Synthesis of Optimal Numerical Algorithms using Real Quantifier Elimination

Mădălina Erașcu

West University of Timișoara and Institute e-Austria Timișoara
bvd. V. Parvan 4, Timișsoara, Romania

madalina.erascu@e-uvt.ro

January 11, 2018

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2$.

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge \ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge\ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge\ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge\ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge\ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2$.

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge\ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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
1975 —	Faster algorithms for special but important subclasses of formulas.

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\exists_y \, (x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2$.

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \ \wedge \ y^2 - 2x + 2 < 0)$

Output: $1 < x < 2.$

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

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

Problem QE over RCF

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

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

Toy Example

Input: $\underset{y}{\exists}\,(x^2 + y^2 - 4 < 0 \;\wedge\; y^2 - 2x + 2 < 0)$

Output: $1 < x < 2$.

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	
1975 —	Faster algorithms for special but important subclasses of formulas.	

Software:

- QEPCAD
- Redlog
- SyNRAC
- Mathematica (Reduce command)

QE over Real-Closed Fields (cont'd)

Applications (see *Stefan Ratschan – Applications of Quantified Constraint Solving over the Reals. Bibliography*):

- Electrical Engineering/Electronics
- Numerical analysis
- Control theory
- Computational Geometry/Motion Planning/Collision Detection
- Constraint Databases
- Theorem Proving in Real Geometry
- Program Analysis
- *Others*: camera motion, constraint logic programming, mechanical engineering, biology, automated theorem proving, optimization, termination of rewrite systems, flight control, hybrid systems, computer assisted proofs, parameter estimation, etc.

In this lecture: *application to synthesis of optimal numerical algorithms*

QE over Real-Closed Fields (cont'd)

Applications (see *Stefan Ratschan – Applications of Quantified Constraint Solving over the Reals. Bibliography*):

- Electrical Engineering/Electronics
- Numerical analysis
- Control theory
- Computational Geometry/Motion Planning/Collision Detection
- Constraint Databases
- Theorem Proving in Real Geometry
- Program Analysis
- *Others*: camera motion, constraint logic programming, mechanical engineering, biology, automated theorem proving, optimization, termination of rewrite systems, flight control, hybrid systems, computer assisted proofs, parameter estimation, etc.

In this lecture: *application to synthesis of optimal numerical algorithms*

QE over Real-Closed Fields (cont'd)

Applications (see *Stefan Ratschan – Applications of Quantified Constraint Solving over the Reals. Bibliography*):

- Electrical Engineering/Electronics
- Numerical analysis
- Control theory
- Computational Geometry/Motion Planning/Collision Detection
- Constraint Databases
- Theorem Proving in Real Geometry
- Program Analysis
- *Others*: camera motion, constraint logic programming, mechanical engineering, biology, automated theorem proving, optimization, termination of rewrite systems, flight control, hybrid systems, computer assisted proofs, parameter estimation, etc.

In this lecture: *application to synthesis of optimal numerical algorithms*

Numerical Algorithms

Problem:

in: x - real number

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \wedge\ y \in I\ \wedge\ f(y) = x$.

Algorithm schema: Interval refining

Initialize I

while width(I) $> \varepsilon$

$I \leftarrow R(I, x)$

return I

Numerical Algorithms

Problem:

in: x - real number

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \wedge\ y \in I\ \wedge\ f(y) = x$.

Algorithm schema: Interval refining

Initialize I

while width$(I) > \varepsilon$

$\quad I \leftarrow R(I, x)$

return I

Numerical Algorithms (Square Root)

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $\mathit{width}(I) < \varepsilon \ \wedge \ y \in I \ \wedge y^2 = x$.

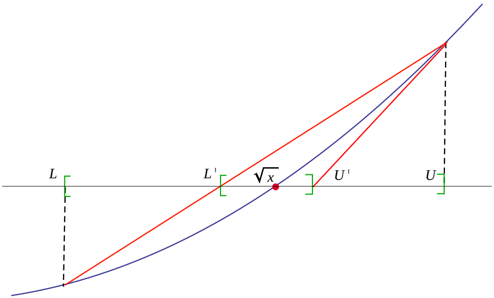
Algorithm schema: Interval refining

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

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)

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

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

Analysis:

- Partial Correctness
- Termination
- Complexity

Analysis – Partial Correctness

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $\mathit{width}(I) < \varepsilon \ \wedge\ y \in I \wedge y^2 = x$.

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

Partial Correctness:

$$\mathit{LoopInv}(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$

To show:

1. the invariant holds at the beginning of the loop
2. the invariant holds after one loop iteration
3. the invariant implies the postcondition

Analysis – Partial Correctness

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $\mathit{width}(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

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

while width$(I) > \varepsilon$

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

return I

Partial Correctness:

$$\mathit{LoopInv}(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$

To show:

1. the invariant holds at the beginning of the loop
2. the invariant holds after one loop iteration
3. the invariant implies the postcondition

Analysis – Partial Correctness

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

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

Partial Correctness:

$$LoopInv(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$

To show:

1. the invariant holds at the beginning of the loop
2. the invariant holds after one loop iteration
3. the invariant implies the postcondition

Analysis – Partial Correctness

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

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

while width$(I) > \varepsilon$

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

return I

Partial Correctness:

$$LoopInv(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$

To show:

1. the invariant holds at the beginning of the loop
2. the invariant holds after one loop iteration
3. the invariant implies the postcondition

Analysis – Partial Correctness

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

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

Partial Correctness:

$$LoopInv(L,U) \iff 0 < L \leq \sqrt{x} \leq U$$

To show:

1. the invariant holds at the beginning of the loop
2. the invariant holds after one loop iteration
3. the invariant implies the postcondition

Analysis – Termination

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

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

Termination:

$$LoopInv(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$

$$width(L, U) = U - L$$

To show:

- $\exists_{c \in (0,1)}$ s.t. $c = \sup_{\substack{L,U,x \\ LoopInv(L,U,x)}} \frac{width(f(L,U))}{width(L,U)}$

Analysis – Termination

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

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

while width(I) $> \varepsilon$

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

return I

Termination:

$$LoopInv(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$
$$width(L, U) = U - L$$

To show:

- $\underset{c \in (0,1)}{\exists}$ s.t. $c = \sup\limits_{\substack{L,U,x \\ LoopInv(L,U,x)}} \frac{width(f(L,U))}{width(L,U)}$

Analysis – Termination

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

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

Termination:

$$LoopInv(L, U) \iff 0 < L \leq \sqrt{x} \leq U$$
$$width(L, U) = U - L$$

To show:

- $\underset{c \in (0,1)}{\exists}$ s.t. $c = \sup\limits_{\substack{L,U,x \\ LoopInv(L,U,x)}} \frac{width(f(L,U))}{width(L,U)}$

Analysis – Complexity

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

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

while width(I) $> \varepsilon$

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

return I

Complexity:

The number of loop iterations n is given by

$$n = \left\lceil \frac{\log_2 \frac{\max(1,x) - \min(1,x)}{\varepsilon}}{\log_2 \frac{1}{c}} \right\rceil$$

where

$$c = \sup_{\substack{L, U, x \\ LoopInv(L, U, x)}} \frac{width(f(L, U))}{width(L, U)}$$

Analysis – Complexity

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

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

while width$(I) > \varepsilon$

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

return I

Complexity:

The number of loop iterations n is given by

$$n = \left\lceil \frac{\log_2 \frac{\max(1,x)-\min(1,x)}{\varepsilon}}{\log_2 \frac{1}{c}} \right\rceil$$

where

$$c = \sup_{\substack{L,U,x \\ LoopInv(L,U,x)}} \frac{width(f(L, U))}{width(L, U)}$$

Analysis – Complexity (cont'd)

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow I_0$

while width$(I) > \varepsilon$

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

return I

Complexity:

# iter	$U' - L'$
0	$\leq l_0$
1	$\leq c \cdot l_0$
2	$\leq c^2 \cdot l_0$
...	...
n	$\leq c^n \cdot l_0 \quad \leq \varepsilon$

$$\Rightarrow n \leq \log_c \frac{\varepsilon}{l_0} \quad \Rightarrow \quad n \leq \left\lceil \log_c \frac{\varepsilon}{l_0} \right\rceil = \left\lceil \frac{\log_2 \frac{\varepsilon}{l_0}}{\log_2 c} \right\rceil = \left\lceil \frac{\log_2 \frac{l_0}{\varepsilon}}{\log_2 \frac{1}{c}} \right\rceil$$

We take $n = \left\lceil \frac{\log_2 \frac{l_0}{\varepsilon}}{\log_2 \frac{1}{c}} \right\rceil$

Analysis – Complexity (cont'd)

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

$I \leftarrow I_0$

while width$(I) > \varepsilon$

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

return I

Complexity:

# iter	$U' - L'$
0	$\leq I_0$
1	$\leq c \cdot I_0$
2	$\leq c^2 \cdot I_0$
...	...
n	$\leq c^n \cdot I_0 \quad \leq \varepsilon$

$$\Rightarrow n \leq \log_c \frac{\varepsilon}{I_0} \quad \Rightarrow \quad n \leq \left\lceil \log_c \frac{\varepsilon}{I_0} \right\rceil = \left\lceil \frac{\log_2 \frac{\varepsilon}{I_0}}{\log_2 c} \right\rceil = \left\lceil \frac{\log_2 \frac{I_0}{\varepsilon}}{\log_2 \frac{1}{c}} \right\rceil$$

We take $n = \left\lceil \frac{\log_2 \frac{I_0}{\varepsilon}}{\log_2 \frac{1}{c}} \right\rceil$

Analysis – Complexity (cont'd)

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

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

Complexity:

Rate of convergence

$$\underset{x>0}{\forall} \ \underset{c>0}{\exists} \ \underset{\substack{L,U \\ LoopInv(L,U)}}{\forall} \quad width(f(L, U)) \leq c(U - L)^2$$

Analysis – Complexity (cont'd)

Problem:

in: x - real number greater than 0

ε - error bound

out: an interval I s.t. $width(I) < \varepsilon \ \wedge \ y \in I \wedge y^2 = x$.

Algorithm schema: Interval refining

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

while width(I) $> \varepsilon$

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

return I

Complexity:

Rate of convergence

$$\underset{x>0}{\forall} \ \underset{c>0}{\exists} \ \underset{\substack{L,U \\ LoopInv(L,U)}}{\forall} \ width(f(L,U)) \leq c(U-L)^2$$

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 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

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

Minimize

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

Subject to

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

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

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

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

Minimize

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

Subject to

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

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

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

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

Minimize

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

Subject to

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

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

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

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

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

Minimize

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

Subject to

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

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

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

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

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

Minimize

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

Subject to

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

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

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

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

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

Minimize

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

Subject to

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

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

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

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

Minimize

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

Subject to

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

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

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

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

Minimize

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

Subject to

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

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

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

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

Minimize

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

Subject to

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

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

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

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

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

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

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

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

Numerical Algorithms (Square Root) Synthesis Optimal by QE

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

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

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

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

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

Numerical Algorithms (Square Root) Synthesis Optimal by QE

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

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

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

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

Strategies:

1. divide the QE problems into several simpler ones
2. apply state of the art QE software
3. manual simplification on the remaining ones

Numerical Algorithms (Square Root) Synthesis Optimal by QE

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

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

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

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

Strategies:

1. divide the QE problems into several simpler ones
2. apply state of the art QE software
3. manual simplification on the remaining ones

Numerical Algorithms (Square Root) Synthesis Optimal by QE

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

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

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

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

Strategies:

1. divide the QE problems into several simpler ones
2. apply state of the art QE software
3. manual simplification on the remaining ones

Numerical Algorithms (Square Root) Synthesis Optimal by QE

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

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

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

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

Strategies:

1. divide the QE problems into several simpler ones
2. apply state of the art QE software
3. manual simplification on the remaining ones

Numerical Algorithms (Square Root) Synthesis Optimal by QE

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

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

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

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

Strategies:

1. divide the QE problems into several simpler ones
2. apply state of the art QE software
3. manual simplification on the remaining ones

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

1. Find $Correctness(p, q)$

   Strategies:

   - Split conjunction in the goal
   - Eliminate universal quantifier using the properties of convex functions

2. Compute $E(p, q) = \sup\limits_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U'-L'}{U-L}$, where

   $$E(p, q) = E_j(p, q) \text{ if } G_j(p, q)$$

   $$G_j(p, q) \text{– a conjunction of equations/inequalities in } p, q$$

   Strategies:

   - Variable elimination using monotonicity of functions

3. Find the minimum of $E_j(p, q)$ over
   $\bigwedge\limits_j Correctness(p, q) \wedge QuadraticConv(p, q) \wedge G_j(p, q)$.

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

1. Find *Correctness*(p, q)

   Strategies:

   - Split conjunction in the goal
   - Eliminate universal quantifier using the properties of convex functions

2. Compute $E(p, q) = \sup\limits_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U'-L'}{U-L}$, where

   $$E(p, q) = E_j(p, q) \text{ if } G_j(p, q)$$

   $G_j(p, q)$– a conjunction of equations/inequalities in p, q

   Strategies:

   - Variable elimination using monotonicity of functions

3. Find the minimum of $E_j(p, q)$ over $\bigwedge\limits_j \mathit{Correctness}(p, q) \wedge \mathit{QuadraticConv}(p, q) \wedge G_j(p, q)$.

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

1. Find *Correctness*(p, q)

   Strategies:

   - Split conjunction in the goal
   - Eliminate universal quantifier using the properties of convex functions

2. Compute $E(p,q) = \sup\limits_{\substack{L,U,x \\ 0<L\leq\sqrt{x}\leq U \\ L\neq U}} \frac{U'-L'}{U-L}$, where

   $$E(p,q) = E_j(p,q) \text{ if } G_j(p,q)$$

   $$G_j(p,q) \text{– a conjunction of equations/inequalities in } p, q$$

   Strategies:

   - Variable elimination using monotonicity of functions

3. Find the minimum of $E_j(p,q)$ over $\bigwedge\limits_j \mathit{Correctness}(p,q) \wedge \mathit{QuadraticConv}(p,q) \wedge G_j(p,q)$.

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

1. Find $Correctness(p, q)$

   Strategies:

   - Split conjunction in the goal
   - Eliminate universal quantifier using the properties of convex functions

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

   $$E(p, q) = E_j(p, q) \text{ if } G_j(p, q)$$

   $$G_j(p, q) \text{– a conjunction of equations/inequalities in } p, q$$

   Strategies:

   - Variable elimination using monotonicity of functions

3. Find the minimum of $E_j(p, q)$ over $\bigwedge_j Correctness(p, q) \wedge QuadraticConv(p, q) \wedge G_j(p, q)$.

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

Key steps in the proof:

$$a_3>0 \wedge a_4 \geq 0 \wedge \underset{\substack{L,W \\ 0<L\leq L+W}}{\forall} \; \underset{\substack{y \\ L\leq y\leq L+W}}{\forall} \; y^2 - y(a_3L+a_4W) - L^2 + a_1LW + a_2W^2 + L(a_3L+a_4W) \leq 0 \iff \ldots$$

$$b_3>0 \wedge b_4 \geq 0 \wedge \underset{\substack{L,W \\ 0<L\leq L+W}}{\forall} \; \underset{\substack{y \\ L\leq y\leq L+W}}{\forall} \; y^2 - y(b_3L+b_4W) - L^2 + b_1LW + b_2W^2 + (L+W)(b_3L+b_4W) \geq 0 \iff \ldots$$

Numerical Algorithms (Square Root) Synthesis Optimal **by QE** (cont'd)

Key steps in the proof:

$$a_3 > 0 \land a_4 \geq 0 \land \underset{\substack{L,W \\ 0<L\leq L+W}}{\forall} \; \underset{\substack{y \\ L\leq y\leq L+W}}{\forall} \; y^2 - y(a_3 L + a_4 W) - L^2 + a_1 LW + a_2 W^2 + L(a_3 L + a_4 W) \leq 0 \iff \ldots$$

$$b_3 > 0 \land b_4 \geq 0 \land \underset{\substack{L,W \\ 0<L\leq L+W}}{\forall} \; \underset{\substack{y \\ L\leq y\leq L+W}}{\forall} \; y^2 - y(b_3 L + b_4 W) - L^2 + b_1 LW + b_2 W^2 + (L+W)(b_3 L + b_4 W) \geq 0 \iff \ldots$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

Key steps in the proof:

$$a_3 > 0 \wedge a_4 \geq 0 \wedge \underset{\substack{L,W \\ 0<L\leq L+W}}{\forall} \underset{\substack{y \\ L\leq y\leq L+W}}{\forall} y^2 - y(a_3 L + a_4 W) - L^2 + a_1 LW + a_2 W^2 + L(a_3 L + a_4 W) \leq 0 \iff \ldots$$

$$b_3 > 0 \wedge b_4 \geq 0 \wedge \underset{\substack{L,W \\ 0<L\leq L+W}}{\forall} \underset{\substack{y \\ L\leq y\leq L+W}}{\forall} y^2 - y(b_3 L + b_4 W) - L^2 + b_1 LW + b_2 W^2 + (L+W)(b_3 L + b_4 W) \geq 0 \iff \ldots$$

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

Key steps in the proof:

$$
\begin{aligned}
E(p,q) &= \sup_{\substack{0<L\le\sqrt{x}\le U\\ L\ne U}} \left[\frac{U'-L'}{U-L}\right]\\
&\quad \cdots\\
&= \begin{cases} h_1 + \left(\frac{c_1 d_1}{4}\right)^2 \sup\limits_{\substack{T\\T>0}} \frac{a_1 T + b_1}{(T+c_1)(T+d_1)} & \text{if} \quad d_1 \ge c_1 > 0 \\ h_2 + \left(\frac{c_2 d_2}{4}\right)^2 \sup\limits_{\substack{T\\T>0}} \frac{a_2 T + b_2}{(T+c_2)(T+d_2)} & \text{if} \quad d_2 \ge c_2 > 0 \end{cases}
\end{aligned}
$$

1. $\sup\limits_{\substack{T\\T>0}} \frac{aT+b}{(T+c)(T+d)} \ge 0$

2. $\sup\limits_{\substack{T\\T>0}} \frac{aT+b}{(T+c)(T+d)} = 0$ iff $a \le 0 \wedge b \le 0$

Numerical Algorithms (Square Root) Synthesis Optimal **by QE** (cont'd)

Key steps in the proof:

$$
\begin{aligned}
E(p,q) \quad &= \quad \sup_{\substack{0<L\le\sqrt{x}\le U\\ L\neq U}} \left[\frac{U'-L'}{U-L}\right] \\
&\quad\;\; \cdots \\
&= \quad \begin{cases}
h_1 + \left(\frac{c_1 d_1}{4}\right)^2 \sup\limits_{\substack{T\\ T>0}} \frac{a_1 T + b_1}{(T+c_1)(T+d_1)} & \text{if} \quad d_1 \ge c_1 > 0 \\
h_2 + \left(\frac{c_2 d_2}{4}\right)^2 \sup\limits_{\substack{T\\ T>0}} \frac{a_2 T + b_2}{(T+c_2)(T+d_2)} & \text{if} \quad d_2 \ge c_2 > 0
\end{cases}
\end{aligned}
$$

1. $\sup\limits_{\substack{T\\ T>0}} \frac{aT+b}{(T+c)(T+d)} \ge 0$

2. $\sup\limits_{\substack{T\\ T>0}} \frac{aT+b}{(T+c)(T+d)} = 0$ iff $a \le 0 \wedge b \le 0$

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

Key steps in the proof:

$$
\begin{aligned}
E(p,q) &= \sup_{\substack{0<L\leq\sqrt{x}\leq U\\ L\neq U}} \left[\frac{U'-L'}{U-L}\right] \\
&\quad \cdots \\
&= \begin{cases}
h_1 + \left(\frac{c_1 d_1}{4}\right)^2 \sup\limits_{\substack{T\\ T>0}} \frac{a_1 T + b_1}{(T+c_1)(T+d_1)} & \text{if} \quad d_1 \geq c_1 > 0 \\
h_2 + \left(\frac{c_2 d_2}{4}\right)^2 \sup\limits_{\substack{T\\ T>0}} \frac{a_2 T + b_2}{(T+c_2)(T+d_2)} & \text{if} \quad d_2 \geq c_2 > 0
\end{cases}
\end{aligned}
$$

1. $\sup\limits_{\substack{T\\ T>0}} \frac{aT+b}{(T+c)(T+d)} \geq 0$

2. $\sup\limits_{\substack{T\\ T>0}} \frac{aT+b}{(T+c)(T+d)} = 0$ iff $a \leq 0 \wedge b \leq 0$

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

Key steps in the proof:

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

$$\ldots$$

$$= \begin{cases} h_1 + \left(\frac{c_1 d_1}{4}\right)^2 \sup_{\substack{T \\ T>0}} \frac{a_1 T + b_1}{(T+c_1)(T+d_1)} & \text{if} \quad d_1 \geq c_1 > 0 \\ h_2 + \left(\frac{c_2 d_2}{4}\right)^2 \sup_{\substack{T \\ T>0}} \frac{a_2 T + b_2}{(T+c_2)(T+d_2)} & \text{if} \quad d_2 \geq c_2 > 0 \end{cases}$$

1. $\sup_{\substack{T \\ T>0}} \frac{aT+b}{(T+c)(T+d)} \geq 0$
2. $\sup_{\substack{T \\ T>0}} \frac{aT+b}{(T+c)(T+d)} = 0$ iff $a \leq 0 \wedge b \leq 0$

Numerical Algorithms (Square Root) Synthesis Optimal by QE (cont'd)

Main Result:

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

(b) $E(p,q) = \frac{1}{4}$ iff $p = (-1,0,0,1,1) \wedge q = \left(-\frac{3}{4}, -\frac{1}{2}, \frac{1}{4}, 1, 1\right)$

In other words

$$L' = L + \frac{x - L^2}{L + U}$$

$$U' = U + \frac{x - \frac{3}{4}U^2 - \frac{1}{2}LU + \frac{1}{4}L^2}{U + L}$$

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

How much improvement?

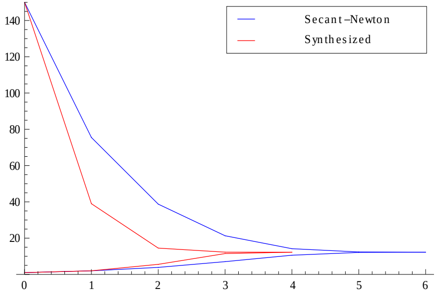
	Secant-Newton Map $R^*(I,x)$	Synthesized Map $\tilde{R}(I,x)$	
Original	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-U^2}{2U}\right]$	$\left[L+\frac{x-L^2}{L+U}, U+\frac{x-\frac{3}{4}U^2-\frac{1}{2}LU+\frac{1}{4}L^2}{U+L}\right]$	
Rewritten	$\left[\frac{x+LU}{L+U}, \frac{x}{U+U}+\frac{1}{4}(U+U)\right]$	$\left[\frac{x+LU}{L+U}, \frac{x}{U+L}+\frac{1}{4}(U+L)\right]$	
# 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: $x = 150 \qquad \varepsilon = 10^{-5}$

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 and Future Work

Conclusions:

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

(2) Semi-automatically an algorithm faster than Secant-Newton.

Current and future work:

(b) derive the result *completely* automatically

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

Conclusions and Future Work

Conclusions:

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

(2) Semi-automatically an algorithm faster than Secant-Newton.

Current and future work:

(b) derive the result *completely* automatically

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

Conclusions and Future Work

Conclusions:

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

(2) Semi-automatically an algorithm faster than Secant-Newton.

Current and future work:

(b) derive the result *completely* automatically

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

Conclusions and Future Work

Conclusions:

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

(2) Semi-automatically an algorithm faster than Secant-Newton.

Current and future work:

(b) derive the result *completely* automatically

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

Conclusions and Future Work

Conclusions:

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

(2) Semi-automatically an algorithm faster than Secant-Newton.

Current and future work:

(b) derive the result *completely* automatically

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