Using Real Quantifier Elimination for Synthesizing Optimal Numerical Algorithms

Mădălina Erașcu

West University of Timișoara and Institute e-Austria Timișoara, Romania

merascu@info.uvt.ro

Joint work with Hoon Hong, NCSU, Raleigh, USA

April 27, 2016

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \ \wedge \ f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \ \wedge \ f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \wedge f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \ \wedge \ f(y) = x$.

Numerical algorithm

Initialize I

while width$(I) > \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$
in: x - real number
ε - error bound
out: an interval $\boldsymbol{I}$ with width less than equal ε such that $y \in \boldsymbol{I} \wedge f(y) = x$.

Numerical algorithm
Initialize $\boldsymbol{I}$
while width($\boldsymbol{I}$) $> \varepsilon$
$\boldsymbol{I} \leftarrow R(\boldsymbol{I}, x)$
return $\boldsymbol{I}$

Optimal
R such that the interval $\boldsymbol{I}$ shrinks fastest.

Synthesize
Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \wedge f(y) = x$.

Numerical algorithm

Initialize I

while width$(I) > \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \wedge f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \wedge f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \ \wedge \ f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \wedge f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $f(y) = x$

Solving $f(y) = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \wedge f(y) = x$.

Numerical algorithm

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Problem

Synthesize an Optimal Numerical algorithm for Solving $y^2 = x$

Solving $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than equal ε such that $y \in I \land y^2 = x$.

Numerical algorithm

Initialize I

while width$(I) > \varepsilon$

$I \leftarrow R(I, x)$

return I

Optimal

R such that the interval I shrinks fastest.

Synthesize

Find such R.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. **Numerical analysis**
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. **Symbolic Computation**
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. **Theorem proving/algorithm synthesis**
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing **18** (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant–Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant–Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant–Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing 18 (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing **18** (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Motivation

Computing the square root of a given real number – fundamental operation.

Importance:

1. Numerical analysis
   Improve the convergence of an existing algorithm for computing the square root of a real number
2. Symbolic Computation
   - Demonstrate the power of symbolic methods over numeric methods
   - Advancing the state-of-the-art of quantifier elimination methods over reals
3. Theorem proving/algorithm synthesis
   (Semi-) automatic synthesis of optimal algorithms by combining in a novel way algebraic and logical methods

Publications:

M. Erascu and H. Hong, *The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation*, Journal of Reliable Computing **18** (2013), 73–81.

M. Erascu and H. Hoon, *Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination (Case Study: Square Root Computation)*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC) (K. Nabeshima, K. Nagasaka, F. Winkler, and A. Szanto, eds.), ACM, 2014, pp. 162–169.

M. Erascu and H. Hong, *Real Quantifier Elimination for the Synthesis of Optimal Numerical Algorithms (Case Study: Square Root Computation)*, J. Symb. Comput. **75** (2016), 110–126.

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

Examples:

- $\underset{x}{\exists}\ ax^2 + bx + c = 0 \wedge a \neq 0$
- $$-1 < p_2 \leq 0 \ \wedge\ p_1 \leq -2p_2 \ \wedge\ 1 + p2 \leq p4 \leq 2 \ \wedge$$
  $$\underset{\substack{L,U,y \\ 0<L\leq y\leq U}}{\forall} \quad (1 - p_1 - p_2 - p_4)L^2 + (p_1 + p_4)LU + p_2U^2 + y^2 > 0$$

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

Examples:

- $\underset{x}{\exists}\ ax^2 + bx + c = 0 \wedge a \neq 0$
- $$-1 < p_2 \leq 0 \ \wedge \ p_1 \leq -2p_2 \ \wedge \ 1 + p2 \leq p4 \leq 2 \ \wedge$$
  $$\underset{\substack{L,U,y \\ 0<L\leq y\leq U}}{\forall} (1 - p_1 - p_2 - p_4)L^2 + (p_1 + p_4)LU + p_2U^2 + y^2 > 0$$

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

Examples:

- $\underset{x}{\exists}\ ax^2 + bx + c = 0 \wedge a \neq 0$
- $$-1 < p_2 \leq 0 \ \wedge\ p_1 \leq -2p_2 \ \wedge\ 1 + p2 \leq p4 \leq 2 \ \wedge$$
  $$\underset{\substack{L,U,y \\ 0<L\leq y\leq U}}{\forall} (1 - p_1 - p_2 - p_4)L^2 + (p_1 + p_4)LU + p_2U^2 + y^2 > 0$$

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

Examples:

- $\underset{x}{\exists}\ ax^2 + bx + c = 0 \wedge a \neq 0$
- $$-1 < p_2 \leq 0 \ \wedge\ p_1 \leq -2p_2 \ \wedge\ 1 + p2 \leq p4 \leq 2\ \wedge$$
  $$\underset{\substack{L,U,y \\ 0<L\leq y\leq U}}{\forall} (1 - p_1 - p_2 - p_4)L^2 + (p_1 + p_4)LU + p_2U^2 + y^2 > 0$$

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski	First algorithm
	Based on Sylvester-Sturm Theorem
	$2^{2^{\cdot^{\cdot^{n}}}}$
1975 Collins	First algorithm with elementary complexity
	Based on Cylindrical Algebraic Decomposition
	$2^{2^{n}}$
1975 —	Doubly exponential in the number of quantifier blocks
	Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^n}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$2^{2^{\cdot^{\cdot^{n}}}}$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$2^{2^{n}}$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^n}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski — First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins — First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^n}$$

1975 — — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{\cdot^{n}}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski — First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins — First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski	First algorithm
	Based on Sylvester-Sturm Theorem
	$2^{2^{\cdot^{\cdot^{n}}}}$
1975 Collins	First algorithm with elementary complexity
	Based on Cylindrical Algebraic Decomposition
	$2^{2^{n}}$
1975 —	Doubly exponential in the number of quantifier blocks
	Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski	First algorithm Based on Sylvester-Sturm Theorem	$2^{2^{\cdot^{\cdot^{n}}}}$
1975 Collins	First algorithm with elementary complexity Based on Cylindrical Algebraic Decomposition	2^{2^n}
1975 —	Doubly exponential in the number of quantifier blocks Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski	First algorithm
	Based on Sylvester-Sturm Theorem
	$2^{2^{\cdot^{\cdot^{\cdot^{n}}}}}$
1975 Collins	First algorithm with elementary complexity
	Based on Cylindrical Algebraic Decomposition
	2^{2^n}
1975 —	Doubly exponential in the number of quantifier blocks
	Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski	First algorithm
	Based on Sylvester-Sturm Theorem
	$2^{2^{\cdot^{\cdot^{n}}}}$
1975 Collins	First algorithm with elementary complexity
	Based on Cylindrical Algebraic Decomposition
	2^{2^n}
1975 —	Doubly exponential in the number of quantifier blocks
	Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Quantifier Elimination (QE) over Real-Closed Fields (RCF)

Problem QE over RCF

Input : ϕ - a formulas in the first-order theory of RCF

Output : ψ - a quantifier-free-formula equivalent to ϕ.

History

1950 Tarski First algorithm

Based on Sylvester-Sturm Theorem

$$2^{2^{\cdot^{\cdot^{n}}}}$$

1975 Collins First algorithm with elementary complexity

Based on Cylindrical Algebraic Decomposition

$$2^{2^{n}}$$

1975 — Doubly exponential in the number of quantifier blocks

Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)
- SMT (Satisfiability Modulo Theory) solvers

Numerical Algorithms (Square Root)

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval $\boldsymbol{I}$ with width less than ε such that $y \in \boldsymbol{I} \ \wedge \ y^2 = x$.

Algorithm schema: Interval refining

Initialize $\boldsymbol{I}$

while width($\boldsymbol{I}$) $> \varepsilon$

$\boldsymbol{I} \leftarrow R(\boldsymbol{I}, x)$

return $\boldsymbol{I}$

Numerical Algorithms (Square Root)

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \ \wedge \ y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Numerical Algorithms (Square Root)

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \;\wedge\; y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while width(I) $> \varepsilon$

$I \leftarrow \left[L + \frac{x - L^2}{L + U}, U + \frac{x - U^2}{2U}\right]$ Secant-Newton Refining Map

return I

Numerical Algorithms (Square Root)

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \ \wedge \ y^2 = x$.

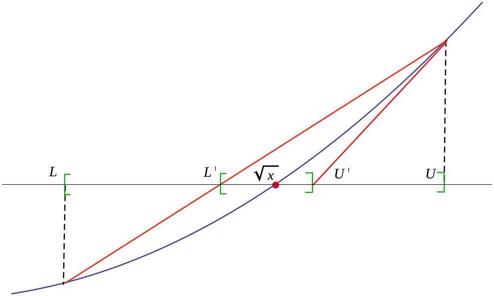
Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while width(I) $> \varepsilon$

$I \leftarrow \left[L + \frac{x-L^2}{L+U}, U + \frac{x-U^2}{2U}\right]$ Secant-Newton Refining Map

return I



Numerical Algorithms (Square Root)

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \ \wedge \ y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while width(I) $> \varepsilon$

$I \leftarrow \left[L + \frac{x - L^2}{L+U}, U + \frac{x - U^2}{2U}\right]$ Secant-Newton Refining Map

return I

Is there any refinement map which is better than Secant-Newton?

Numerical Algorithms (Square Root) Synthesis

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \ \wedge \ y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while width(I) $> \varepsilon$

$I \leftarrow \left[L + \frac{x - L^2}{L + U}, U + \frac{x - U^2}{2U}\right]$

return I

Numerical Algorithms (Square Root) Synthesis

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \ \wedge \ y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while width$(I) > \varepsilon$

$$I \leftarrow \left[L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}, U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right]$$

Quadratic Refining Map

return I

Numerical Algorithms (Square Root) Synthesis

Problem: solve $y^2 = x$

in: x - real number

ε - error bound

out: an interval I with width less than ε such that $y \in I \;\wedge\; y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow [\min(1, x), \max(1, x)]$

while $\mathrm{width}(I) > \varepsilon$

$$I \leftarrow \left[L + \frac{x+(-1)L^2+0LU+0U^2}{1L+1U}, U + \frac{x+(-1)U^2+0UL+0L^2}{2U+0L}\right]$$ Secant-Newton

return I

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\iff \forall_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}} \; 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ 1>c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U-L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U'-L'}{U-L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\Longleftrightarrow \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \underset{\substack{c \\ 1>c>0}}{\exists} \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} U' - L' \leq c\,(U-L)$$

$$QuadraticConv(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \underset{\substack{c \\ c>0}}{\exists} \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} U' - L' \leq c\,(U-L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U'-L'}{U-L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\Longleftrightarrow \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} U' - L' \leq c\,(U-L)$$

$$QuadraticConv(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} U' - L' \leq c\,(U-L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U'-L'}{U-L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p, q) :\Longleftrightarrow \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p, q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \underset{\substack{c \\ 1>c>0}}{\exists} \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} U' - L' \leq c\,(U - L)$$

$$QuadraticConv(p, q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \underset{\substack{c \\ c>0}}{\exists} \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} U' - L' \leq c\,(U - L)^2$$

Minimize

$$E(p, q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U' - L'}{U - L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\iff \forall_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}} \; 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ 1>c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U - L)$$

$$QuadraticConv(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U - L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U' - L'}{U - L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\Longleftrightarrow \underset{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U}}{\forall} \; 0 < L' \le \sqrt{x} \le U'$$

$$Termination(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)$$

$$QuadraticConv(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U \\ L\neq U}} \frac{U'-L'}{U-L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\Longleftrightarrow \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)$$

$$QuadraticConv(p,q) :\Longleftrightarrow \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U' - L'}{U - L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$\mathit{Correctness}(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \; 0 < L' \leq \sqrt{x} \leq U'$$

$$\mathit{Termination}(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \; U' - L' \leq c\,(U - L)$$

$$\mathit{QuadraticConv}(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \; U' - L' \leq c\,(U - L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U' - L'}{U - L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

Subject to

$$Correctness(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)$$

$$QuadraticConv(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)^2$$

Minimize

$$E(p,q) = \sup_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U' - L'}{U - L}$$

Standard numerical optimization methods cannot be applied because:

1. The objective function is itself the result of parametric optimization (sup).
2. The constraints are quantified formulas.
3. It turns out that there are infinitely many values of p and q with the same minimum.

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U}}{\forall} \; 0 < L' \le \sqrt{x} \le U'$$

$$Termination(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)^2$$

$$Optimality(p,q): \iff \quad \ldots$$

Trouble: state-of-the-art QE software take very long time ($\gg$ several days)

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U}}{\forall} \; 0 < L' \le \sqrt{x} \le U'$$

$$Termination(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)^2$$

$$Optimality(p,q) :\iff \ldots$$

Trouble: state-of-the-art QE software take very long time ($\gg$ several days)

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p, q) :\iff \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p, q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)$$

$$QuadraticConv(p, q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U - L)^2$$

$$Optimality(p, q) :\iff \ldots$$

Strategies:

1. divide the QE problems into several simpler ones
2. apply state-of-the-art QE software on some of them
3. reducing the number of quantified variables by judicially instantiating some of them
4. manual simplification on the remaining ones
5. reduce the search space p, q by imposing the condition

$$K(p, q) \;:\iff\; 0 < p_4 \leq 2 \;\wedge\; p_3 + p_4 = 2 \;\wedge\; 0 < q_3 \leq 2 \;\wedge\; q_3 + q_4 = 2$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \; 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \; U' - L' \leq c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \; U' - L' \leq c\,(U-L)^2$$

$$Optimality(p,q) :\iff \ldots$$

Strategies:

1. divide the QE problems into several simpler ones
2. apply state-of-the-art QE software on some of them
3. reducing the number of quantified variables by judicially instantiating some of them
4. manual simplification on the remaining ones
5. reduce the search space p, q by imposing the condition

$$K(p,q) \;:\iff\; 0 < p_4 \leq 2 \;\wedge\; p_3 + p_4 = 2 \;\wedge\; 0 < q_3 \leq 2 \;\wedge\; q_3 + q_4 = 2$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\le\sqrt{x}\le U}}{\forall} \; 0 < L' \le \sqrt{x} \le U'$$

$$Termination(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ 1>c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \; \underset{\substack{c \\ c>0}}{\exists} \; \underset{\substack{L,U \\ 0<L\le\sqrt{x}\le U}}{\forall} \; U' - L' \le c\,(U-L)^2$$

$$Optimality(p,q) :\iff \ldots$$

Strategies:

1. divide the QE problems into several simpler ones
2. apply state-of-the-art QE software on some of them
3. reducing the number of quantified variables by judicially instantiating some of them
4. manual simplification on the remaining ones
5. reduce the search space p, q by imposing the condition

$$K(p,q) \;:\iff\; 0 < p_4 \le 2 \;\wedge\; p_3 + p_4 = 2 \;\wedge\; 0 < q_3 \le 2 \;\wedge\; q_3 + q_4 = 2$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \underset{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \quad \underset{\substack{c \\ 1>c>0}}{\exists} \quad \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \underset{\substack{x \\ x>0}}{\forall} \quad \underset{\substack{c \\ c>0}}{\exists} \quad \underset{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}}{\forall} \quad U' - L' \leq c\,(U-L)^2$$

$$Optimality(p,q) :\iff \ldots$$

Strategies:

1. divide the QE problems into several simpler ones
2. apply state-of-the-art QE software on some of them
3. reducing the number of quantified variables by judicially instantiating some of them
4. manual simplification on the remaining ones
5. reduce the search space p, q by imposing the condition

$$K(p,q) \quad :\iff \quad 0 < p_4 \leq 2 \;\wedge\; p_3 + p_4 = 2 \;\wedge\; 0 < q_3 \leq 2 \;\wedge\; q_3 + q_4 = 2$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \forall_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}} \; 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ 1>c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U-L)$$

$$QuadraticConv(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U-L)^2$$

$$Optimality(p,q) :\iff \ldots$$

Strategies:

1. divide the QE problems into several simpler ones
2. apply state-of-the-art QE software on some of them
3. reducing the number of quantified variables by judicially instantiating some of them
4. manual simplification on the remaining ones
5. reduce the search space p, q by imposing the condition

$$K(p,q) \quad :\iff \quad 0 < p_4 \leq 2 \;\wedge\; p_3 + p_4 = 2 \;\wedge\; 0 < q_3 \leq 2 \;\wedge\; q_3 + q_4 = 2$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE

$$L' = L + \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U} \qquad U' = U + \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}$$

$$Correctness(p,q) :\iff \forall_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U}} \; 0 < L' \leq \sqrt{x} \leq U'$$

$$Termination(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ 1>c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U - L)$$

$$QuadraticConv(p,q) :\iff \forall_{\substack{x \\ x>0}} \; \exists_{\substack{c \\ c>0}} \; \forall_{\substack{L,U \\ 0<L\leq\sqrt{x}\leq U}} \; U' - L' \leq c\,(U - L)^2$$

$$Optimality(p,q) :\iff \; \ldots$$

Strategies:

1. divide the QE problems into several simpler ones
2. apply state-of-the-art QE software on some of them
3. reducing the number of quantified variables by judicially instantiating some of them
4. manual simplification on the remaining ones
5. reduce the search space p, q by imposing the condition

$$K\,(p,q) \;:\iff\; 0 < p_4 \leq 2 \;\wedge\; p_3 + p_4 = 2 \;\wedge\; 0 < q_3 \leq 2 \;\wedge\; q_3 + q_4 = 2$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE

Main Result:

(a) $E(p,q) \geq \frac{1}{4}$ $\quad\quad \left(E(p^*,q^*) = \frac{1}{2},\ \text{where } p^*, q^* \text{ are for Secant-Newton}\right)$

(b) $E(p,q) = \frac{1}{4}$ iff

$$p_3 = q_3 = p_4 = q_4 = 1 \ \wedge\ p_2 = 0 \ \wedge\ q_0 = -\frac{3}{4} \ \wedge$$
$$p_0 + p_1 = -1 \ \wedge\ q_1 + q_2 = -\frac{1}{4} \ \wedge$$
$$-\frac{1}{2} \leq q_1 \leq p_1 \leq 0$$

In other words

$$L' = L + \frac{x - (1 + p_1)\, L^2 + p_1 LU}{L + U}$$
$$U' = U + \frac{x - \frac{3}{4} U^2 + q_1 UL - \left(\frac{1}{4} + q_1\right) L^2}{U + L}$$

where $-\frac{1}{2} \leq q_1 \leq p_1 \leq 0$

How much improvement?

	Secant-Newton Map $R^*(I,x)$	Synthesized Map $\tilde{R}(I,x)$	
Original	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-U^2}{2U}\right]$	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-\frac{3}{4}U^2-\frac{1}{2}LU+\frac{1}{4}L^2}{U+L}\right]$	
Rewritten	$\left[\frac{x+LU}{L+U}, \frac{x}{U+U}+\frac{1}{4}(U+U)\right]$	$\left[\frac{x+LU}{L+U}, \frac{x}{U+L}+\frac{1}{4}(U+L)\right]$	
# of ops.	9	9	The same
Convergence	Quadratic	Quadratic	The same
Lipschitz	$\frac{1}{2}$	$\frac{1}{4}$	Better
# of loop iters.	$\log_2 \frac{I_0}{\varepsilon}$	$\frac{\log_2 \frac{I_0}{\varepsilon}}{2}$	Better

Input: $x = 150$ $\varepsilon = 10^{-5}$

How much improvement?

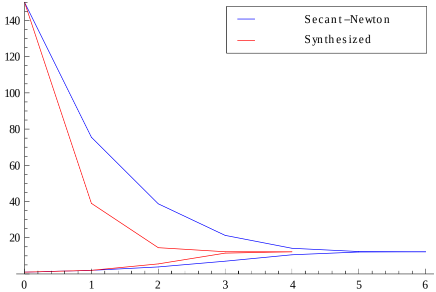
	Secant-Newton Map $R^*(I,x)$	Synthesized Map $\tilde{R}(I,x)$	
Original	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-U^2}{2U}\right]$	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-\frac{3}{4}U^2-\frac{1}{2}LU+\frac{1}{4}L^2}{U+L}\right]$	
Rewritten	$\left[\frac{x+LU}{L+U}, \frac{x}{U+U}+\frac{1}{4}(U+U)\right]$	$\left[\frac{x+LU}{L+U}, \frac{x}{U+L}+\frac{1}{4}(U+L)\right]$	
# of ops.	9	9	The same
Convergence	Quadratic	Quadratic	The same
Lipschitz	$\frac{1}{2}$	$\frac{1}{4}$	Better
# of loop iters.	$\log_2 \frac{I_0}{\varepsilon}$	$\frac{\log_2 \frac{I_0}{\varepsilon}}{2}$	Better

Input: $x = 150$ $\varepsilon = 10^{-5}$

How much improvement?

	Secant-Newton Map $R^*(I,x)$	Synthesized Map $\tilde{R}(I,x)$	
Original	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-U^2}{2U}\right]$	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-\frac{3}{4}U^2-\frac{1}{2}LU+\frac{1}{4}L^2}{U+L}\right]$	
Rewritten	$\left[\frac{x+LU}{L+U}, \frac{x}{U+U}+\frac{1}{4}(U+U)\right]$	$\left[\frac{x+LU}{L+U}, \frac{x}{U+L}+\frac{1}{4}(U+L)\right]$	
# of ops.	9	9	The same
Convergence	Quadratic	Quadratic	The same
Lipschitz	$\frac{1}{2}$	$\frac{1}{4}$	Better
# of loop iters.	$\log_2 \frac{I_0}{\varepsilon}$	$\frac{\log_2 \frac{I_0}{\varepsilon}}{2}$	Better

Input: $x = 150 \qquad \varepsilon = 10^{-5}$



Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p, q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0L^2 + p_1LU + p_2U^2}{p_3L + p_4U}\right)$$

$$U' = U + min\left(0, \frac{x + q_0U^2 + q_1UL + q_2L^2}{q_3U + q_4L}\right)$$

Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p, q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}\right)$$

$$U' = U + min\left(0, \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right)$$

Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p, q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}\right)$$

$$U' = U + min\left(0, \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right)$$

Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p, q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}\right)$$

$$U' = U + min\left(0, \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right)$$

Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p, q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}\right)$$
$$U' = U + min\left(0, \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right)$$

Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p, q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}\right)$$

$$U' = U + min\left(0, \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right)$$

Conclusions

(1) Carried out a case study on the synthesis of optimal numerical algorithms for square root computation.

(2) Semi-automatically synthesized algorithms faster than Secant-Newton.

(3) Current and future work:

(a) *remove* the condition $K(p,q)$ - *in revision*

(b) derive the result *completely* automatically - *simplification techniques for checking positivity of linear and quadratic polynomials (with parametric coefficients) over an interval: in revision*

(c) generalize the work to cubic, quartic, and eventually n-th root computation (see work of Bishop-Hong)

(d) consider the following class of refinement maps:

$$L' = L + max\left(0, \frac{x + p_0 L^2 + p_1 LU + p_2 U^2}{p_3 L + p_4 U}\right)$$
$$U' = U + min\left(0, \frac{x + q_0 U^2 + q_1 UL + q_2 L^2}{q_3 U + q_4 L}\right)$$