Proving Unsatisfiability in Non-linear Arithmetic by Duality

[work in progress]

Daniel Larraz, Albert Oliveras, Enric Rodríguez-Carbonell and Albert Rubio

Universitat Politècnica de Catalunya, Barcelona, Spain

Dagstuhl Seminar on Deduction and Arithmetic, October 2013

Overview of the Talk

- **Introduction**
- **Motivation**
  - Constraint-based program analysis
- **Non-linear constraint solving**
  - Related work in SMT(NA) [NA = Non-linear Arithmetic]
  - Review of [Borralleras et al., JAR'12]: pros and cons
  - Duality: Positivstellensatz
  - Proving unsatisfiability by finding solutions
- **Open questions and future work**

Introduction

- Non-linear Constraint Solving: Given a quantifier-free formula F containing polynomial inequality atoms, is F satisfiable?
- In $\mathbb{Z}$: undecidable (Hilbert's 10th problem)
- In $\mathbb{R}$: decidable, even with quantifiers (Tarski).
  But traditional algorithms have prohibitive worst-case complexity
- Lots of applications: non-linear constraints arise in many contexts.
  Here, focus will be on program analysis
- Goal: a procedure that works well in practice for our application

Targeted Programs

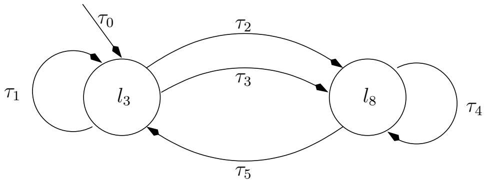
- Imperative programs
- Integer variables and linear expressions
  (other constructions considered unknowns)

```
int gcd ( int a, int b ) {
    int tmp;
    while ( a >= 0 && b > 0 ) {
        tmp = b;
        if (a == b)   b = 0;
        else {
            int z = a;
            while ( z > b )   z -= b;
            b = z; }
        a = tmp; }
    return a; }
```

Targeted Programs

- Imperative programs
- Integer variables and linear expressions
  (other constructions considered unknowns)

As a transition system:



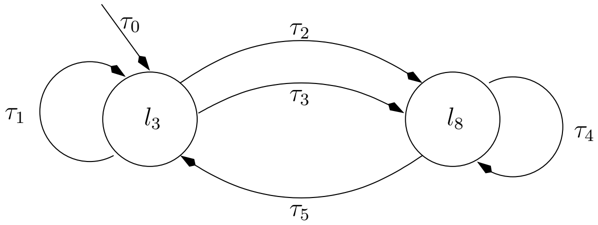
$$
\begin{array}{lllllllll}
\tau_0: & & & & a' = ?, & b' = ?, & tmp' = ?, & z' = ? \\
\tau_1: & b \geq 1, & a \geq 0, & a = b, & a' = b, & b' = 0, & tmp' = b, & z' = z \\
\tau_2: & b \geq 1, & a \geq 0, & a < b, & a' = a, & b' = b, & tmp' = b, & z' = a \\
\tau_3: & b \geq 1, & a \geq 0, & a > b, & a' = a, & b' = b, & tmp' = b, & z' = a \\
\tau_4: & b < z, & & & a' = a, & b' = b, & tmp' = tmp, & z' = z - b \\
\tau_5: & b \geq z, & & & a' = tmp, & b' = z, & tmp' = tmp, & z' = z
\end{array}
$$

Invariants

- An invariant of a program at a location is an assertion over the program variables that is true whenever the location is reached
- Useful in safety analysis:
  if F are forbidden states, prove that $\neg F$ is (implied by an) invariant
- An invariant is inductive at a program location if:
  - *Initiation condition:* it holds the first time the location is reached
  - *Consecution condition:* it is preserved by every cycle back to location

We are interested in inductive invariants

Invariants



$$
\begin{array}{llllllll}
\tau_0: & & & & a' = ?, & b' = ?, & tmp' = ?, & z' = ? \\
\tau_1: & b \geq 1, & a \geq 0, & a = b, & a' = b, & b' = 0, & tmp' = b, & z' = z \\
\tau_2: & b \geq 1, & a \geq 0, & a < b, & a' = a, & b' = b, & tmp' = b, & z' = a \\
\tau_3: & b \geq 1, & a \geq 0, & a > b, & a' = a, & b' = b, & tmp' = b, & z' = a \\
\tau_4: & b < z, & & & a' = a, & b' = b, & tmp' = tmp, & z' = z - b \\
\tau_5: & b \geq z, & & & a' = tmp, & b' = z, & tmp' = tmp, & z' = z
\end{array}
$$

Assertion $b \geq 1$ is invariant at l_8

Constraint-based (Linear) Invariant Generation

Introduced in [Colón,Sankaranarayanan & Sipma, CAV'03]

Keys:

Constraint-based (Linear) Invariant Generation

Introduced in [Colón,Sankaranarayanan & Sipma, CAV'03]

Keys:

- Fix a template of candidate invariant

$$\alpha_1 x_1 + \ldots + \alpha_n x_n \geq \beta$$

where $\alpha_1, \ldots, \alpha_n, \beta$ are unknowns, for each program location

Constraint-based (Linear) Invariant Generation

Introduced in [Colón,Sankaranarayanan & Sipma, CAV'03]

Keys:

- Fix a template of candidate invariant

$$\alpha_1 x_1 + \ldots + \alpha_n x_n \geq \beta$$

  where $\alpha_1, \ldots, \alpha_n, \beta$ are unknowns, for each program location

- Impose initiation and consecution conditions obtaining $\exists\forall$ problem

Constraint-based (Linear) Invariant Generation

Introduced in [Colón,Sankaranarayanan & Sipma, CAV'03]

Keys:

- Fix a template of candidate invariant

$$\alpha_1 x_1 + \ldots + \alpha_n x_n \geq \beta$$

  where $\alpha_1, \ldots, \alpha_n, \beta$ are unknowns, for each program location

- Impose initiation and consecution conditions obtaining $\exists\forall$ problem

- Transform into $\exists$ problem over non-linear arith. with Farkas' Lemma

Constraint-based (Linear) Invariant Generation

Introduced in [Colón,Sankaranarayanan & Sipma, CAV'03]

Keys:

- Fix a template of candidate invariant

$$\alpha_1 x_1 + \ldots + \alpha_n x_n \geq \beta$$

  where $\alpha_1, \ldots, \alpha_n, \beta$ are unknowns, for each program location

- Impose initiation and consecution conditions obtaining $\exists\forall$ problem
- Transform into $\exists$ problem over non-linear arith. with Farkas' Lemma
- Solve resulting non-linear constraints

Farkas' Lemma

In matrix notation:

$$(\forall x \in \mathbb{R}^n)\ (Ax \geq b \Rightarrow c^T x \geq d)$$

iff

$$(\exists \lambda \in \mathbb{R}^m)\ (\lambda \geq 0\ \wedge\ ((\lambda^T A = c^T\ \wedge\ \lambda^T b \geq d) \vee (\lambda^T A = 0\ \wedge\ \lambda^T b = -1))$$

Particularities of Our SMT(NA) Problems

- Existentially quantified variables are:
  - unknown template coefficients of invariants and ranking functions
  - Farkas' multipliers
- Non-linear monomials are quadratic of the form

unknown template coefficient · Farkas' multiplier

- Existentially quantified variables are of real type...
  But it is reasonable to assume that if satisfiable there is a solution where unknown template coefficients are integers

(when we program, we think invariants/ranking functs. with integer coefficients, right?)

Related Work in SMT(NA)

- Methods aimed at proving unsatisfiability:
  - Gröbner bases [Tiwari, CSL'05; De Moura, Passmore, SMT'09]
  - Semidefinite programming [Parrilo, MP'03]
  - Mixed approaches [Platzer, Quesel, Rummer, CADE'09]
- Methods aimed at proving satisfiability:
  - Cylindrical Algebraic Decomposition (CAD) [Collins, ATFL'75]
  - Translating into
    - SAT [Fuhs et al., SAT'07]
    - SMT(BV) [Zankl, Middeldorp, LPAR'10]
    - SMT(LA) [Borralleras et al., JAR'12]
  - Model-constructing satisfiability calculus [De Moura, Jovanovic, IJCAR'12]

Review of [Borralleras et al., JAR'12]

- Our method is aimed at proving satisfiability in the integers (as opposed to finding non-integer solutions, or proving unsatisfiability)
- Basic idea: use bounds on integer variables to linearize the formula
- Refinement: analyze unsatisfiable cores to enlarge bounds (and sometimes even prove unsatisfiability)

Review of [Borralleras et al., JAR'12]

- For any formula there is an equisatisfiable one of the form

$$F \wedge (\bigwedge_i y_i = M_i)$$

  where F is linear and each M_i is non-linear

- Example

$$u^4 v^2 + 2u^2 vw + w^2 \leq 4 \ \wedge \ 1 \leq u, v, w \leq 2$$

$$x_{u^4 v^2} + 2x_{u^2 vw} + x_{w^2} \leq 4 \ \wedge \ 1 \leq u, v, w \leq 2 \ \wedge$$
$$x_{u^4 v^2} = u^4 v^2 \ \wedge \ x_{u^2 vw} = u^2 vw \ \wedge \ x_{w^2} = w^2$$

Review of [Borralleras et al., JAR'12]

- Idea: linearize non-linear monomials with case analysis on some of the variables with finite domain
- Assume variables are in $\mathbb{Z}$
- $F \;\wedge\; x_{u^4v^2} = u^4v^2 \;\wedge\; x_{u^2vw} = u^2vw \;\wedge\; x_{w^2} = w^2$
  where F is $x_{u^4v^2} + 2x_{u^2vw} + x_{w^2} \leq 4 \;\wedge\; 1 \leq u, v, w \leq 2$
- Since $1 \leq w \leq 2$, add $x_{u^2v} = u^2v$ and
  $w = 1 \;\rightarrow\; x_{u^2vw} = x_{u^2v}$
  $w = 2 \;\rightarrow\; x_{u^2vw} = 2x_{u^2v}$

Review of [Borralleras et al., JAR'12]

Applying the same idea recursively, the following linear formula is obtained:

$$
\begin{array}{l}
x_{u^4v^2} + 2x_{u^2vw} + x_{w^2} \leq 4 \\
\wedge\ 1 \leq u, v, w \leq 2 \\
\wedge\ w = 1 \ \rightarrow\ x_{u^2vw} = x_{u^2v} \\
\wedge\ w = 2 \ \rightarrow\ x_{u^2vw} = 2x_{u^2v} \\
\wedge\ u = 1 \ \rightarrow\ x_{u^2v} = v \\
\wedge\ u = 2 \ \rightarrow\ x_{u^2v} = 4v \\
\wedge\ w = 1 \ \rightarrow\ x_{w^2} = 1 \\
\wedge\ w = 2 \ \rightarrow\ x_{w^2} = 4 \\
\wedge\ v = 1 \ \rightarrow\ x_{u^4v^2} = x_{u^4} \\
\wedge\ v = 2 \ \rightarrow\ x_{u^4v^2} = 4x_{u^4} \\
\wedge\ u = 1 \ \rightarrow\ x_{u^4} = 1 \\
\wedge\ u = 2 \ \rightarrow\ x_{u^4} = 16
\end{array}
$$

A model can be computed:

$$
\begin{array}{l}
u = 1 \\
v = 1 \\
w = 1 \\
x_{u^4v^2} = 1 \\
x_{u^4} = 1 \\
x_{u^2vw} = 1 \\
x_{u^2v} = 1 \\
x_{w^2} = 1
\end{array}
$$

Review of [Borralleras et al., JAR'12]

- If linearization achieves a linear formula then we have a sound and complete decision procedure

  Note also that actually not all variables need to be integers: only enough to get a linear formula

- If we don't have enough variables with finite domain...
  ... we can add bounds at cost of losing completeness
  We cannot trust UNSAT answers any more!

- But we can analyze why the CNF is UNSAT:
  an unsatisfiable core (= unsatisfiable subset of clauses) can be obtained from the trace of the DPLL execution [Zhang & Malik'03]

- If core contains no extra bound: truly UNSAT

  If core contains extra bound: guide to enlarge domains

Review of [Borralleras et al., JAR'12]

- $u^4 v^2 + 2u^2 vw + w^2 \leq 3$ cannot be linearized
- Consider $u^4 v^2 + 2u^2 vw + w^2 \leq 3 \;\; \wedge 1 \leq u, v, w \leq 2$
- The linearization is unsatisfiable:

$$
\begin{array}{l}
x_{u^4 v^2} + 2x_{u^2 vw} + x_{w^2} \leq 3 \\
\wedge\, 1 \leq x_{u^4 v^2} \;\; \wedge\, x_{u^4 v^2} \leq 64 \\
\wedge\, 1 \leq x_{u^2 vw} \;\; \wedge\, x_{u^2 vw} \leq 16 \\
\wedge\, 1 \leq x_{w^2} \;\; \wedge\, x_{w^2} \leq 4 \\
\wedge\, 1 \leq u \;\; \wedge\, u \leq 2 \\
\wedge\, 1 \leq v \;\; \wedge\, v \leq 2 \\
\wedge\, 1 \leq w \;\; \wedge\, w \leq 2 \\
\ldots
\end{array}
$$

- Should decrease lower bounds for u, v, w

Review of [Borralleras et al., JAR'12]

- In favour: very effective when handling satisfiable instances
  - Best solver in QF_NIA division in SMT-COMP'09, SMT-COMP'10
  - According to our experiments, even faster than latest version of Z3 on benchmarks coming from our application

- Against: often fails to detect unsatisfiability on unsatisfiable instances (and then keeps enlarging domains forever!)

  Need more powerful non-linear reasoning than with unsat cores!

- Let's focus on conjunctions of polynomial inequalities from now on

Duality: Positivstellensatz

- Idea: (following [Parrilo, MP'03])
  exploit the effectiveness on sat instances by applying duality

Duality: Positivstellensatz

- Idea: (following [Parrilo, MP'03])
  exploit the effectiveness on sat instances by applying duality
- Some definitions: given $A \subseteq \mathbb{Q}[x]$, where $x = x_1, \ldots, x_n$:
  - the multiplicative monoid generated by A, $\mathrm{Monoid}(A)$, is the set of products of zero or more elements in A

Duality: Positivstellensatz

- Idea: (following [Parrilo, MP'03])
  exploit the effectiveness on sat instances by applying duality
- Some definitions: given $A \subseteq \mathbb{Q}[x]$, where $x = x_1, \ldots, x_n$:
  - the multiplicative monoid generated by A, $\mathrm{Monoid}(A)$, is the set of products of zero or more elements in A
  - the cone generated by A, $\mathrm{Cone}(A)$, is the set of sums of products of the form pPQ^2, where $p \in \mathbb{Q}$, $p > 0$, $P \in \mathrm{Monoid}(A)$ and $Q \in \mathbb{Q}[x]$

Duality: Positivstellensatz

- Idea: (following [Parrilo, MP'03])
  exploit the effectiveness on sat instances by applying duality
- Some definitions: given $A \subseteq \mathbb{Q}[x]$, where $x = x_1, \ldots, x_n$:
  - the multiplicative monoid generated by A, $\mathrm{Monoid}(A)$, is the set of products of zero or more elements in A
  - the cone generated by A, $\mathrm{Cone}(A)$, is the set of sums of products of the form pPQ^2, where $p \in \mathbb{Q}$, $p > 0$, $P \in \mathrm{Monoid}(A)$ and $Q \in \mathbb{Q}[x]$
  - the ideal generated by A, $\mathrm{Ideal}(A)$, is the set of sums of products of the form PQ, where $P \in A$ and $Q \in \mathbb{Q}[x]$

Duality: Positivstellensatz

- Idea: (following [Parrilo, MP'03])
  exploit the effectiveness on sat instances by applying duality
- Some definitions: given $A \subseteq \mathbb{Q}[x]$, where $x = x_1, \ldots, x_n$:
  - the multiplicative monoid generated by A, $\mathrm{Monoid}(A)$, is the set of products of zero or more elements in A
  - the cone generated by A, $\mathrm{Cone}(A)$, is the set of sums of products of the form pPQ^2, where $p \in \mathbb{Q}$, $p > 0$, $P \in \mathrm{Monoid}(A)$ and $Q \in \mathbb{Q}[x]$
  - the ideal generated by A, $\mathrm{Ideal}(A)$, is the set of sums of products of the form PQ, where $P \in A$ and $Q \in \mathbb{Q}[x]$
- **Positivstellensatz:** Let $F_>, F_\geq, F_= \subset \mathbb{Q}[x]$. The system

$$\{f > 0 \mid f \in F_>\} \ \cup \ \{f \geq 0 \mid f \in F_\geq\} \ \cup \ \{f = 0 \mid f \in F_=\}$$

  is unsatisfiable in $\mathbb{R}^n$ iff there are $P \in \mathrm{Monoid}(F_>)$, $Q \in \mathrm{Cone}(F_> \cup F_\geq)$ and $R \in \mathrm{Ideal}(F_=)$ such that $P + Q + R = 0$

Proving Unsatisfiability by Finding Solutions

We can prove a system unsatisfiable by finding a solution to another one!

Find the Positivstellensatz witness P, Q, R as follows:

- Set a degree bound d

Proving Unsatisfiability by Finding Solutions

We can prove a system unsatisfiable by finding a solution to another one!

Find the Positivstellensatz witness P, Q, R as follows:

- Set a degree bound d
- If $F_= = \{f_1, \ldots, f_m\}$, then R is of the form $R = \sum_{i=1}^{m} R_i f_i$. Let R_i be template polynomial with unknown coeffs and $deg(R_i) = d - deg(f_i)$.

Proving Unsatisfiability by Finding Solutions

We can prove a system unsatisfiable by finding a solution to another one!

Find the Positivstellensatz witness P, Q, R as follows:

- Set a degree bound d
- If $F_= = \{f_1, \ldots, f_m\}$, then R is of the form $R = \sum_{i=1}^m R_i f_i$. Let R_i be template polynomial with unknown coeffs and $deg(R_i) = d - deg(f_i)$.
- Let $\mathrm{Monoid}_d(F_>) = \{f_1, \ldots, f_m\}$ be the polynomials in $\mathrm{Monoid}(F_>)$ of degree $\leq d$. Then P is of the form $P = \sum_{i=1}^m p_i f_i$, where $p_i \geq 0$ are unknown coefficients with the additional constraint $\vee_{i=1}^m p_i > 0$.

Proving Unsatisfiability by Finding Solutions

We can prove a system unsatisfiable by finding a solution to another one!

Find the Positivstellensatz witness P, Q, R as follows:

- Set a degree bound d
- If $F_= = \{f_1, \ldots, f_m\}$, then R is of the form $R = \sum_{i=1}^m R_i f_i$. Let R_i be template polynomial with unknown coeffs and $deg(R_i) = d - deg(f_i)$.
- Let $\mathrm{Monoid}_d(F_>) = \{f_1, \ldots, f_m\}$ be the polynomials in $\mathrm{Monoid}(F_>)$ of degree $\leq d$. Then P is of the form $P = \sum_{i=1}^m p_i f_i$, where $p_i \geq 0$ are unknown coefficients with the additional constraint $\vee_{i=1}^m p_i > 0$.
- Let $\mathrm{Monoid}_d(F_> \cup F_\geq) = \{f_1, \ldots, f_m\}$. Then Q is of the form $Q = \sum_{i=1}^m Q_i f_i$, where Q_i is a template polynomial with unknown coeffs which is a sum of squares and has $deg(Q_i) = d - deg(f_i)$.

Proving Unsatisfiability by Finding Solutions

We can prove a system unsatisfiable by finding a solution to another one!

Find the Positivstellensatz witness P, Q, R as follows:

- Set a degree bound d
- If $F_= = \{f_1, \ldots, f_m\}$, then R is of the form $R = \sum_{i=1}^{m} R_i f_i$. Let R_i be template polynomial with unknown coeffs and $deg(R_i) = d - deg(f_i)$.
- Let $\mathrm{Monoid}_d(F_>) = \{f_1, \ldots, f_m\}$ be the polynomials in $\mathrm{Monoid}(F_>)$ of degree $\leq d$. Then P is of the form $P = \sum_{i=1}^{m} p_i f_i$, where $p_i \geq 0$ are unknown coefficients with the additional constraint $\vee_{i=1}^{m} p_i > 0$.
- Let $\mathrm{Monoid}_d(F_> \cup F_\geq) = \{f_1, \ldots, f_m\}$. Then Q is of the form $Q = \sum_{i=1}^{m} Q_i f_i$, where Q_i is a template polynomial with unknown coeffs which is a sum of squares and has $deg(Q_i) = d - deg(f_i)$.
- In $P + Q + R$, make coeffs of every monomial in the x vars equal to 0

Proving Unsatisfiability by Finding Solutions

We can prove a system unsatisfiable by finding a solution to another one!

Find the Positivstellensatz witness P, Q, R as follows:

- Set a degree bound d
- If $F_= = \{f_1, \ldots, f_m\}$, then R is of the form $R = \sum_{i=1}^m R_i f_i$. Let R_i be template polynomial with unknown coeffs and $deg(R_i) = d - deg(f_i)$.
- Let $\mathrm{Monoid}_d(F_>) = \{f_1, \ldots, f_m\}$ be the polynomials in $\mathrm{Monoid}(F_>)$ of degree $\leq d$. Then P is of the form $P = \sum_{i=1}^m p_i f_i$, where $p_i \geq 0$ are unknown coefficients with the additional constraint $\vee_{i=1}^m p_i > 0$.
- Let $\mathrm{Monoid}_d(F_> \cup F_\geq) = \{f_1, \ldots, f_m\}$. Then Q is of the form $Q = \sum_{i=1}^m Q_i f_i$, where Q_i is a template polynomial with unknown coeffs which is a sum of squares and has $deg(Q_i) = d - deg(f_i)$.
- In $P + Q + R$, make coeffs of every monomial in the x vars equal to 0

Solutions to these constraints yield unsatisfiability witnesses!

Example

- Let us consider the system $-x^2 - xy - x - 1 \geq 0 \ \wedge \ y = 0.$
- Then $F_> = \{\}$, $F_\geq = \{-x^2 - xy - x - 1\}$, $F_= = \{y\}$.
- Set degree bound $d = 2$.

Example

- Let us consider the system $-x^2 - xy - x - 1 \geq 0 \ \wedge \ y = 0.$
- Then $F_> = \{\}$, $F_\geq = \{-x^2 - xy - x - 1\}$, $F_= = \{y\}$.
- Set degree bound $d = 2$.
- $R \equiv (\alpha_x \, x + \alpha_y \, y + \alpha_0) \, y,$ where α_x, α_y, α_0 are unknowns

Example

- Let us consider the system $-x^2 - xy - x - 1 \geq 0 \ \wedge \ y = 0.$
- Then $F_> = \{\}$, $F_\geq = \{-x^2 - xy - x - 1\}$, $F_= = \{y\}$.
- Set degree bound $d = 2$.
- $R \equiv (\alpha_x\, x + \alpha_y\, y + \alpha_0)\, y$, where α_x, α_y, α_0 are unknowns
- As $F_> = \{\}$, we have $\mathrm{Monoid}_d(F_>) = \mathrm{Monoid}(F_>) = \{1\}$
  $P \equiv \beta$, where β is an unknown constrained to $\beta > 0$

Example

- Let us consider the system $-x^2 - xy - x - 1 \geq 0 \ \wedge \ y = 0.$
- Then $F_> = \{\}$, $F_\geq = \{-x^2 - xy - x - 1\}$, $F_= = \{y\}$.
- Set degree bound $d = 2$.
- $R \equiv (\alpha_x\, x + \alpha_y\, y + \alpha_0)\ y$, where α_x, α_y, α_0 are unknowns
- As $F_> = \{\}$, we have $\mathrm{Monoid}_d(F_>) = \mathrm{Monoid}(F_>) = \{1\}$
  $P \equiv \beta$, where β is an unknown constrained to $\beta > 0$
- $\mathrm{Monoid}_d(F_> \cup F_\geq) = \{1, -x^2 - xy - x - 1\}$.

  $$Q \equiv \underbrace{(\gamma_{x^2} x^2 + \gamma_{xy} xy + \gamma_{y^2} y^2 + \gamma_x x + \gamma_y y + \gamma_0)}_{\Gamma(x,y)} + \gamma'_0(-x^2 - xy - x - 1)$$

  where γ_* are unknowns s.t. $\Gamma(x, y)$ is a sum of squares, and $\gamma'_0 \geq 0$

Example

- Let us consider the system $-x^2 - xy - x - 1 \geq 0 \ \wedge \ y = 0.$
- Then $F_> = \{\}$, $F_\geq = \{-x^2 - xy - x - 1\}$, $F_= = \{y\}$.
- Set degree bound $d = 2$.
- $R \equiv (\alpha_x\, x + \alpha_y\, y + \alpha_0)\, y$, where α_x, α_y, α_0 are unknowns
- As $F_> = \{\}$, we have $\mathrm{Monoid}_d(F_>) = \mathrm{Monoid}(F_>) = \{1\}$
  $P \equiv \beta$, where β is an unknown constrained to $\beta > 0$
- $\mathrm{Monoid}_d(F_> \cup F_\geq) = \{1, -x^2 - xy - x - 1\}$.

  $$Q \equiv \underbrace{(\gamma_{x^2} x^2 + \gamma_{xy} xy + \gamma_{y^2} y^2 + \gamma_x x + \gamma_y y + \gamma_0)}_{\Gamma(x,y)} + \gamma_0'(-x^2 - xy - x - 1)$$

  where γ_* are unknowns s.t. $\Gamma(x, y)$ is a sum of squares, and $\gamma_0' \geq 0$
- Hence $P + Q + R$ is: $(\gamma_{x^2} - \gamma_0')x^2 + (\gamma_{xy} - \gamma_0')xy + \cdots$
  yielding equations $\gamma_{x^2} - \gamma_0' = \gamma_{xy} - \gamma_0' = \cdots = 0$

Example

- Let us consider the system $-x^2 - xy - x - 1 \geq 0 \ \wedge \ y = 0.$
- Then $F_> = \{\}$, $F_\geq = \{-x^2 - xy - x - 1\}$, $F_= = \{y\}$.
- Set degree bound $d = 2$.
- $R \equiv (\alpha_x\, x + \alpha_y\, y + \alpha_0)\, y$, where α_x, α_y, α_0 are unknowns
- As $F_> = \{\}$, we have $\mathrm{Monoid}_d(F_>) = \mathrm{Monoid}(F_>) = \{1\}$
  $P \equiv \beta$, where β is an unknown constrained to $\beta > 0$
- $\mathrm{Monoid}_d(F_> \cup F_\geq) = \{1, -x^2 - xy - x - 1\}$.

  $$Q \equiv \underbrace{(\gamma_{x^2}x^2 + \gamma_{xy}xy + \gamma_{y^2}y^2 + \gamma_x x + \gamma_y y + \gamma_0}_{\Gamma(x,y)}) + \gamma'_0(-x^2 - xy - x - 1)$$

  where γ_* are unknowns s.t. $\Gamma(x, y)$ is a sum of squares, and $\gamma'_0 \geq 0$
- Hence $P + Q + R$ is: $(\gamma_{x^2} - \gamma'_0)x^2 + (\gamma_{xy} - \gamma'_0)xy + \cdots$
  yielding equations $\gamma_{x^2} - \gamma'_0 = \gamma_{xy} - \gamma'_0 = \cdots = 0$
- A solution is $\Gamma(x, y) = (x + \frac{1}{2})^2, \gamma'_0 = 1, \alpha_x = 1, \beta = \frac{3}{4}$, rest $= 0$

Sums of Squares

- How can we solve the constraint: "polynomial P is a sum of squares"?
- In [Parrilo, MP'03]: semidefinite programming. Some disadvantages:
  - Some SDP algorithms can fail to converge if the problem is not strictly feasible ($=$ solution set is not full-dimensional).
    Some works [Monniaux, Corbineau, ITP'11] try to alleviate this problem
  - SDP algorithms use floating-point: postprocessing is needed!

Sums of Squares

- How can we solve the constraint: "polynomial P is a sum of squares"?
- In [Parrilo, MP'03]: semidefinite programming. Some disadvantages:
  - Some SDP algorithms can fail to converge if the problem is not strictly feasible ($=$ solution set is not full-dimensional).
    Some works [Monniaux, Corbineau, ITP'11] try to alleviate this problem
  - SDP algorithms use floating-point: postprocessing is needed!
- Let's use an SMT(NA) solver instead of an SDP solver!
- The basic idea in SDP techniques can be reused:
  polynomial $P \in \mathbb{Q}[x]$ is a sum of squares iff there exist a vector of monomials $\mu^T = (m_1, \ldots, m_k)$ over variables x, and a positive semidefinite matrix M with coefficients in $\mathbb{Q}$ such that $P = \mu^T M \mu$.
- Recall: a symmetric matrix $M \in \mathbb{R}^{N \times N}$ is *positive semidefinite* if for all $x \in \mathbb{R}^N$, we have $x^T M x \geq 0$.

Sums of Squares

- There exist several equivalent conditions that ensure that a symmetric matrix $M \in \mathbb{R}^{N \times N}$ is positive semidefinite:
  - Sylvester criterion: all principal minors are non-negative
  - Cholesky decomposition: there exists a lower triangular matrix L with non-negative diagonal coefficients such that $M = LL^T$
  - Gram matrix: there exists a matrix $R \in \mathbb{R}^{N \times N}$ such that $M = R^T R$

Sums of Squares

- There exist several equivalent conditions that ensure that a symmetric matrix $M \in \mathbb{R}^{N \times N}$ is positive semidefinite:
  - Sylvester criterion: all principal minors are non-negative
  - Cholesky decomposition: there exists a lower triangular matrix L with non-negative diagonal coefficients such that $M = LL^T$
  - Gram matrix: there exists a matrix $R \in \mathbb{R}^{N \times N}$ such that $M = R^T R$
- However, for efficiency reasons we opt for a light-weight approach:

  Instead of general sums of squares,
  we consider products of polys of the form $\sum_{i=1}^{n} q_i(x_i)$,
  where each $q_i(x_i)$ is a univariate non-negative polynomial of degree 2.
- A univariate polynomial $ax^2 + bx + c$ is non-negative if and only if $(a = 0 \ \wedge \ b = 0 \ \wedge \ c \geq 0) \ \ \vee \ \ (a > 0 \ \wedge \ b^2 - 4ac \leq 0)$
- In our experiments so far, we have been able to prove unsatisfiability for all problems (from our program analysis application) we tried

Filtering with Unsat Cores

- If the conjuntion of polynomial inequalities to be proved unsat is long, the resulting SMT problem can be huge, even with low degree bound

Filtering with Unsat Cores

- If the conjuntion of polynomial inequalities to be proved unsat is long, the resulting SMT problem can be huge, even with low degree bound
- Idea: to exploit failed attempts with the SAT-aimed approach
- Use unsat cores to heuristically select candidate relevant constraints
  - Let P be an (unsat) conjunction of polynomial inequalities
  - Let C be core obtained after ? iterations of [Borralleras et al., JAR'12]
  - Let $P' = P \cap C$ be the original inequalities that appear in the core
  - P' is a good candidate to be unsat

Filtering with Unsat Cores

- If the conjuntion of polynomial inequalities to be proved unsat is long, the resulting SMT problem can be huge, even with low degree bound
- Idea: to exploit failed attempts with the SAT-aimed approach
- Use unsat cores to heuristically select candidate relevant constraints
  - Let P be an (unsat) conjunction of polynomial inequalities
  - Let C be core obtained after ? iterations of [Borralleras et al., JAR'12]
  - Let $P' = P \cap C$ be the original inequalities that appear in the core
  - P' is a good candidate to be unsat
- In most cases, as far as we have experimented, this procedure:
  - does reduce significantly the size of the conjunction, and
  - does preserve unsatisfiability
- If unsatisfiability of P' fails, we can always try with original P

Open Questions and Future Work

- Any completeness result for the kind of problems under consideration?

Open Questions and Future Work

- Any completeness result for the kind of problems under consideration?
- Is there any way of simplifying even further the refutation template?

Open Questions and Future Work

- Any completeness result for the kind of problems under consideration?
- Is there any way of simplifying even further the refutation template?
- We sketched a theory solver for NA in a DPLL(T) framework. But:
  - Cheap way of making it incremental?
  - Explanations may not be minimal.
    Worth looking for minimal explanations (e.g., with Max-SMT)?

Open Questions and Future Work

- Any completeness result for the kind of problems under consideration?
- Is there any way of simplifying even further the refutation template?
- We sketched a theory solver for NA in a DPLL(T) framework. But:
  - Cheap way of making it incremental?
  - Explanations may not be minimal.
    Worth looking for minimal explanations (e.g., with Max-SMT)?
- We implemented a prototype in Prolog that,
  given a conjunction of polynomial inequalities,
  produces the SMT problem of finding a Positivstellensatz refutation.

  This is what we have used in the experiments referred here.

  Future work: full integration into an SMT(NA) system

Thank you!