# Weak Quantifier Elimination for the Integers
## Beyond the Linear Case

A. Lasaruk[1] T. Sturm[2]

FORWISS, University of Passau

FIM, University of Passau

September 17, 2007

# Quantifier Elimination (QE) by Virtual Substitution

- **Input:** First order formula $\exists x \varphi$
- **Output:** Quantifier-free formula $\varphi'$ with

$$\exists x \varphi \longleftrightarrow \varphi'$$

- **General idea:** Compute an elimination set $E$, such that

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} (\gamma \land \varphi[t /\!/ x])$$

- **For the reals and for the integers:** Elements of elimination sets are built essentially from interval boundaries

# Quantifier Elimination (QE) by Virtual Substitution

- **Input:** First order formula $\exists x \varphi$
- **Output:** Quantifier-free formula $\varphi'$ with

$$\exists x \varphi \longleftrightarrow \varphi'$$

- **General idea:** Compute an elimination set $E$, such that

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma, t) \in E} (\gamma \wedge \varphi[t /\!/ x])$$

- **For the reals and for the integers:** Elements of elimination sets are built essentially from interval boundaries

# Quantifier Elimination (QE) by Virtual Substitution

- **Input:** First order formula $\exists x \varphi$
- **Output:** Quantifier-free formula $\varphi'$ with

$$\exists x \varphi \longleftrightarrow \varphi'$$

- **General idea:** Compute an elimination set $E$, such that

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma, t) \in E} (\gamma \wedge \varphi[t /\!/ x])$$

- **For the reals and for the integers:** Elements of elimination sets are built essentially from interval boundaries

# Quantifier Elimination (QE) by Virtual Substitution

- **Input:** First order formula $\exists x \varphi$
- **Output:** Quantifier-free formula $\varphi'$ with

$$\exists x \varphi \longleftrightarrow \varphi'$$

- **General idea:** Compute an elimination set $E$, such that

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma, t) \in E} (\gamma \wedge \varphi[t /\!/ x])$$

- **For the reals and for the integers:** Elements of elimination sets are built essentially from interval boundaries

# Recall Real QE by Virtual Substitution

**Virtual substitution scheme:**

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} (\gamma \wedge \varphi[t /\!\!/ x])$$

- Consider: $\mathbb{R}$, arithmetic, ordering, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0)$$

- One possible QE result using $E = \{(\mathrm{true}, b/3)\}$:

$$\varphi \longleftrightarrow \bigvee_{t\in\{(\mathrm{true},b/3)\}} (3x - b = 0)[t /\!\!/ x] \longleftrightarrow 0 = 0 \longleftrightarrow \mathrm{true}.$$

- **Fact:** For linear formulas one can always find elimination sets.
- **Fact:** This can be extended to higher degrees to some extent.

# Recall Real QE by Virtual Substitution

**Virtual substitution scheme:**

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} (\gamma \wedge \varphi[t /\!/ x])$$

- Consider: $\mathbb{R}$, arithmetic, ordering, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0)$$

- One possible QE result using $E = \{(\mathrm{true}, b/3)\}$:

$$\varphi \longleftrightarrow \bigvee_{t\in\{(\mathrm{true},b/3)\}} (3x - b = 0)[t /\!/ x] \longleftrightarrow 0 = 0 \longleftrightarrow \mathrm{true}.$$

- **Fact:** For linear formulas one can always find elimination sets.
- **Fact:** This can be extended to higher degrees to some extent.

# Recall Real QE by Virtual Substitution

**Virtual substitution scheme:**

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} (\gamma \wedge \varphi[t /\!/ x])$$

- Consider: $\mathbb{R}$, arithmetic, ordering, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0)$$

- One possible QE result using $E = \{(\mathrm{true}, b/3)\}$:

$$\varphi \longleftrightarrow \bigvee_{t\in\{(\mathrm{true},b/3)\}} (3x - b = 0)[t /\!/ x] \longleftrightarrow 0 = 0 \longleftrightarrow \mathrm{true}.$$

- **Fact:** For linear formulas one can always find elimination sets.
- **Fact:** This can be extended to higher degrees to some extent.

# Recall Real QE by Virtual Substitution

**Virtual substitution scheme:**

$$\exists x\varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} (\gamma \wedge \varphi[t /\!/ x])$$

- Consider: $\mathbb{R}$, arithmetic, ordering, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0)$$

- One possible QE result using $E = \{(\mathrm{true}, b/3)\}$:

$$\varphi \longleftrightarrow \bigvee_{t\in\{(\mathrm{true},b/3)\}} (3x - b = 0)[t /\!/ x] \longleftrightarrow 0 = 0 \longleftrightarrow \mathrm{true}.$$

- **Fact:** For linear formulas one can always find elimination sets.
- **Fact:** This can be extended to higher degrees to some extent.

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \wedge (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \wedge k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x (3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \land (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \land k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \wedge (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \wedge k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \land (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \land k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \wedge (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \wedge k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \land (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \land k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# The Same Problem Over the Integers

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(3x - b = 0).$$

- One possible QE result:

$$\varphi \longleftrightarrow \bigvee_{k=-3}^{3} \left( b + k \equiv_3 0 \land (3x - b = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

$$\longleftrightarrow \bigvee_{k=-3}^{3} (b + k \equiv_3 0 \land k = 0) \longleftrightarrow b \equiv_3 0.$$

- Presburger Arithmetic: No multiplication (coefficients are integers, $3x$ is short for $x + x + x$) **[Presburger 1929]**
- **Observation:** QE in the virtual substitution framework: $E = \{(b + k \equiv_3 0, (b + k)/3) \mid |k| \le 3\}$.
- **Observation:** Systematically occurring formal $\bigvee$-notation decreases complexity **[Weispfenning 1990]**

# Introducing Parameters Into Presburger Arithmetic

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x (a \cdot x - b = 0)$$

- Trying the same technique:

$$\varphi \longleftrightarrow b = 0 \lor \bigvee_{k=-a}^{a} \left( a \neq 0 \land b + k \equiv_a 0 \land (ax - b = 0) \left[ \frac{b+k}{a} /\!/ x \right] \right)$$

$$\longleftrightarrow b = 0 \lor \bigvee_{k=-a}^{a} (a \neq 0 \land b + k \equiv_a 0 \land k = 0) \longleftrightarrow b \equiv_a 0.$$

- **Problem:** $\bigvee_{k=-a}^{a} \left( a \neq 0 \land b + k \equiv_a 0 \land (ax - b = 0) \left[ \frac{b+k}{a} /\!/ x \right] \right)$ is not a first-order formula!

# Introducing Parameters Into Presburger Arithmetic

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(a \cdot x - b = 0)$$

- Trying the same technique:

$$\varphi \longleftrightarrow b = 0 \vee \bigvee_{k=-a}^{a} \left( a \neq 0 \wedge b + k \equiv_a 0 \wedge (ax - b = 0) \left[ \frac{b+k}{a} /\!/ x \right] \right)$$

$$\longleftrightarrow b = 0 \vee \bigvee_{k=-a}^{a} (a \neq 0 \wedge b + k \equiv_a 0 \wedge k = 0) \longleftrightarrow b \equiv_a 0.$$

- **Problem:** $\bigvee_{k=-a}^{a} \left( a \neq 0 \wedge b + k \equiv_a 0 \wedge (ax - b = 0) \left[ \frac{b+k}{a} /\!/ x \right] \right)$ is not a first-order formula!

# Introducing Parameters Into Presburger Arithmetic

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(a \cdot x - b = 0)$$

- Trying the same technique:

$$\begin{aligned} \varphi \quad \longleftrightarrow \quad & b = 0 \lor \\ & \bigvee_{k=-a}^{a} \left( a \neq 0 \land b + k \equiv_a 0 \land (ax - b = 0) \left[ \frac{b+k}{a} /\!/ x \right] \right) \\ \longleftrightarrow \quad & b = 0 \lor \bigvee_{k=-a}^{a} (a \neq 0 \land b + k \equiv_a 0 \land k = 0) \longleftrightarrow b \equiv_a 0. \end{aligned}$$

- **Problem:** $\bigvee_{k=-a}^{a} \left( a \neq 0 \land b + k \equiv_a 0 \land (ax - b = 0) \left[ \frac{b+k}{a} /\!/ x \right] \right)$ is not a first-order formula!

# Introducing Parameters Into Presburger Arithmetic

- Consider: $\mathbb{Z}$, arithmetic, ordering, *congruences*, Boolean combination, first-order quantification

$$\varphi = \exists x(a \cdot x - b = 0)$$

- Trying the same technique:

$$\begin{aligned}\varphi \quad \longleftrightarrow \quad & b = 0 \vee \\ & \bigvee_{k=-a}^{a} \left(a \neq 0 \wedge b + k \equiv_a 0 \wedge (ax - b = 0)\left[\frac{b+k}{a} /\!/ x\right]\right) \\ \longleftrightarrow \quad & b = 0 \vee \bigvee_{k=-a}^{a} (a \neq 0 \wedge b + k \equiv_a 0 \wedge k = 0) \longleftrightarrow b \equiv_a 0.\end{aligned}$$

- **Problem:** $\bigvee_{k=-a}^{a} \left(a \neq 0 \wedge b + k \equiv_a 0 \wedge (ax - b = 0)\left[\frac{b+k}{a} /\!/ x\right]\right)$ is not a first-order formula!

# Introducing Bounded Quantifiers

- Formal extension of logic by new quantifiers with the semantics:

$$\bigsqcup_{k:\,\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\,\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi).$$

- **Bounded quantifiers:** Range $\beta$ is finite for all choices of parameters
- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} \mid \beta(z)\}} \varphi[i/k]$
- **Questionable formula:**

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \wedge b + k \equiv_3 0 \wedge (ax - y = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

- **Weak quantifier elimination:** Results contain bounded quantifiers
- **Fact:** The discussed framework is sufficient for linear weak QE with polynomial coefficients **[L. and S. AAECC 2007]**.

# Introducing Bounded Quantifiers

- Formal extension of logic by new quantifiers with the semantics:

$$\bigsqcup_{k:\,\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\,\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi).$$

- **Bounded quantifiers:** Range $\beta$ is finite for all choices of parameters
- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} \mid \beta(z)\}} \varphi[i/k]$
- **Questionable formula:**

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \wedge b + k \equiv_3 0 \wedge (ax - y = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

- **Weak quantifier elimination:** Results contain bounded quantifiers
- **Fact:** The discussed framework is sufficient for linear weak QE with polynomial coefficients **[L. and S. AAECC 2007]**.

# Introducing Bounded Quantifiers

- Formal extension of logic by new quantifiers with the semantics:

$$\bigsqcup_{k:\,\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\,\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi).$$

- **Bounded quantifiers:** Range $\beta$ is finite for all choices of parameters
- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} | \beta(z)\}} \varphi[i/k]$
- **Questionable formula:**

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \wedge b + k \equiv_3 0 \wedge (ax - y = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

- **Weak quantifier elimination:** Results contain bounded quantifiers
- **Fact:** The discussed framework is sufficient for linear weak QE with polynomial coefficients **[L. and S. AAECC 2007]**.

# Introducing Bounded Quantifiers

- Formal extension of logic by new quantifiers with the semantics:

$$\bigsqcup_{k:\,\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\,\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi).$$

- **Bounded quantifiers:** Range $\beta$ is finite for all choices of parameters
- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} \mid \beta(z)\}} \varphi[i/k]$
- **Questionable formula:**

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \wedge b + k \equiv_3 0 \wedge (ax - y = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

- **Weak quantifier elimination:** Results contain bounded quantifiers
- **Fact:** The discussed framework is sufficient for linear weak QE with polynomial coefficients **[L. and S. AAECC 2007]**.

# Introducing Bounded Quantifiers

- Formal extension of logic by new quantifiers with the semantics:

$$\bigsqcup_{k:\,\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\,\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi).$$

- **Bounded quantifiers:** Range $\beta$ is finite for all choices of parameters
- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} \mid \beta(z)\}} \varphi[i/k]$
- **Questionable formula:**

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \wedge b + k \equiv_3 0 \wedge (ax - y = 0) \left[ \frac{b+k}{3} /\!\!/ x \right] \right)$$

- **Weak quantifier elimination:** Results contain bounded quantifiers
- **Fact:** The discussed framework is sufficient for linear weak QE with polynomial coefficients **[L. and S. AAECC 2007]**.

# Introducing Bounded Quantifiers

- Formal extension of logic by new quantifiers with the semantics:

$$\bigsqcup_{k:\,\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\,\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi).$$

- **Bounded quantifiers:** Range $\beta$ is finite for all choices of parameters
- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} \mid \beta(z)\}} \varphi[i/k]$
- **Questionable formula:**

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \wedge b + k \equiv_3 0 \wedge (ax - y = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$

- **Weak quantifier elimination:** Results contain bounded quantifiers
- **Fact:** The discussed framework is sufficient for linear weak QE with polynomial coefficients **[L. and S. AAECC 2007].**

# Towards Higher Degrees

- Is our extension of logic suitable even for nonlinear formulas?
- **Yes**, for certain ones!

**Example**

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$$

**Output:** $\varphi$ is equivalent to

$$\bigsqcup_{k:\,|k| \leq |a|} (a \neq 0 \wedge y + k \equiv_a 0 \wedge k < 0 \wedge |ay + ak| > |a|^3 + 2a^2) \vee \bigsqcup_{k:\,|k| \leq |a|+2} (ak - y < 0 \wedge k^2 + k + a > 0).$$

# Towards Higher Degrees

- Is our extension of logic suitable even for nonlinear formulas?
- **Yes**, for certain ones!

**Example**

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$$

**Output:** $\varphi$ is equivalent to

$$\bigsqcup_{k\colon |k| \leq |a|} (a \neq 0 \wedge y + k \equiv_a 0 \wedge k < 0 \wedge |ay + ak| > |a|^3 + 2a^2) \vee$$

$$\bigsqcup_{k\colon |k| \leq |a|+2} (ak - y < 0 \wedge k^2 + k + a > 0).$$

# Towards Higher Degrees

- Is our extension of logic suitable even for nonlinear formulas?
- **Yes**, for certain ones!

## Example

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$$

**Output:** $\varphi$ is equivalent to

$$\bigsqcup_{k\colon |k| \leq |a|} \left(a \neq 0 \wedge y + k \equiv_a 0 \wedge k < 0 \wedge |ay + ak| > |a|^3 + 2a^2\right) \vee$$

$$\bigsqcup_{k\colon |k| \leq |a|+2} \left(ak - y < 0 \wedge k^2 + k + a > 0\right).$$

# Towards Higher Degrees

- Is our extension of logic suitable even for nonlinear formulas?
- **Yes**, for certain ones!

**Example**

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \land x^2 + x + a > 0)$$

**Output:** $\varphi$ is equivalent to

$$\bigsqcup_{k:\,|k|\leq|a|} \big(a \neq 0 \land y + k \equiv_a 0 \land k < 0 \land |ay + ak| > |a|^3 + 2a^2\big) \lor \bigsqcup_{k:\,|k|\leq|a|+2} \big(ak - y < 0 \land k^2 + k + a > 0\big).$$

# Towards Higher Degrees

- Is our extension of logic suitable even for nonlinear formulas?
- **Yes**, for certain ones!

**Example**

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \land x^2 + x + a > 0)$$

**Output:** $\varphi$ is equivalent to

$$\bigvee_{k=-10}^{10} \big(y + k \equiv_{10} 0 \land k < 0 \land |y + k| > 120\big) \lor \bigvee_{k=-12}^{12} \big(10k - y < 0 \land k^2 + k + 10 > 0\big).$$

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

- **(i)** linear in the quantified variables or
- **(ii)** superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

- **(i)** linear in the quantified variables or
- **(ii)** superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

(i) linear in the quantified variables or

(ii) superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

- **(i)** linear in the quantified variables or
- **(ii)** superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

- **(i)** linear in the quantified variables or
- **(ii)** superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

- **(i)** linear in the quantified variables or
- **(ii)** superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Formulas We Can Handle

We are able to eliminate all the regular quantifiers from formulas $\varphi$ specified as follows:

**Univariately nonlinear formulas:**

**($U_1$)** None of the quantified variables occurs within moduli of congruences or incongruences.

**($U_2$)** Congruences are linear in the quantified variables.

**($U_3$)** Equations and inequalities are either

- **(i)** linear in the quantified variables or
- **(ii)** superlinear univariate in one of the quantified variables.

**Consequences:**

- Linear formulas are just special univariately nonlinear formulas
- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

# Explanation of Notions

- Equations, inequalities, congruences (w.r.t. $x$ and $y$)
  - **Linear:**

$$ax - y < 0, \quad ax - y \equiv_m 0$$

  - **Superlinear univariate:** $x^2 + x + a > 0$
  - **Neither linear nor superlinear univariate:**

$$x^2 + xy + y^2 > 0, \quad x^2 + y^2 + a > 0$$

- Formulas
  - **Linear:** $\forall a \forall b (a < b \longrightarrow \exists z (a < z \wedge z < b))$
  - **Univariately nonlinear:**

$$\forall y \exists x (ax - y < 0 \wedge x^2 + x + a > 0)$$

  - **Neither linear nor univariately nonlinear:**

$$\exists x \exists y \exists z (x^5 + y^5 = z^5)$$

# Explanation of Notions

- Equations, inequalities, congruences (w.r.t. $x$ and $y$)
  - **Linear:**

$$ax - y < 0, \quad ax - y \equiv_m 0$$

  - **Superlinear univariate:** $x^2 + x + a > 0$
  - **Neither linear nor superlinear univariate:**

$$x^2 + xy + y^2 > 0, \quad x^2 + y^2 + a > 0$$

- Formulas
  - **Linear:** $\forall a \forall b(a < b \longrightarrow \exists z(a < z \wedge z < b))$
  - **Univariately nonlinear:**

$$\forall y \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$$

  - **Neither linear nor univariately nonlinear:**

$$\exists x \exists y \exists z(x^5 + y^5 = z^5)$$

# Explanation of Notions

- Equations, inequalities, congruences (w.r.t. *x* and *y*)
  - **Linear:**

$$ax - y < 0, \quad ax - y \equiv_m 0$$

  - **Superlinear univariate:** $x^2 + x + a > 0$
  - **Neither linear nor superlinear univariate:**

$$x^2 + xy + y^2 > 0, \quad x^2 + y^2 + a > 0$$

- Formulas
  - **Linear:** $\forall a \forall b (a < b \longrightarrow \exists z (a < z \land z < b))$
  - **Univariately nonlinear:**

$$\forall y \exists x (ax - y < 0 \land x^2 + x + a > 0)$$

  - **Neither linear nor univariately nonlinear:**

$$\exists x \exists y \exists z (x^5 + y^5 = z^5)$$

# Explanation of Notions

- Equations, inequalities, congruences (w.r.t. $x$ and $y$)
  - **Linear:**
$$ax - y < 0, \quad ax - y \equiv_m 0$$
  - **Superlinear univariate:** $x^2 + x + a > 0$
  - **Neither linear nor superlinear univariate:**
$$x^2 + xy + y^2 > 0, \quad x^2 + y^2 + a > 0$$
- Formulas
  - **Linear:** $\forall a \forall b (a < b \longrightarrow \exists z (a < z \wedge z < b))$
  - **Univariately nonlinear:**
$$\forall y \exists x (ax - y < 0 \wedge x^2 + x + a > 0)$$
  - **Neither linear nor univariately nonlinear:**
$$\exists x \exists y \exists z (x^5 + y^5 = z^5)$$

# Explanation of Notions

- Equations, inequalities, congruences (w.r.t. $x$ and $y$)
  - **Linear:**
    $$ax - y < 0, \quad ax - y \equiv_m 0$$
  - **Superlinear univariate:** $x^2 + x + a > 0$
  - **Neither linear nor superlinear univariate:**
    $$x^2 + xy + y^2 > 0, \quad x^2 + y^2 + a > 0$$
- Formulas
  - **Linear:** $\forall a \forall b (a < b \longrightarrow \exists z (a < z \wedge z < b))$
  - **Univariately nonlinear:**
    $$\forall y \exists x (ax - y < 0 \wedge x^2 + x + a > 0)$$
  - **Neither linear nor univariately nonlinear:**
    $$\exists x \exists y \exists z (x^5 + y^5 = z^5)$$

# Explanation of Notions

- Equations, inequalities, congruences (w.r.t. $x$ and $y$)
  - **Linear:**
$$ax - y < 0, \quad ax - y \equiv_m 0$$
  - **Superlinear univariate:** $x^2 + x + a > 0$
  - **Neither linear nor superlinear univariate:**
$$x^2 + xy + y^2 > 0, \quad x^2 + y^2 + a > 0$$
- Formulas
  - **Linear:** $\forall a \forall b (a < b \longrightarrow \exists z (a < z \wedge z < b))$
  - **Univariately nonlinear:**
$$\forall y \exists x (ax - y < 0 \wedge x^2 + x + a > 0)$$
  - **Neither linear nor univariately nonlinear:**
$$\exists x \exists y \exists z (x^5 + y^5 = z^5)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \leq b)[\frac{b'}{a'}//x] := (aa'b' \leq a'^2 b), \quad (ax \equiv_m b)[\frac{b'}{a'}//x] := (ab' \equiv_{ma'} a'b)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/ congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \le b)[\frac{b'}{a'}//x] := (aa'b' \le a'^2b), \quad (ax \equiv_m b)[\frac{b'}{a'}//x] := (ab' \equiv_{ma'} a'b)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \leq b)\left[\frac{b'}{a'}//x\right] := (aa'b' \leq a'^2 b), \quad (ax \equiv_m b)\left[\frac{b'}{a'}//x\right] := (ab' \equiv_{ma'} a'b)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \leq b)[\frac{b'}{a'}//x] := (aa'b' \leq a'^2b), \quad (ax \equiv_m b)[\frac{b'}{a'}//x] := (ab' \equiv_{ma'} a'b)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/ congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \leq b)[\frac{b'}{a'}//x] := (aa'b' \leq a'^2 b), \quad (ax \equiv_m b)[\frac{b'}{a'}//x] := (ab' \equiv_{ma'} a'b)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \leq b)[\frac{b'}{a'}//x] := (aa'b' \leq a'^2 b), \quad (ax \equiv_m b)[\frac{b'}{a'}//x] := (ab' \equiv_{ma'} a'b)$$

# Basic Technical Ideas

- **Test points** depend on on the equation/inequality/congruence, which has generated the test point:
  - Known test points for the linear case **[L. and S. 2007]**
  - Terms consisting only of one variable and Cauchy bounds as ranges for the superlinear univariate case
- **Virtual substitution** depends on the equation/inequality/ congruence, which the test point is substituted into
  - Regular virtual substitution methods for the linear case
  - **Constrained virtual substitution** for the superlinear univariate case

**Remider: Regular virtual substitution**

$$(ax \leq b)\left[\frac{b'}{a'} /\!/ x\right] := (aa'b' \leq a'^2 b), \quad (ax \equiv_m b)\left[\frac{b'}{a'} /\!/ x\right] := (ab' \equiv_{ma'} a'b)$$

# Parametric Elimination Sets

$$E = \{\, (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \leq i \leq n \,\}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \leq j \leq m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Parametric Elimination Sets

$$E = \{\, (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \le i \le n \,\}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \le j \le m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \dots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Parametric Elimination Sets

$$E = \{\, (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \le i \le n \,\}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \le j \le m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Parametric Elimination Sets

$$E = \{ (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \leq i \leq n \}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \leq j \leq m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \dots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Parametric Elimination Sets

$$E = \{\, (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \leq i \leq n \,\}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \leq j \leq m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Parametric Elimination Sets

$$E = \{\, (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \le i \le n \,\}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \le j \le m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \dots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Parametric Elimination Sets

$$E = \{\, (\gamma_i, t_i, \sigma_i, B_i) \mid 1 \leq i \leq n \,\}, \quad B_i = ((k_{ij}, \beta_{ij}) \mid 1 \leq j \leq m_i)$$

- Substitution procedure $\sigma_i$
- Ranges of bounded quantifiers $B_i$
- Elimination result:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, B_i) \in E} \ \bigsqcup_{k_{i1} : \beta_{i1}} \dots \bigsqcup_{k_{im_i} : \beta_{im_i}} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

- **Notice:** Substitute $t_i$ into $\varphi$ applying $\sigma_i$ to each equality, inequality and congruence in $\varphi$

# Example

Consider $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$
**Result**:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, (k, \beta)) \in E} \ \bigsqcup_{k:\,\beta} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \frac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big),$$
$$\big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

Consider the first entry of $E$:

- The pseudo-term $\frac{y+k}{a}$ describes a finite set of points around the solution of $ax - y = 0$ using the range $|k| \leq |a|$.
- The guard $a \neq 0 \wedge y + k \equiv_a 0$ ensures that $\frac{y+k}{a}$ evaluates to an integer.
- $[\cdot /\!/ \cdot]$ is our constrained virtual substitution.

# Example

Consider $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$

**Result**:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, (k, \beta)) \in E} \; \bigsqcup_{k:\,\beta} \left(\gamma_i \wedge \sigma_i(\varphi, t_i, x)\right)$$

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \tfrac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big), \quad \big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

Consider the first entry of $E$:

- The pseudo-term $\frac{y+k}{a}$ describes a finite set of points around the solution of $ax - y = 0$ using the range $|k| \leq |a|$.
- The guard $a \neq 0 \wedge y + k \equiv_a 0$ ensures that $\frac{y+k}{a}$ evaluates to an integer.
- $[\cdot /\!/ \cdot]$ is our constrained virtual substitution.

# Example

Consider $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$
**Result**:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, (k, \beta)) \in E} \bigsqcup_{k:\,\beta} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \tfrac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big), \big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

Consider the first entry of $E$:

- The pseudo-term $\frac{y+k}{a}$ describes a finite set of points around the solution of $ax - y = 0$ using the range $|k| \leq |a|$.
- The guard $a \neq 0 \wedge y + k \equiv_a 0$ ensures that $\frac{y+k}{a}$ evaluates to an integer.
- $[\cdot /\!/ \cdot]$ is our constrained virtual substitution.

# Example

Consider $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$
**Result**:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, (k, \beta)) \in E} \bigsqcup_{k \colon \beta} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \frac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big), \\ \big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

Consider the first entry of $E$:

- The pseudo-term $\frac{y+k}{a}$ describes a finite set of points around the solution of $ax - y = 0$ using the range $|k| \leq |a|$.
- The guard $a \neq 0 \wedge y + k \equiv_a 0$ ensures that $\frac{y+k}{a}$ evaluates to an integer.
- $[\cdot /\!/ \cdot]$ is our constrained virtual substitution.

# Example

Consider $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$
**Result**:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, \sigma_i, (k, \beta)) \in E} \bigsqcup_{k:\, \beta} (\gamma_i \wedge \sigma_i(\varphi, t_i, x))$$

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \frac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big), \big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

Consider the first entry of $E$:

- The pseudo-term $\frac{y+k}{a}$ describes a finite set of points around the solution of $ax - y = 0$ using the range $|k| \leq |a|$.
- The guard $a \neq 0 \wedge y + k \equiv_a 0$ ensures that $\frac{y+k}{a}$ evaluates to an integer.
- $[\cdot /\!/ \cdot]$ is our constrained virtual substitution.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0) \left[\frac{y+k}{a} /\!\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0) \left[\frac{y + k}{a} /\!\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example of Constrained Virtual Substitution

**Problem:** How do we define $(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right]$?

- Naive formal substitution yields $(y + k)^2 + a(y + k) + a^3 > 0$. This is neither linear nor superlinear univariate wrt. $y$ and $k$.
- We define the (constrained virtual) substitution as follows:

$$(x^2 + x + a > 0)\left[\frac{y+k}{a} /\!/ x\right] := |ay + ak| > |a|^3 + 2a^2.$$

- Division of $|ay + ak| > |a|^3 + 2a^2$ by $a^2$ yields $|\frac{y+k}{a}| > |a| + 2$
- $|a| + 2$ is the Cauchy bound plus 1 of $x^2 + x + a$.
- **Intuitive idea:** State that the test term $\frac{y+k}{a}$ lies **outside** the Cauchy-bounds of $x^2 + x + a$ and thus satisfies $x^2 + x + a > 0$.
- **Warning:** For the possible case that $\frac{y+k}{a}$ lies in fact **within** the Cauchy bounds but still satisfies $x^2 + x + a > 0$ there is something left to do.

# Example

Consider once more $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$

$$E = \{(a \neq 0 \wedge y + k \equiv_a 0,\ \frac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))),\ (\text{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2)))\}.$$

Consider the second entry of $E$:

- $k$ represents each value inside the Cauchy bound of $x^2 + x + a$.
- $|k| \leq |a| + 2$ is the range of a bounded quantifier that substituting $k$ within its scope exactly covers every single point within the Cauchy bounds of $x^2 + x + a$.
- The substitution $[\cdot / \cdot]$ is the regular substitution of terms for variables.

# Example

Consider once more $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$

$$E = \{ (a \neq 0 \wedge y + k \equiv_a 0, \frac{y+k}{a}, [\cdot /\!/ \cdot], ((k, |k| \leq |a|))), \\ (\mathrm{true}, k, [\cdot / \cdot], ((k, |k| \leq |a| + 2))) \}.$$

Consider the second entry of $E$:

- $k$ represents each value inside the Cauchy bound of $x^2 + x + a$.
- $|k| \leq |a| + 2$ is the range of a bounded quantifier that substituting $k$ within its scope exactly covers every single point within the Cauchy bounds of $x^2 + x + a$.
- The substitution $[\cdot / \cdot]$ is the regular substitution of terms for variables.

# Example

Consider once more $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$

$$E = \{ (a \neq 0 \wedge y + k \equiv_a 0, \frac{y+k}{a}, [\cdot /\!/ \cdot], ((k, |k| \leq |a|))), \\ (\text{true}, k, [\cdot / \cdot], ((k, |k| \leq |a| + 2))) \}.$$

Consider the second entry of $E$:

- $k$ represents each value inside the Cauchy bound of $x^2 + x + a$.
- $|k| \leq |a| + 2$ is the range of a bounded quantifier that substituting $k$ within its scope exactly covers every single point within the Cauchy bounds of $x^2 + x + a$.
- The substitution $[\cdot / \cdot]$ is the regular substitution of terms for variables.

# Example

Consider once more $\exists x \varphi$ with $\varphi = ax - y < 0 \wedge x^2 + x + a > 0$

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \tfrac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big),$$
$$\big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}.$$

Consider the second entry of $E$:

- $k$ represents each value inside the Cauchy bound of $x^2 + x + a$.
- $|k| \leq |a| + 2$ is the range of a bounded quantifier that substituting $k$ within its scope exactly covers every single point within the Cauchy bounds of $x^2 + x + a$.
- The substitution $[\cdot / \cdot]$ is the regular substitution of terms for variables.

# Towards Higher Degrees

## Example

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$$

**Elimination set:**

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \tfrac{y+k}{a},\ [\cdot /\!/ \cdot],\ ((k, |k| \leq |a|))\big),\ \big(\mathrm{true},\ k,\ [\cdot / \cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

**Output:** $\varphi$ is equivalent to

$$\bigsqcup_{k:\,|k| \leq |a|} (a \neq 0 \wedge y + k \equiv_a 0 \wedge k < 0 \wedge |ay + ak| > |a|^3 + 2a^2) \vee \bigsqcup_{k:\,|k| \leq |a|+2} (ak - y < 0 \wedge k^2 + k + a > 0).$$

# Towards Higher Degrees

## Example

**Input:** Eliminate $\exists x$ from

$$\varphi = \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$$

**Elimination set:**

$$E = \Big\{ \big(a \neq 0 \wedge y + k \equiv_a 0,\ \tfrac{y+k}{a},\ [\cdot/\!\!/\cdot],\ ((k, |k| \leq |a|))\big), \\ \big(\mathrm{true},\ k,\ [\cdot/\cdot],\ ((k, |k| \leq |a| + 2))\big) \Big\}$$

**Output:** $\varphi$ is equivalent to

$$\bigsqcup_{k\colon |k| \leq |a|} \big(a \neq 0 \wedge y + k \equiv_a 0 \wedge k < 0 \wedge |ay + ak| > |a|^3 + 2a^2\big) \vee \\ \bigsqcup_{k\colon |k| \leq |a|+2} \big(ak - y < 0 \wedge k^2 + k + a > 0\big).$$

# The Main Result of This Talk

**Theorem (Elimination Theorem)**

*The ordered ring of the integers with congruences admits weak quantifier elimination for univariately nonlinear formulas.*

**Corollary (Decidability of Sentences)**

*In the ordered ring of the integers with congruences univariately nonlinear sentences are decidable.*

**Notice:** For regular first-order decision framework no bounded quantifiers come to existence!

# The Main Result of This Talk

**Theorem (Elimination Theorem)**

*The ordered ring of the integers with congruences admits weak quantifier elimination for univariately nonlinear formulas.*

**Corollary (Decidability of Sentences)**

*In the ordered ring of the integers with congruences univariately nonlinear sentences are decidable.*

**Notice:** For regular first-order decision framework no bounded quantifiers come to existence!

# The Main Result of This Talk

**Theorem (Elimination Theorem)**

*The ordered ring of the integers with congruences admits weak quantifier elimination for univariately nonlinear formulas.*

**Corollary (Decidability of Sentences)**

*In the ordered ring of the integers with congruences univariately nonlinear sentences are decidable.*

**Notice:** For regular first-order decision framework no bounded quantifiers come to existence!

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# What is REDLOG

**Implementation:** Our methods are implemented in REDLOG and are publicly available !

- REDUCE logic system
- Component of the computer algebra system REDUCE
- Continuous development since 1992
- REDLOG 3.0 is part of REDUCE 3.8
- Current version is freely distributed on the web (e.g. 3.070127)
- Currently 30 kloc (LISP)

**REDLOG homepage**

www.redlog.eu

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. $p$-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# REDLOG Domains

BOOLEAN Quantified propositional calculus **[CASC 2003]**

COMPLEX The class of algebraically closed fields (e.g. complex numbers over the language of rings)

DIFFERENTIAL Differentially closed fields **[CASC 2004]**

PADICS Discretely valued fields (e.g. *p*-adic numbers)

QUEUES Two-sided queues with elements of some basic type

REALS The class of real closed fields (e.g. the real numbers with ordering)

TERMS Free Malcev-type term algebras **[CASC 2002]**

**Work discussed here:**

INTEGERS Originally introduced for the full linear theory of the integers **[Weispfenning 1990], [L. and S. 2007]**

Natural extension to univariately nonlinear formulas without loosing any of its previous features

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Computation Examples

**Application domains include the following:**

- Nonlinear discrete optimization problems
- Integer linear optimization with superlinear univariate constraints
- Software security
- Automatic code verification of programs with superlinear univariate expressions
- Automatic loop parallelization
- Scheduling problems

All our computations discussed in the following have been performed on a 1.66 GHz Intel Core 2 Duo processor T5500 using only one core and 128 MB RAM.

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1\ \varrho_1\ 0, \quad \ldots, \quad p_r\ \varrho_r\ 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=, \neq, \leq, >, \geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \wedge \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \wedge \bigwedge_{s=1}^{r} p_s\ \varrho_s\ 0 \Big)$$

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1 \ \varrho_1 \ 0, \quad \ldots, \quad p_r \ \varrho_r \ 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=$, $\neq$, $\leq$, $>$, $\geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \wedge \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \wedge \bigwedge_{s=1}^{r} p_s \ \varrho_s \ 0 \Big)$$

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1 \; \varrho_1 \; 0, \quad \ldots, \quad p_r \; \varrho_r \; 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=, \neq, \leq, >, \geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \wedge \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \wedge \bigwedge_{s=1}^{r} p_s \; \varrho_s \; 0 \Big)$$

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1 \ \varrho_1 \ 0, \quad \ldots, \quad p_r \ \varrho_r \ 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=$, $\neq$, $\leq$, $>$, $\geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \land \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \land \bigwedge_{s=1}^{r} p_s \ \varrho_s \ 0 \Big)$$

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1 \ \varrho_1 \ 0, \quad \ldots, \quad p_r \ \varrho_r \ 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=, \neq, \leq, >, \geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \wedge \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \wedge \bigwedge_{s=1}^{r} p_s \ \varrho_s \ 0 \Big)$$

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1 \ \varrho_1 \ 0, \quad \ldots, \quad p_r \ \varrho_r \ 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=$, $\neq$, $\leq$, $>$, $\geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \wedge \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \wedge \bigwedge_{s=1}^{r} p_s \ \varrho_s \ 0 \Big)$$

# Optimization

A *parametric linear optimization problem with univariately nonlinear constraints*: Minimize a cost function $\gamma_1 x_1 + \cdots + \gamma_n x_n$ subject to

$$A\mathbf{x} \geq \mathbf{b}, \quad p_1 \ \varrho_1 \ 0, \quad \ldots, \quad p_r \ \varrho_r \ 0.$$

- $A = (\alpha_{ij})$ is an $m \times n$-matrix, and $\mathbf{b} = (\beta_1, \ldots, \beta_m)$ is an $m$-vector.
- All these coefficients $\alpha_{ij}$, $\beta_i$, and $\gamma_j$ are possibly parametric.
- The $p_1, \ldots, p_r$ are parametric univariate polynomials.
- Each corresponding $\varrho_s$ is one of $=$, $\neq$, $\leq$, $>$, $\geq$, or $<$.

**Formulation within our framework**

Let $z$ be a new variable.

$$\exists x_1 \ldots \exists x_n \Big( \sum_{j=1}^{n} \gamma_j x_j \leq z \wedge \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j \geq \beta_i \wedge \bigwedge_{s=1}^{r} p_s \ \varrho_s \ 0 \Big)$$

# Optimization Example

Minimize $x + y$ subject to the following constraints:

$$x \geq 0, \quad y \geq 0, \quad x - y \geq 0, \quad \text{and} \quad x^2 - a < 0.$$

Formulation as a quantifier elimination problem:

$$\exists x \exists y (x + y \leq z \wedge x \geq 0 \wedge y \geq 0 \wedge x - y \geq 0 \wedge x^2 - a < 0).$$

**Results:**

- Within 20 ms a weakly quantifier-free equivalent containing 26 atomic formulas
- Setting $a = 10$ and automatically simplifying yields within 2980 ms the result $z > 8$, i.e., the minimum for $x + y$ is 4.
- If we plug in $a = 10$ before the elimination, then we directly obtain $z > 3$ in only 780 ms.

# Optimization Example

Minimize $x + y$ subject to the following constraints:

$$x \geq 0, \quad y \geq 0, \quad x - y \geq 0, \quad \text{and} \quad x^2 - a < 0.$$

Formulation as a quantifier elimination problem:

$$\exists x \exists y \big(x + y \leq z \wedge x \geq 0 \wedge y \geq 0 \wedge x - y \geq 0 \wedge x^2 - a < 0\big).$$

**Results:**

- Within 20 ms a weakly quantifier-free equivalent containing 26 atomic formulas
- Setting $a = 10$ and automatically simplifying yields within 2980 ms the result $z > 8$, i.e., the minimum for $x + y$ is 4.
- If we plug in $a = 10$ before the elimination, then we directly obtain $z > 3$ in only 780 ms.

# Optimization Example

Minimize $x + y$ subject to the following constraints:

$$x \geq 0, \quad y \geq 0, \quad x - y \geq 0, \quad \text{and} \quad x^2 - a < 0.$$

Formulation as a quantifier elimination problem:

$$\exists x \exists y \big( x + y \leq z \wedge x \geq 0 \wedge y \geq 0 \wedge x - y \geq 0 \wedge x^2 - a < 0 \big).$$

**Results:**

- Within 20 ms a weakly quantifier-free equivalent containing 26 atomic formulas
- Setting $a = 10$ and automatically simplifying yields within 2980 ms the result $z > 8$, i.e., the minimum for $x + y$ is 4.
- If we plug in $a = 10$ before the elimination, then we directly obtain $z > 3$ in only 780 ms.

# Optimization Example

Minimize $x + y$ subject to the following constraints:

$$x \geq 0, \quad y \geq 0, \quad x - y \geq 0, \quad \text{and} \quad x^2 - a < 0.$$

Formulation as a quantifier elimination problem:

$$\exists x \exists y \big(x + y \leq z \wedge x \geq 0 \wedge y \geq 0 \wedge x - y \geq 0 \wedge x^2 - a < 0\big).$$

**Results:**

- Within 20 ms a weakly quantifier-free equivalent containing 26 atomic formulas
- Setting $a = 10$ and automatically simplifying yields within 2980 ms the result $z > 8$, i.e., the minimum for $x + y$ is 4.
- If we plug in $a = 10$ before the elimination, then we directly obtain $z > 3$ in only 780 ms.

# Optimization Example

Minimize $x + y$ subject to the following constraints:

$$x \geq 0, \quad y \geq 0, \quad x - y \geq 0, \quad \text{and} \quad x^2 - a < 0.$$

Formulation as a quantifier elimination problem:

$$\exists x \exists y \big(x + y \leq z \wedge x \geq 0 \wedge y \geq 0 \wedge x - y \geq 0 \wedge x^2 - a < 0\big).$$

**Results:**

- Within 20 ms a weakly quantifier-free equivalent containing 26 atomic formulas
- Setting $a = 10$ and automatically simplifying yields within 2980 ms the result $z > 8$, i.e., the minimum for $x + y$ is 4.
- If we plug in $a = 10$ before the elimination, then we directly obtain $z > 3$ in only 780 ms.

# Software Security—Data and Control Flow

**Example code**

```
if (a < b) then
    if (a+b mod 2 = 0) then
        n := (a+b)/2
    else
        n := (a+b+1)/2
    fi
    A[n*n] := get_sensitive_data(x)
   send_sensitive_data(trusted_receiver,A[n*n])
fi
y := A[abs(b-a)]
```

**Security risk:** There exist choices for `a` and `b` such that `y` is assigned the value of `A[n*n]`.

# Software Security—Data and Control Flow

**Example code**

```
if (a < b) then
   if (a+b mod 2 = 0) then
      n := (a+b)/2
   else
      n := (a+b+1)/2
   fi
   A[n*n] := get_sensitive_data(x)
   send_sensitive_data(trusted_receiver,A[n*n])
fi
y := A[abs(b-a)]
```

**Security risk:** There exist choices for `a` and `b` such that `y` is assigned the value of `A[n*n]`.

# Software Security—Data and Control Flow

$$\exists n\big((a < b \wedge a + b \equiv_2 0 \wedge 2n = a + b \wedge ((a < b \wedge b - a = n^2) \vee (a \geq b \wedge a - b = n^2))) \vee (a < b \wedge a + b \not\equiv_2 0 \wedge 2n = a + b + 1 \wedge ((a < b \wedge b - a = n^2) \vee (a \geq b \wedge a - b = n^2)))\big).$$

Our implementation computes in less than 10 ms the following weakly quantifier-free description:

$$\bigsqcup_{k\colon |k| \leq (a-b)^2+2} (a-b < 0 \wedge a-b+k^2 = 0 \wedge a+b \not\equiv_2 0 \wedge a+b-2k+1 = 0) \vee$$

$$\bigsqcup_{k\colon |k| \leq (a-b)^2+2} (a-b < 0 \wedge a-b+k^2 = 0 \wedge a+b \equiv_2 0 \wedge a+b-2k = 0).$$

# Software Security—Data and Control Flow

$$\exists n\big((a < b \wedge a + b \equiv_2 0 \wedge 2n = a + b \wedge \\ ((a < b \wedge b - a = n^2) \vee (a \geq b \wedge a - b = n^2))) \vee \\ (a < b \wedge a + b \not\equiv_2 0 \wedge 2n = a + b + 1 \wedge \\ ((a < b \wedge b - a = n^2) \vee (a \geq b \wedge a - b = n^2)))\big).$$

Our implementation computes in less than 10 ms the following weakly quantifier-free description:

$$\bigsqcup_{k\colon |k| \leq (a-b)^2+2} (a-b < 0 \wedge a-b+k^2 = 0 \wedge a+b \not\equiv_2 0 \wedge a+b-2k+1 = 0) \vee$$

$$\bigsqcup_{k\colon |k| \leq (a-b)^2+2} (a-b < 0 \wedge a-b+k^2 = 0 \wedge a+b \equiv_2 0 \wedge a+b-2k = 0).$$

# Conclusions

- Weak quantifier elimination procedure for the univariately nonlinear formulas
- Price to pay: Bounded quantifiers
- Expansion into regular first-order formulas for fixed choices of parameters
- Decision procedure even for the regular first-order framework
- Efficient publicly available implementation within the computer logic system REDLOG, which is part of REDUCE
- Demonstration of applicability of our new method and its implementation by means of various application examples

# Conclusions

- Weak quantifier elimination procedure for the univariately nonlinear formulas
- Price to pay: Bounded quantifiers
- Expansion into regular first-order formulas for fixed choices of parameters
- Decision procedure even for the regular first-order framework
- Efficient publicly available implementation within the computer logic system REDLOG, which is part of REDUCE
- Demonstration of applicability of our new method and its implementation by means of various application examples

# Conclusions

- Weak quantifier elimination procedure for the univariately nonlinear formulas
- Price to pay: Bounded quantifiers
- Expansion into regular first-order formulas for fixed choices of parameters
- Decision procedure even for the regular first-order framework
- Efficient publicly available implementation within the computer logic system REDLOG, which is part of REDUCE
- Demonstration of applicability of our new method and its implementation by means of various application examples

# Conclusions

- Weak quantifier elimination procedure for the univariately nonlinear formulas
- Price to pay: Bounded quantifiers
- Expansion into regular first-order formulas for fixed choices of parameters
- Decision procedure even for the regular first-order framework
- Efficient publicly available implementation within the computer logic system REDLOG, which is part of REDUCE
- Demonstration of applicability of our new method and its implementation by means of various application examples

# Conclusions

- Weak quantifier elimination procedure for the univariately nonlinear formulas
- Price to pay: Bounded quantifiers
- Expansion into regular first-order formulas for fixed choices of parameters
- Decision procedure even for the regular first-order framework
- Efficient publicly available implementation within the computer logic system REDLOG, which is part of REDUCE
- Demonstration of applicability of our new method and its implementation by means of various application examples

# Conclusions

- Weak quantifier elimination procedure for the univariately nonlinear formulas
- Price to pay: Bounded quantifiers
- Expansion into regular first-order formulas for fixed choices of parameters
- Decision procedure even for the regular first-order framework
- Efficient publicly available implementation within the computer logic system REDLOG, which is part of REDUCE
- Demonstration of applicability of our new method and its implementation by means of various application examples