure of a vector space V:

- $\forall p \in V, q$ any polynomial such that $\deg(pq) \leq d$, then $pq \in V$

Explicitly Adding Other Inference Rules

polynomial $= 0$

$$\frac{p = 0 \quad \deg(pq) \leq d}{pq = 0}$$

Closure of a vector space V:

- $\forall p \in V, q$ any polynomial such that $\deg(pq) \leq d$, then $pq \in V$

polynomial ≥ 0

$$\frac{p \geq 0 \quad p \leq 0 \quad \deg(pq) \leq d}{pq \geq 0}$$

$$\frac{p \geq 0 \quad q \geq 0 \quad \deg(pq) \leq d}{pq \geq 0}$$

Closure of a polynomial cone C:

- $\forall p \in C, q$ any polynomial such that $-p \in C$ and $\deg(pq) \leq d$, then $pq \in C$
- $\forall p, q \in C$ such that $\deg(pq) \leq d$, then $pq \in C$

Abstract Semantics

- (Multiple) assignments $x := f(x)$
  $\forall\ x_i$ variable, $f_i(x) \equiv\ ?$ or is a polynomial
  Term x^α is updated according to the following cases:

Abstract Semantics

- (Multiple) assignments $x \ := \ f(x)$
  $\forall\ x_i$ variable, $f_i(x) \equiv \ ?$ or is a polynomial
  Term x^α is updated according to the following cases:
  - $\exists\ x_i$ with $\alpha_i \neq 0$ and $x_i \ := \ ? \rightarrow x^\alpha \ := \ ?$

$$a \ := \ ? \quad \longrightarrow \quad a^2 \ := \ ?$$

Abstract Semantics

- (Multiple) assignments $x \ := \ f(x)$
  $\forall\ x_i$ variable, $f_i(x) \equiv\ ?$ or is a polynomial
  Term x^α is updated according to the following cases:
  - $\exists\ x_i$ with $\alpha_i \neq 0$ and $x_i \ := \ ? \rightarrow x^\alpha \ := \ ?$
  - $\deg(\Pi f_i^{\alpha_i}(x)) > d \rightarrow x^\alpha \ := \ ?$

$$d = 2, \ \ a \ := \ a^2 \quad \longrightarrow \quad a^2 \ := \ ?$$

Abstract Semantics

- (Multiple) assignments $x \; := \; f(x)$
  $\forall \, x_i$ variable, $f_i(x) \equiv \; ?$ or is a polynomial
  Term x^α is updated according to the following cases:
  - $\exists \, x_i$ with $\alpha_i \neq 0$ and $x_i \; := \; ? \rightarrow x^\alpha \; := \; ?$
  - $\deg(\Pi f_i^{\alpha_i}(x)) > d \rightarrow x^\alpha \; := \; ?$
  - otherwise $\rightarrow$ linearization of $\Pi f_i^{\alpha_i}(x)$

$$d = 2, \;\; a \; := \; a + 1 \quad \longrightarrow \quad a^2 \; := \; a^2 + 2a + 1$$

Abstract Semantics

- (Multiple) assignments $x := f(x)$
  $\forall\ x_i$ variable, $f_i(x) \equiv\ ?$ or is a polynomial
  Term x^α is updated according to the following cases:
  - $\exists\ x_i$ with $\alpha_i \neq 0$ and $x_i := ? \rightarrow x^\alpha := ?$
  - $\deg(\Pi f_i^{\alpha_i}(x)) > d \rightarrow x^\alpha := ?$
  - otherwise $\rightarrow$ linearization of $\Pi f_i^{\alpha_i}(x)$
- Intersection: closure (by bounded-degree product)

Abstract Semantics

- (Multiple) assignments $x := f(x)$
  $\forall\, x_i$ variable, $f_i(x) \equiv\ ?$ or is a polynomial
  Term x^α is updated according to the following cases:
  - $\exists\, x_i$ with $\alpha_i \neq 0$ and $x_i := ? \to x^\alpha := ?$
  - $\deg(\Pi f_i^{\alpha_i}(x)) > d \to x^\alpha := ?$
  - otherwise $\to$ linearization of $\Pi f_i^{\alpha_i}(x)$
- Intersection: closure (by bounded-degree product)
- Union: same as convex polyhedra
- Test for inclusion: same as convex polyhedra
- Widening: same as convex polyhedra

Approximating closure

- Bounded-degree product closure not finitely computable: might yield not finitely generated polynomial cones

Approximating closure

- Bounded-degree product closure not finitely computable: might yield not finitely generated polynomial cones
- Conservatively approximated: given a formula

$$f_1 = 0 \ \wedge \ \cdots \ \wedge \ f_n = 0 \ \wedge \ g_1 \geq 0 \ \wedge \ \cdots \ \wedge \ g_m \geq 0$$

  we add the constraints:
  - $x^\alpha f_i = 0$ where $\deg(x^\alpha f_i) \leq d$
  - $\Pi g_{i_j} \geq 0$ where $\deg(\Pi g_{i_j}) \leq d$

- Overview of the Method
- Abstract Domain & Semantics: Polynomial Cones
- Experimental Evaluation
- Future Work & Conclusions

Implementation

- Prototype implemented in C/C++ for $d = 2$
- Based on the Parma Polyhedra Library (PPL)
  [Bagnara, Ricci, Hill & Zaffanella'02]
  - Efficient and robust implementation of polyhedra
  - Support for time-bounded computations

Implementation

- Prototype implemented in C/C++ for $d = 2$
- Based on the Parma Polyhedra Library (PPL)
  [Bagnara, Ricci, Hill & Zaffanella'02]
- Analysis performed in two steps:
  1. Linear analysis
  2. Quadratic analysis exploiting linear invariants
- Widening strategies
  - Standard widening [Cousot & Halbwachs'78]
  - Widening "up to" [Halbwachs'93]
  - Refined widenings [Bagnara, Hill, Ricci & Zaffanella'03'05]

Evaluation on Benchmark Suite

- Benchmark suite consisting of:
  - FAST suite [Bardin et al.'03]
  - StInG suite [Sankaranarayanan et al.'04]
  - Programs from the literature
- Results:
  - For 80 % of programs:
    our linear invariants are as strong as StInG's
  - For 33 % of programs:
    our linear invariants are better than StInG's
  - For 50 % of programs:
    quadratic invariants improve on linear invariants

- Overview of the Method
- Abstract Domain & Semantics: Polynomial Cones
- Experimental Evaluation
- Future Work & Conclusions

Future Work

- Extend admissible assignments to nondeterministic assignments of the form

$$f(x) \leq x_i' \leq g(x)$$

- Introduce other inference rules to improve precision

$$\frac{p \text{ is a sum of squares}}{p \geq 0}$$

- Adapt widenings to nonlinear invariant generation
- Improve the current implementation

Conclusions

- Abstract domain for generating invariant conjunctions of polynomial inequalities
- Built upon the abstract domain of convex polyhedra
- Implemented in C/C++ with encouraging experimental results

Thank you!