# REDLOG as a Tool in Symbolic Algebra and Trustworthy Computing

Thomas Sturm, Universität Passau, Germany

SRATC ’08, ECNU, Shanghai, China, April 5, 2008

# Overview

- Real Quantifier Elimination and Variants
- Other Domains
- Online Resources
- Integer Quantifier Elimination
- Work in Progress and Visions for the Future
- Summary

# REDLOG
## Joint Project with Andreas Dolzmann

- 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. 3.060805)
- currently 30–40 kloc

# Two Real Examples

## Quantifier Elimination

Fix **syntax**: a set of function symbols and relation symbols.

Fix **semantics**: a domain and an interpretation for these symbols.

Given a first-order formula $\varphi$ find quantifier-free $\varphi'$ such that $\varphi' \longleftrightarrow \varphi$.

## Easy Example (syntax $(0, 1, +, -, \cdot, =, \leq, \neq, <)$ / semantics $\mathbb{R}$)

$$\varphi \equiv \exists x(ax^2 + bx + c = 0)$$

$$\updownarrow$$

$$\varphi' \equiv (a \neq 0 \land b^2 - 4ac \geq 0) \lor (a = 0 \land b \neq 0) \lor (a = 0 \land b = 0 \land c = 0)$$

## Example by Hoon Hong (syntax / semantics as above)

$$\varphi \equiv \forall x \exists y(x^2 + xy + b > 0 \land x + ay^2 + b \leq 0)$$

$$\updownarrow$$

$$\varphi' \equiv a < 0 \land b > 0$$

# Real Elimination Methods in REDLOG

## Partial CAD (Collins 1973, Collins and Hong 1991)

- Doubly exponential in the number of all variables
- Generally applicable
- Reasonably simple results

## Virtual Substitution (Weispfenning 1988)

- Doubly exponential in the number of quantifier changes
- Restricted to formulas with low-degree polynomials
- Produces a large number of atomic formulas

## Hermitian Quantifier Elimination (Weispfenning 1993)

- Not elementary recursive
- Aims at formulas with many equations
- Produces huge polynomials with huge coefficients

# Automatic Combination of Methods

## Currently Fallback Quantifier Elimination

- Virtual Substitution as long as possible
- Then partial CAD

## Hong's Example

$$\varphi \equiv \forall x \exists y (x^2 + xy + b > 0 \wedge x + ay^2 + b \leq 0) \rightsquigarrow \varphi' \equiv a < 0 \wedge b > 0$$

- First eliminate $\exists y$ by virtual substitution, then eliminate $\forall x$ by partial CAD.

  Takes 0.7 seconds altogether.
- Virtual substitution for $\forall x$ fails (degree 4).
- Partial CAD for the entire problem takes 86 seconds.

  Factor > 100.

Long-term goal: Meta Quantifier Elimination.

# Virtual Substitution

- Given $\exists x \varphi$, where $\varphi \equiv ax + b = 0$.
- For fixed $a$, $b$ every such $\varphi$ describes a finite union of intervals.
- Collect all endpoints of intervals **guarded** by conditions for their existence:

$$E = \left\{ \left( a \neq 0, -b/a \right) \right\}.$$

- Add to the **elimination set** one point with "true" as its guard:

$$E = \left\{ \left( a \neq 0, -b/a \right), (\text{true}, 0) \right\}.$$

- Use modified substitution for the pseudo-terms:

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

- The formal result:

$$\left( a \neq 0 \wedge \left( a \cdot \frac{-b}{a} + b \right) \cdot a = 0 \cdot a \right) \vee \left( \text{true} \wedge a \cdot 0 + b = 0 \right).$$

- Simplify the result: $\varphi' \equiv a \neq 0 \vee b = 0$.

# Extended Quantifier Elimination

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

to the **extended quantifier elimination scheme**

$$\exists x\varphi \rightsquigarrow \begin{bmatrix} \vdots & \vdots \\ \gamma \wedge \varphi[x /\!/ t] & \{x = t\} \\ \vdots & \vdots \end{bmatrix}.$$

## Semantics

Fix all parameters.
If some left hand side condition holds, then $\exists x\varphi$ holds
and the corresponding right hand side term is **one** sample solution.

## In Our Example

$$\begin{bmatrix} a \neq 0 & \{x = -\frac{b}{a}\} \\ b = 0 & \{x = 0\} \end{bmatrix}.$$

# Successful Real Applications of REDLOG

- parametric and nonlinear optimization
- transportation problems
- circuit analysis, -design, -diagnosis
- generalized scheduling problems
- real implicitation
- automated theorem proving
- computational geometry
- solid modeling
- robot motion planning
- algebraic biology
- factorization of LPDOs
- automatic loop parallelization (Lengauer)
- bifurcation analysis (El Kahoui, Weber)
- theoretical mechanics (Ioakimidis)
- stability of differential equations (Hong, Liska, Steinberg)
- hybrid control theory (Yovine, Anai/FUJITSU)
- atmosphere chemistry (Lustfeld)
- hydraulic network diagnosis (ROSE)
- runtime properties of programs (Anderson et al.)
- reasoning in complex theories (Sofronie-Stokkermans)

# Many More Domains and Applications

## Reals (JSC 97, JAR 98, AAECC 99, CASC 00, ISSAC 97/00/03/04, ...)

- discussed before

## Complex

- language of rings only

## Differential (CASC 2004)

- language of rings with unary differential operator
- computation in differentially closed field (A. Robinson, Blum)

## Padics (JSC 00, ISSAC 99, CASC 01)

- linear formulas over $p$-adic fields for $p$ prime
- optionally uniform in $p$
- used e.g. for solving parametric systems of congruences over the integers

# Yet More Domains and Applications

## Terms (CASC 2002)

- Malcev-type term algebras (with functions instead of relations)

## Queues (C. Straßer at RWCA 2006)

- two-sided queues over the other domains (2-sorted)
- Implemented at present for queues of reals

## Boolean (CASC 2003, C. Zengler 2008)

- generalization of SAT-checking
- quantified propositional calculus (parametric QSAT-checking)

## First-Order Theorem Proving (S. Käser 2007)

- Generalized Gröbner bases approach by Kapur and Narendran.

## Integers (AAECC 2007, CASC 2007)

- Some details soon . . .

# Online Resources: The REDLOG Website

- Regular REDLOG updates for download.
- Documentation as both HTML and for download.
- References (generated from the REMIS database)
  - REDLOG system papers
  - REDLOG applications
  - REDLOG 3rd-party applications
  - Theoretical foundations.
- REMIS = REDLOG Example Management and Information System

**Where?**

**www.redlog.eu**

# Recall Real QE by Virtual Substitution

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

## Example

- Consider $\mathbb{R}$, arithmetic, ordering:

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

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

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

- For linear formulas one can always find elimination sets [Weispfenning 1988].
- This can be extended to higher degrees to some extent [Weispfenning 1997].

# The Same Problem Over the Integers

## Example

- Consider $\mathbb{Z}$, arithmetic, ordering, **congruences**:

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

- One possible QE result:

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

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

- Systematic use of formal $\bigvee$-notation decreases complexity by one exponential step [Weispfenning 1990].
- QE can be interpreted within the virtual substitution framework: $E = \{ (b + k \equiv_3 0, (b + k)/3) \mid |k| \leq 3 \}$ [Lasaruk 2005, Lasaruk + S. 2007].

# Presburger Arithmetic

Presburger Arithmetic is the **additive** theory of $\mathbb{Z}$ with ordering and congruences:

## Mojzesz Presburger

*Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt*

Dissertation, Warsaw 1929

- $3x$ is possibly short for $x + x + x$.
- Our example $\exists x(3x - b = 0)$ is a Presburger formula.
- In contrast, $\exists x(ax - b = 0)$ is **not** a Presburger formula.

# Introducing Parameters Into Presburger Arithmetic

- Again $\mathbb{Z}$, arithmetic, ordering, congruences.
- Now make essential use of multiplication:

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

- Copy the elimination approach from before:

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

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

**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.

# Bounded Quantifiers and Weak QE

Formally extend logic by new quantifiers with the following semantics:

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

We say **bounded quantifier** if the **range** $\beta$ is finite for all choices of parameters.

## This solves our previous problem

$$\bigsqcup_{k:\,|k|<|a|} \left( a \neq 0 \land b + k \equiv_3 0 \land (ax - y = 0) \left[ \frac{b+k}{3} /\!/ x \right] \right)$$ is OK in extended logic.

- If $\beta$ contains only $k$, then $\bigsqcup_{k:\,\beta} \varphi \longleftrightarrow \bigvee_{i \in \{z \in \mathbb{Z} \mid \beta(z)\}} \varphi[i/k]$.

**Weak quantifier elimination:** Results may contain bounded quantifiers.

## Major Result (Lasaruk + S., AAECC 2007)

Linear formulas (with arbitrary polynomial coefficients) admit weak QE.

# Application to Information Flow Control

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

## Question

Can the sensitive information `A[n]` possibly become **nonlocal** via assignment to `y`?

# Our Contribution to the Solution

Path condition automatically generated by software engineering tools:

$$\begin{aligned}\exists a \exists b \exists n \big( & (a < b \land a + b \equiv_2 0 \land 2n = a + b \land \\ & \quad ((a < b \land b - a = n) \lor (a \geq b \land a - b = n))) \lor \\ & (a < b \land a + b \not\equiv_2 0 \land 2n = a + b + 1 \land \\ & \quad ((a < b \land b - a = n) \lor (a \geq b \land a - b = n)))\big).\end{aligned}$$

Extended quantifier elimination for attackers:

$$\left[\begin{array}{c|c} \text{true} & \{n = 1, b = 1, a = 0\} \\ \text{true} & \{n = 2, b = 3, a = 1\} \end{array}\right].$$



Regular quantifier elimination for defenders:

$$(3a - b + 1 = 0 \land a + b \equiv_2 0 \land a < b) \lor (3a - b = 0 \land a + b \not\equiv_2 0 \land a < b).$$

# Towards Higher Degrees

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

**Example**

Weakly eliminate $\exists x$ from $\varphi \equiv \exists x(ax - y < 0 \wedge x^2 + x + a > 0)$.

Our result:

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

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

- For $a = 10$ this can be turned into a regular first-order formula:

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

# Which Formulas Can We Handle So Far?

The set of **univariately nonlinear formulas** is defined as follows:

1. No quantified variables within moduli of (in)congruences.
2. (In)congruences are linear in the quantified variables.
3. Equations and inequalities are either linear in the quantified variables or **superlinear univariate** in one of the quantified variables: i.e., they contain exactly one quantified variable, but with arbitrary degrees.

## Examples

- linear: $\forall a \forall b (a < b \longrightarrow \exists z (a < z \wedge z < b) \vee ax - y \equiv_{m+7} 0)$.
- univariately nonlinear: $\forall y \exists x (ax - y < 0 \wedge 5a^7 x^2 + 3x + a + b > 0)$.
- **not** univariately nonlinear: $\forall y \exists x (ax - y < 0 \wedge 5a^7 x^2 + 3x + a + y > 0)$.
- **not** univariately nonlinear: $\exists x \exists y \exists z (x^5 + y^5 = z^5)$.

- Linear formulas are special cases of univariately nonlinear formulas.

# Recent Major Result

**Theorem (Lasaruk + S. CASC 2007)**

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

- We can positively decide in advance, whether or not all quantifiers can be eliminated by our method.

**Fact**

Let $L$ be a language, and let $A$ be an $L$-Structure.

If $A$ admits QE and variable-free atomic formulas are decidable in $A$, then $A|_{L'}$ is decidable for all $L' \subseteq L$.

- The argument remains correct even for weak QE!

**Corollary (Decidability of Sentences)**

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

# Basic Technical Ideas

- Known test points for the linear case [Lasaruk + S. AAECC 2007]
- On the one hand, proceed on the assumption that everything is happening outside the **Cauchy bounds** for superlinear univariate atomic formulas.
- On the other hand, introduce further bounded quantifiers completely covering the (parametric) range within the Cauchy bounds.
- Need generalized concept of **constrained virtual substitution**.
- Technically, elements of parametric elimination sets contain in addition
  - bounded quantifiers to be introduced
  - substitution to be used.

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

- Elimination scheme:

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

# Information Flow Control Revisited

## Example code

```
if (a < b) then
   // BEGIN SECURE CODE
   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])
   // END SECURE CODE
fi
y := A[abs(b-a)]
```

- Are there choices for `a` and `b` such that `y` is assigned the value of `A[n*n]`?
- That would be a security risk.

# Our solution with REDLOG

## First-order formulation of both code and question

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

- This is univariately nonlinear.

## Applying weak QE with REDLOG

Weakly quantifier-free description in less than 10 ms:

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

# Serious Applications of Integer QE?

Everything discussed so far is implemented and publicly available in REDLOG.

Possible application domains include the following:

- nonlinear discrete optimization problems
- integer linear optimization with superlinear univariate constraints
- **software security**
- automatic loop parallelization
- scheduling problems

**Unfortunately**

No convincing killer application so far.

**BUT**

Excellent basis for mixed real-integer QE.

# Towards Mixed Real-integer QE

**Mixed Real-Integer Quantifier Elimination (Weispfenning, 1999)**

- Presburger Arithmetic + Real QE for Presburger-like atomic formulas.

- Prototype implementation in REDLOG exists.
- Ongoing research on possible generalizations.
- In particular in view of our generalized integer quantifier elimination.

# Probabilistic Quantifier Elimination

- Real quantifier elimination is doubly exponential. Pretty bad!
- Presburger integer quantifier elimination is triply exponential. Even worse!
- Term algebras QE in the 3rd class of the Grzegorczyk hierarchy. Hmm ...

## Idea

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

- Substitute only a subset of terms.
- If we obtain (fixing parameters) "true" anyway, then this is certainly correct.
- Can we substitute sufficiently many terms to have positively bounded correctness probability in case of "false"? ($\rightsquigarrow$ RP-like class, Monte-Carlo)

## First steps: Work in progress with Aless Lasaruk

- Focuses on bounded quantifiers introduced with integer QE.
- Implementation `pqe(phi,p)` exists.
- Theoretic concepts are mostly worked out for this special case.

# Programmatic Quantifier Elimination

- Why do we want quantifier-free equivalents at all?
  1. Understanding the quantifier-free formulas gives insights.
  2. We want to plug in values for parameters, and evaluate to true/false.
- Let us focus on point 2.

## Facts

- QE complexity is driven by the size of the results.
- If we leave the framework of logic,
  then the known lower bounds are not necessarily valid anymore.

## My Vision ...

- Consider instead of quantifier-free formulas, primitive recursive programs.
- Accept such programs also as input.
- Reasonable first step: straight-line programs (suggested by Joos Heintz).

# From an Interactive Tool to a Library

- There are many successful applications of REDLOG in the literature
  (a) Interactive applications
  (b) Applications, where REDLOG performs as pre/postprocessor

## Goal

- Use REDLOG as a library from other applications
- Frequently asked in particular by users from computer science

## But

- Linking C-like code (or Java) with Lisp is a well-known problem.

## Recent Project

- `libreduce.a`, which can be linked to C.
- Prototype exists (works with SPASS prover of MPI Saarbrücken).
- Extension to C++ etc. is straightforward.
- Extension to Java is possible.

# Summary

- Real quantifier elimination and variants are well-established.
- Numerous other less established but interesting domains.
- The REDLOG website and REMIS.
- Major progresses in integer quantifier elimination.
- Possible application in information flow control (software security)
- Work in progress (mixed real-integer QE).
- REDLOG can be used as a library from C (and the like).

**Have a look at REDLOG anytime**

**www.redlog.eu**