Programming Techniques for Exact Real Arithmetic

Andrej Bauer
Faculty of Mathematics and Physics
University of Ljubljana, Slovenia

(joint work with Ivo List & Paul Taylor)

SCAN 2014, Würzburg, September 2014

In this talk

*We present a mathematical language **Marshall** which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

In this talk

*We present a mathematical language **Marshall** which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

In this talk

*We present a mathematical language **Marshall** which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.

In this talk

*We present a mathematical language **Marshall** which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.
- *How* to compute?

In this talk

We present a mathematical language ***Marshall*** *which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.
- *How* to compute?
    - We present a simple-minded execution strategy

In this talk

*We present a mathematical language **Marshall** which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.
- *How* to compute?
  - We present a simple-minded execution strategy
  - Many possibilities for optimization

In this talk

*We present a mathematical language **Marshall** which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.
- *How* to compute?
  - We present a simple-minded execution strategy
  - Many possibilities for optimization
- Applications:

In this talk

We present a mathematical language ***Marshall*** *which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.
- *How* to compute?
  - We present a simple-minded execution strategy
  - Many possibilities for optimization
- Applications:
  - specification of computational problems

In this talk

We present a mathematical language ***Marshall*** *which is powerful enough to let us talk about real analysis, but also simple enough to be an efficient programming language.*

- *Descriptive* language – *what* to compute.
- *How* to compute?
  - We present a simple-minded execution strategy
  - Many possibilities for optimization
- Applications:
  - specification of computational problems
  - verification of safety and liveness properties

Foundations: Abstract Stone Duality

- Our language is based on *Abstract Stone Duality* (ASD) by Paul Taylor.

Foundations: Abstract Stone Duality

- Our language is based on *Abstract Stone Duality* (ASD) by Paul Taylor.
- ASD is a variant of λ-calculus which directly axiomatizes spaces and continuous maps.

Foundations: Abstract Stone Duality

- Our language is based on *Abstract Stone Duality* (ASD) by Paul Taylor.
- ASD is a variant of λ-calculus which directly axiomatizes spaces and continuous maps.
- We use a fragment of ASD which can be understood on its own.

Foundations: Abstract Stone Duality

- Our language is based on *Abstract Stone Duality* (ASD) by Paul Taylor.
- ASD is a variant of λ-calculus which directly axiomatizes spaces and continuous maps.
- We use a fragment of ASD which can be understood on its own.
- Further material: `http://www.paultaylor.eu/ASD/`

Axioms for real numbers

The real numbers $\mathbb{R}$ are:

- an ordered field ("can compute with reals")

Axioms for real numbers

The real numbers $\mathbb{R}$ are:

- an ordered field (“can compute with reals”)
- with Archimedean property (“can obtain approximations”)

Axioms for real numbers

The real numbers $\mathbb{R}$ are:

- an ordered field ("can compute with reals")
- with Archimedean property ("can obtain approximations")
- Dedekind complete ("can use iterative methods")

Axioms for real numbers

The real numbers $\mathbb{R}$ are:

- an ordered field ("can compute with reals")
- with Archimedean property ("can obtain approximations")
- Dedekind complete ("can use iterative methods")
- overt Hausdorff space ("can search for a witness")

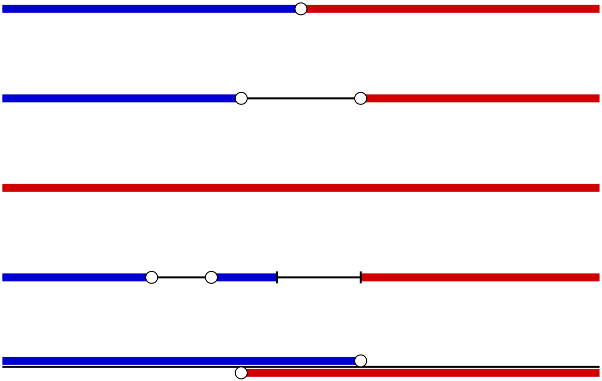
Axioms for real numbers

The real numbers $\mathbb{R}$ are:

- an ordered field ("can compute with reals")
- with Archimedean property ("can obtain approximations")
- Dedekind complete ("can use iterative methods")
- overt Hausdorff space ("can search for a witness")
- and $[a, b]$ is compact ("can verify something holds")

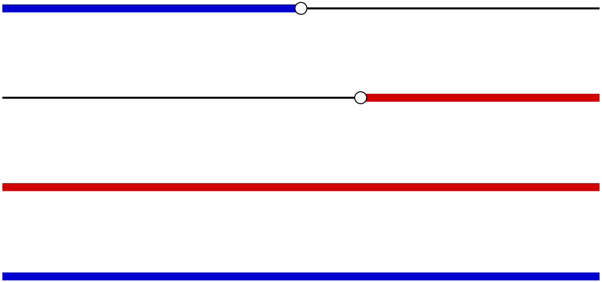
Dedekind cuts

A *cut* is a pair of *rounded*, *bounded*, *disjoint*, and *located* open sets.



Lower and upper reals

By taking the lower rounded sets we obtain the *lower reals*, and similarly for *upper reals*. These are more fundamental than reals.



Examples of cuts

- A number a determines a cut, which determines a:

$$a = (\mathsf{cut}\ x\ \mathsf{left}\ x < a\ \mathsf{right}\ a < x)$$

Examples of cuts

- A number a determines a cut, which determines a:

$$a = (\mathsf{cut}\ x\ \mathsf{left}\ x < a\ \mathsf{right}\ a < x)$$

- $\sqrt{a}$ is the cut

$$\mathsf{cut}\ x\ \mathsf{left}\ (x < 0 \vee x^2 < a)\ \mathsf{right}\ (x > 0 \wedge x^2 > a)$$

Examples of cuts

- A number a determines a cut, which determines a:

$$a = (\mathsf{cut}\ x\ \mathsf{left}\ x < a\ \mathsf{right}\ a < x)$$

- $\sqrt{a}$ is the cut

$$\mathsf{cut}\ x\ \mathsf{left}\ (x < 0 \lor x^2 < a)\ \mathsf{right}\ (x > 0 \land x^2 > a)$$

- Exercise:

$$\mathsf{cut}\ x\ \mathsf{left}\ (x < -a \lor x < a)\ \mathsf{right}\ (-a < x \land a < x)$$

Examples of cuts

- A number a determines a cut, which determines a:

$$a = (\mathsf{cut}\ x\ \mathsf{left}\ x < a\ \mathsf{right}\ a < x)$$

- $\sqrt{a}$ is the cut

$$\mathsf{cut}\ x\ \mathsf{left}\ (x < 0 \lor x^2 < a)\ \mathsf{right}\ (x > 0 \land x^2 > a)$$

- Exercise:

$$\mathsf{cut}\ x\ \mathsf{left}\ (x < -a \lor x < a)\ \mathsf{right}\ (-a < x \land a < x)$$

- The full notation for cuts is

$$\mathsf{cut}\ x : [a, b]\ \mathsf{left}\ \phi(x)\ \mathsf{right}\ \psi(x)$$

  This means that the cut determines a number in $[a, b]$.

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$
- Semidecidable order relation $<$ on $\mathbb{R}$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$
- Semidecidable order relation $<$ on $\mathbb{R}$
- Logic:

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+, -, \times, /$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$
- Semidecidable order relation $<$ on $\mathbb{R}$
- Logic:
  - truth $\top$ and falsehood $\bot$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$
- Semidecidable order relation $<$ on $\mathbb{R}$
- Logic:
  - truth $\top$ and falsehood $\bot$
  - connectives $\wedge$ and $\vee$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+$, $-$, $\times$, $/$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$
- Semidecidable order relation $<$ on $\mathbb{R}$
- Logic:
  - truth $\top$ and falsehood $\bot$
  - connectives $\wedge$ and $\vee$
  - existential quantifiers:

$$\exists x : \mathbb{R}, \quad \exists x : [a, b], \quad \exists x : (a, b), \quad \exists n : \mathbb{N}, \quad \exists q : \mathbb{Q}$$

A language for real analysis

- Number types $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$
- Arithmetic $+, -, \times, /$
- Decidable equality $=$ and decidable order $<$ on $\mathbb{N}$ and $\mathbb{Q}$
- General recursion on $\mathbb{N}$
- Semidecidable order relation $<$ on $\mathbb{R}$
- Logic:
  - truth $\top$ and falsehood $\bot$
  - connectives $\wedge$ and $\vee$
  - existential quantifiers:

$$\exists x : \mathbb{R}, \quad \exists x : [a, b], \quad \exists x : (a, b), \quad \exists n : \mathbb{N}, \quad \exists q : \mathbb{Q}$$

  - universal quantifier: $\quad \forall x : [a, b]$

"Topologic"

- A logical formula $\phi(x)$ where $x \in A$ has two readings:

“Topologic”

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A

“Topologic”

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:

"Topologic"

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:
    - an open subset of A: $\phi(x) \iff \top$

“Topologic”

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:
    - an open subset of A: $\phi(x) \iff \top$
    - a closed subset of A: $\phi(x) \iff \bot$

"Topologic"

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:
    - an open subset of A: $\phi(x) \iff \top$
    - a closed subset of A: $\phi(x) \iff \bot$
- In particular, a formula ϕ without parameters is

"Topologic"

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:
    - an open subset of A: $\phi(x) \iff \top$
    - a closed subset of A: $\phi(x) \iff \bot$
- In particular, a formula ϕ without parameters is
  - *logically*, a truth value

“Topologic”

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:
    - an open subset of A: $\phi(x) \iff \top$
    - a closed subset of A: $\phi(x) \iff \bot$
- In particular, a formula ϕ without parameters is
  - *logically*, a truth value
  - *topologically*, an element of Sierpinski space $\Sigma = \{\bot, \top\}$

“Topologic”

- A logical formula $\phi(x)$ where $x \in A$ has two readings:
  - *logical*: a predicate on A
  - *topological*:
    - an open subset of A: $\phi(x) \iff \top$
    - a closed subset of A: $\phi(x) \iff \bot$
- In particular, a formula ϕ without parameters is
  - *logically*, a truth value
  - *topologically*, an element of Sierpinski space $\Sigma = \{\bot, \top\}$
- We use this to express topological and analytic notions *logically*.

Example: $\mathbb{R}$ is locally compact

- Classically: for open $U \subseteq \mathbb{R}$ and $x \in \mathbb{R}$,

$$x \in U \iff \exists\, d, u \in \mathbb{Q} \,.\, x \in (d, u) \subseteq [d, u] \subseteq U$$

Example: $\mathbb{R}$ is locally compact

- Classically: for open $U \subseteq \mathbb{R}$ and $x \in \mathbb{R}$,

$$x \in U \iff \exists\, d, u \in \mathbb{Q} \,.\, x \in (d, u) \subseteq [d, u] \subseteq U$$

- Topologically: for $\phi : \mathbb{R} \to \Sigma$ and $x : \mathbb{R}$,

$$\phi(x) \iff \exists\, d, u \in \mathbb{Q} \,.\, d < x < u \wedge \forall\, y \in [d, u] \,.\, \phi(y)$$

Example: $[0,1]$ is connected

- Classically: for open $U, V \subseteq [0,1]$, if

$$U \cap V = \emptyset \quad \text{and} \quad U \cup V = [0,1]$$

then $U = [0,1]$ or $V = [0,1]$.

Example: $[0,1]$ is connected

- Classically: for open $U, V \subseteq [0,1]$, if

$$U \cap V = \emptyset \quad \text{and} \quad U \cup V = [0,1]$$

then $U = [0,1]$ or $V = [0,1]$.
- Topologically: for $\phi, \psi : [0,1] \to \Sigma$, if

$$\begin{aligned} (\exists\, x \in [0,1] \,.\, \phi(x) \wedge \psi(x)) &\iff \bot \quad \text{and} \\ (\forall\, x \in [0,1] \,.\, \phi(x) \vee \psi(x)) &\iff \top \end{aligned}$$

then $(\forall\, x \in [0,1] \,.\, \phi(x)) \vee (\forall\, x \in [0,1] \,.\, \psi(x))$.

$\forall\exists$ statements

$$\forall\, x \in A\,.\,\exists\, y \in B\,.\,\phi(x, y)$$

$\forall\exists$ statements

$$\forall\, x \in A \,.\, \exists\, y \in B \,.\, \phi(x, y)$$

- "For every parameter x there is solution y."

$\forall\exists$ statements

$$\forall x \in A \,.\, \exists y \in B \,.\, \phi(x, y)$$

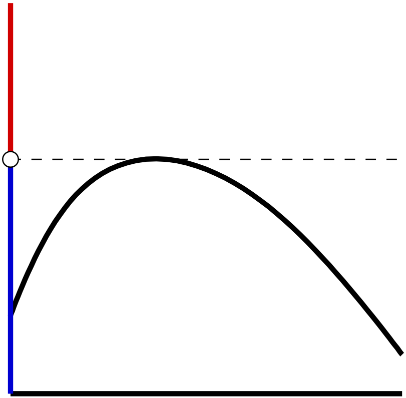
- "For every parameter x there is solution y."
- "In every state x good thing y happens."

$\forall\exists$ statements

$$\forall x \in A \,.\, \exists y \in B \,.\, \phi(x, y)$$

- "For every parameter x there is solution y."
- "In every state x good thing y happens."
- Note: A must be *overt* and B *compact*.

The maximum of $f : [0,1] \to \mathbb{R}$



cut x left $(\exists\, y \in [0,1]\,.\, x < f(y))$
right $(\forall\, z \in [0,1]\,.\, f(z) < x)$

Cauchy completeness

- A *rapid* Cauchy sequence $(a_n)_n$ satisfies

$$|a_{n+1} - a_n| < 2^{-n}.$$

Cauchy completeness

- A *rapid* Cauchy sequence $(a_n)_n$ satisfies

$$|a_{n+1} - a_n| < 2^{-n}.$$

- Its limit is the cut

$$\begin{aligned} \mathsf{cut}\ x\ \ \mathsf{left}\ & (\exists\, n \in \mathbb{N} \,.\, x < a_n - 2^{-n+1}) \\ \mathsf{right}\ & (\exists\, n \in \mathbb{N} \,.\, a_n + 2^{-n+1} < x) \end{aligned}$$

From mathematics to programming

- We would like to *compute* with our language.

From mathematics to programming

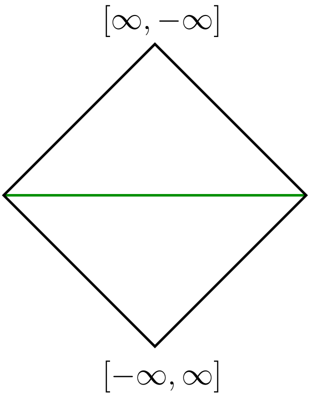
- We would like to *compute* with our language.
- We limit attention to logic and $\mathbb{R}$.

From mathematics to programming

- We would like to *compute* with our language.
- We limit attention to logic and $\mathbb{R}$.
- Not surprisingly, we compute with (improper) intervals.

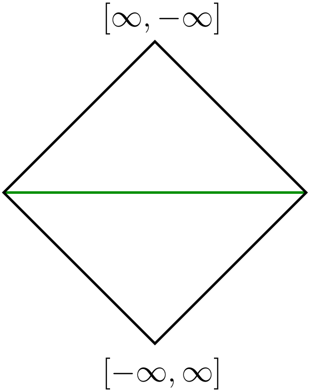
The interval lattice L

- The lattice of **pairs** $[a, b]$, where a is *upper* and b *lower real*.



The interval lattice L

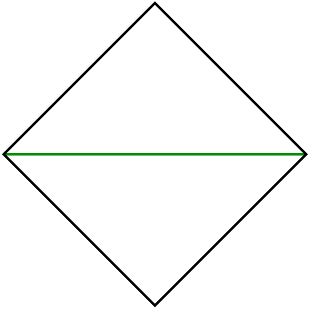
- The lattice of **pairs** $[a, b]$, where a is *upper* and b *lower real*.
- Ordered by $[a, b] \sqsubseteq [c, d] \iff a \leq c \wedge d \leq b$.



The interval lattice L

- The lattice of **pairs** $[a, b]$, where a is *upper* and b *lower real*.
- Ordered by $[a, b] \sqsubseteq [c, d] \iff a \leq c \wedge d \leq b$.
- The lattice contains $\mathbb{R}$ as $[a, a]$.

$[\infty, -\infty]$



$[-\infty, \infty]$

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:
  - L is equipped with the Scott topology

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:
  - L is equipped with the Scott topology
  - *any* continuous extension is acceptable

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:
  - L is equipped with the Scott topology
  - *any* continuous extension is acceptable
  - (improper) intervals are understood *order-theoretically*

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:
  - L is equipped with the Scott topology
  - *any* continuous extension is acceptable
  - (improper) intervals are understood *order-theoretically*
- The interesting case is *Kaucher multiplication*.

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:
  - L is equipped with the Scott topology
  - *any* continuous extension is acceptable
  - (improper) intervals are understood *order-theoretically*
- The interesting case is *Kaucher multiplication*.
- Given an arithmetical expression e we compute its *lower* and *upper* approximants e^- and e^+ in L:

$$e^- \sqsubseteq e \sqsubseteq e^+.$$

Extending arithmetic to L

- Extend operations from $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ to $L \times L \to L$:
  - L is equipped with the Scott topology
  - *any* continuous extension is acceptable
  - (improper) intervals are understood *order-theoretically*
- The interesting case is *Kaucher multiplication*.
- Given an arithmetical expression e we compute its *lower* and *upper* approximants e^- and e^+ in L:

$$e^- \sqsubseteq e \sqsubseteq e^+.$$

- We also extend $<$ to $L \times L \to \Sigma$:

$$[a, b] < [c, d] \iff b < c$$

Lower and upper approximants

- For each sentence ϕ we define a *lower* and *upper approximants* $\phi^-, \phi^+ \in \{\top, \bot\}$ such that

$$\phi^- \implies \phi \implies \phi^+.$$

Lower and upper approximants

- For each sentence ϕ we define a *lower* and *upper approximants* $\phi^-, \phi^+ \in \{\top, \bot\}$ such that

$$\phi^- \implies \phi \implies \phi^+.$$

- The approximants should be easy to compute.

Lower and upper approximants

- For each sentence ϕ we define a *lower* and *upper approximants* $\phi^-, \phi^+ \in \{\top, \bot\}$ such that

$$\phi^- \implies \phi \implies \phi^+.$$

- The approximants should be easy to compute.
- If $\phi^- = \top$ then $\phi = \top$, and if $\phi^+ = \bot$ then $\phi = \bot$.

Lower and upper approximants

- For each sentence ϕ we define a *lower* and *upper approximants* $\phi^-, \phi^+ \in \{\top, \bot\}$ such that

$$\phi^- \implies \phi \implies \phi^+.$$

- The approximants should be easy to compute.
- If $\phi^- = \top$ then $\phi = \top$, and if $\phi^+ = \bot$ then $\phi = \bot$.
- Easy cases:

$$\begin{aligned} \bot^- &= \bot & \bot^+ &= \bot \\ \top^- &= \top & \top^+ &= \top \\ (\phi \wedge \psi)^- &= \phi^- \wedge \psi^- & (\phi \wedge \psi)^+ &= \phi^+ \wedge \psi^+ \\ (\phi \vee \psi)^- &= \phi^- \vee \psi^- & (\phi \vee \psi)^+ &= \phi^+ \vee \psi^+ \\ (e_1 < e_2)^- &= (e_1^- < e_2^-) & (e_1 < e_2)^+ &= (e_1^+ < e_2^+). \end{aligned}$$

Approximants for cuts and quantifiers

- Cuts:

$$(\mathsf{cut}\ x : [a,b]\ \mathsf{left}\ \phi(x)\ \mathsf{right}\ \psi(x))^{-} = [a,b]$$
$$(\mathsf{cut}\ x : [a,b]\ \mathsf{left}\ \phi(x)\ \mathsf{right}\ \psi(x))^{+} = [b,a]$$

Approximants for cuts and quantifiers

- Cuts:

$$(\mathsf{cut}\ x : [a,b]\ \mathsf{left}\ \phi(x)\ \mathsf{right}\ \psi(x))^- = [a,b]$$
$$(\mathsf{cut}\ x : [a,b]\ \mathsf{left}\ \phi(x)\ \mathsf{right}\ \psi(x))^+ = [b,a]$$

- Quantifiers:

$$\phi([a,b]) \implies \forall\, x \in [a,b]\,.\,\phi(x) \implies \phi(\tfrac{a+b}{2})$$

$$\phi(\tfrac{a+b}{2}) \implies \exists\, x \in [a,b]\,.\,\phi(x) \implies \phi([b,a])$$

Refinement

$$\phi^- \implies \phi \implies \phi^+$$

- If $\phi^- = \bot$ and $\phi^+ = \top$ we cannot say much about ϕ.

Refinement

$$\phi^- \implies \phi \implies \phi^+$$

- If $\phi^- = \bot$ and $\phi^+ = \top$ we cannot say much about ϕ.
- To make progress, we *refine* ϕ to an equivalent formula in which quantifers range over smaller intervals.

Refinement

$$\phi^- \implies \phi \implies \phi^+$$

- If $\phi^- = \bot$ and $\phi^+ = \top$ we cannot say much about ϕ.
- To make progress, we *refine* ϕ to an equivalent formula in which quantifers range over smaller intervals.
- A simple strategy is to split quantified intervals in halves:

Refinement

$$\phi^- \implies \phi \implies \phi^+$$

- If $\phi^- = \bot$ and $\phi^+ = \top$ we cannot say much about ϕ.
- To make progress, we *refine* ϕ to an equivalent formula in which quantifers range over smaller intervals.
- A simple strategy is to split quantified intervals in halves:
  - $\forall x \in [a, b] \,.\, \phi(x)$ is refined to

$$(\forall x \in [a, \tfrac{a+b}{2}] \,.\, \phi(x)) \land (\forall x \in [\tfrac{a+b}{2}, b] \,.\, \phi(x))$$

Refinement

$$\phi^- \implies \phi \implies \phi^+$$

- If $\phi^- = \bot$ and $\phi^+ = \top$ we cannot say much about ϕ.
- To make progress, we *refine* ϕ to an equivalent formula in which quantifers range over smaller intervals.
- A simple strategy is to split quantified intervals in halves:
  - $\forall\, x \in [a, b] \,.\, \phi(x)$ is refined to

$$(\forall\, x \in [a, \tfrac{a+b}{2}] \,.\, \phi(x)) \wedge (\forall\, x \in [\tfrac{a+b}{2}, b] \,.\, \phi(x))$$

  - $\exists\, x \in [a, b] \,.\, \phi(x)$ is refined to

$$(\exists\, x \in [a, \tfrac{a+b}{2}] \,.\, \phi(x)) \vee (\exists\, x \in [\tfrac{a+b}{2}, b] \,.\, \phi(x))$$

Refinement

$$\phi^- \implies \phi \implies \phi^+$$

- If $\phi^- = \bot$ and $\phi^+ = \top$ we cannot say much about ϕ.
- To make progress, we *refine* ϕ to an equivalent formula in which quantifers range over smaller intervals.
- A simple strategy is to split quantified intervals in halves:
  - $\forall\, x \in [a, b] \,.\, \phi(x)$ is refined to

    $$(\forall\, x \in [a, \tfrac{a+b}{2}] \,.\, \phi(x)) \wedge (\forall\, x \in [\tfrac{a+b}{2}, b] \,.\, \phi(x))$$

  - $\exists\, x \in [a, b] \,.\, \phi(x)$ is refined to

    $$(\exists\, x \in [a, \tfrac{a+b}{2}] \,.\, \phi(x)) \vee (\exists\, x \in [\tfrac{a+b}{2}, b] \,.\, \phi(x))$$

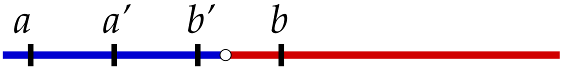
- This amounts to searching with *bisection*.

Refinement of cuts

- To refine a cut

$$\text{cut } x : [a, b] \text{ left } \phi(x) \text{ right } \psi(x)$$

we try to move $a \mapsto a'$ and $b \mapsto b'$.

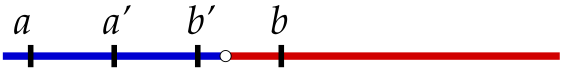


Refinement of cuts

- To refine a cut

$$\text{cut } x : [a, b] \text{ left } \phi(x) \text{ right } \psi(x)$$

we try to move $a \mapsto a'$ and $b \mapsto b'$.



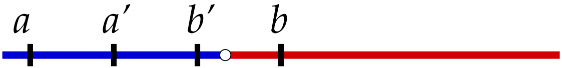
- If $\phi^{-}(a') = \top$ then move $a \mapsto a'$.

Refinement of cuts

- To refine a cut

$$\text{cut } x : [a, b] \text{ left } \phi(x) \text{ right } \psi(x)$$

we try to move $a \mapsto a'$ and $b \mapsto b'$.



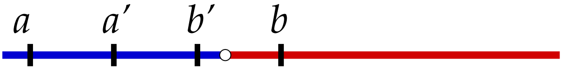
- If $\phi^-(a') = \top$ then move $a \mapsto a'$.
- If $\psi^-(b') = \top$ then move $b \mapsto b'$.

Refinement of cuts

- To refine a cut

$$\text{cut } x : [a, b] \text{ left } \phi(x) \text{ right } \psi(x)$$

we try to move $a \mapsto a'$ and $b \mapsto b'$.



- If $\phi^-(a') = \top$ then move $a \mapsto a'$.
- If $\psi^-(b') = \top$ then move $b \mapsto b'$.
- One or the other endpoint moves eventually because cuts are located.

Evaluation

- To evaluate a sentence ϕ:

Evaluation

- To evaluate a sentence ϕ:
  - if $\phi^- = \top$ then output $\top$,

Evaluation

- To evaluate a sentence ϕ:
  - if $\phi^- = \top$ then output $\top$,
  - if $\phi^+ = \bot$ then output $\bot$,

Evaluation

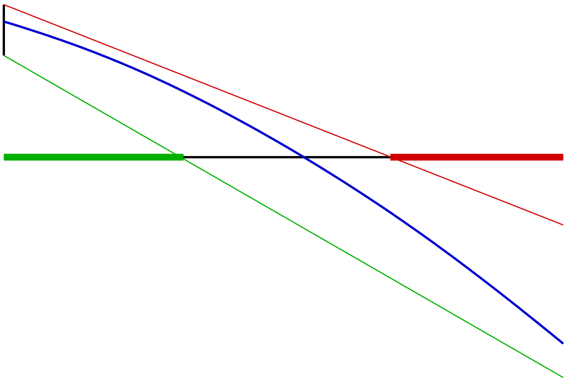
- To evaluate a sentence ϕ:
  - if $\phi^- = \top$ then output $\top$,
  - if $\phi^+ = \bot$ then output $\bot$,
  - otherwise refine ϕ and repeat.

Evaluation

- To evaluate a sentence ϕ:
  - if $\phi^- = \top$ then output $\top$,
  - if $\phi^+ = \bot$ then output $\bot$,
  - otherwise refine ϕ and repeat.
- Evaluation may not terminate, but this is expected, as ϕ is only *semi*decidable.

Speeding up the computation

Estimate an inequality $f(x) < 0$ on $[a, b]$ by approximating f with a linear map from above and below.



This is essentially Newton's interval method.

Questions

- How do we incorporate $\mathbb{N}$ and recursion?

Questions

- How do we incorporate $\mathbb{N}$ and recursion?
- How to extend Newton's method to improper intervals?

Questions

- How do we incorporate $\mathbb{N}$ and recursion?
- How to extend Newton's method to improper intervals?
- How to extend Newton's method to the multivariate case?

Questions

- How do we incorporate $\mathbb{N}$ and recursion?
- How to extend Newton's method to improper intervals?
- How to extend Newton's method to the multivariate case?
- Can we do higher-type computations $\int$ and $\frac{d}{dx}$?

Questions

- How do we incorporate $\mathbb{N}$ and recursion?
- How to extend Newton's method to improper intervals?
- How to extend Newton's method to the multivariate case?
- Can we do higher-type computations $\int$ and $\frac{d}{dx}$?
- Can this lead to a useful domain-specific language?