Logic for real number computation

Helmut Schwichtenberg
(j.w.w. Ulrich Berger, Kenji Miyamoto and Hideki Tsuiki)

Mathematisches Institut, LMU, München

PCC, Oslo, 23. May 2015
Classical and constructive proofs

- View classical logic as a fragment of minimal logic.
- Use both $\exists_x A$ and its “weak” variant $\tilde{\exists}_x A := \neg \forall_x \neg A$.

Brouwer-Heyting-Kolmogorov interpretation

- Proofs can have (hidden) computational content.
- Extract such computational content (‘‘realizability’’).

Tools

- Computationally relevant and irrelevant logical connectives.
- Inductively/coinductively defined predicates.
Representation of real numbers \( x \in [-1, 1] \)

Binary code.

\[
\frac{a_0}{2^1} + \frac{a_1}{2^2} + \frac{a_2}{2^3} + \cdots = \sum_{i=0}^{\infty} \frac{a_i}{2^{i+1}} \quad \text{with } a_i \in \{-1, 1\}.
\]

View \( a_0 a_1 a_2 \ldots \) as “stream”. Dyadic rationals:

\[
\frac{a_0}{2^1} + \frac{a_1}{2^2} + \cdots + \frac{a_n}{2^{n+1}} = \sum_{i=0}^{n} \frac{a_i}{2^{i+1}}.
\]

For example

\[
\frac{3}{8} = \frac{1}{2} - \frac{1}{4} + \frac{1}{8}, \quad \frac{5}{8} = \frac{1}{2} + \frac{1}{4} - \frac{1}{8}.
\]

**Uniqueness:** +, **Continuity:** −, **Productivity:** −.
Signed digit code. Cure for P: add 0 as a digit.

\[
\begin{align*}
3 &= 0 + 1 + 1 \\
\frac{8}{8} &= \frac{2}{2} + \frac{4}{4} + \frac{8}{8} \\
1 &= 0 + 1 \\
\frac{1}{2} &= \frac{1}{2} + \frac{1}{4} - \frac{1}{8} \\
1 &= 1 + 1 \\
\frac{1}{2} &= \frac{1}{2} - \frac{1}{4} + \frac{1}{8}
\end{align*}
\]

Widely used for real number computation.
Uniqueness: $\text{—}$, Continuity: $\text{—}$, Productivity: $\text{+}$. 
Gray code (or binary reflected code). Cure for C: flip after 1. Instead of

\[ C_{-1}(x) := \frac{x - 1}{2} \]
\[ C_1(x) := \frac{x + 1}{2} \]

we now have

\[ LR_{-1}(x) := C_{-1}(x) = \frac{x - 1}{2} \]
\[ LR_1(x) := C_1(-x) = \frac{-x + 1}{2} = -\frac{x - 1}{2} \]
Hence \( \frac{3}{8} \sim RRR, \quad \frac{5}{8} \sim RLR. \)

Extended Gray code. Cure for P: delay (via U, D, Fin_{L/R}).

\[ U \circ \text{Fin}_L = L \circ R, \quad D \circ \text{Fin}_L = \text{Fin}_L \circ L, \]
\[ U \circ \text{Fin}_R = R \circ R, \quad D \circ \text{Fin}_R = \text{Fin}_R \circ L. \]

\[ R \circ U \circ \text{Fin}_L = R \circ L \circ R, \quad U \circ \text{Fin}_R \circ R \circ L = R \circ R \circ R \circ L, \]
\[ R \circ U \circ D \circ \text{Fin}_R = R \circ U \circ \text{Fin}_R \circ L = R \circ R \circ R \circ L. \]

\textbf{Uniqueness: +, Continuity: +, Productivity: +.}
Proofs on predicates with streams as content

Keep reals $\times$ abstract (via $\forall^\text{nc}_x \ldots$), and let the stream be the computational content of an atomic proposition.

How to define predicates with a stream as content? Coinductive predicates.
Define inductively a set $I$ of (abstract) reals, by the single clause

$$\forall_{x}^{nc} \forall_{d \in SD}(Ix \rightarrow I \frac{x + d}{2}).$$

Corresponding operator (or predicate transformer):

$$\Phi(X) := \{ y | \exists_{x \in X}^{r} \exists_{d \in SD}(y = \frac{x + d}{2}) \},$$

$$I := \mu_{X} \Phi(X) \quad \text{least fixed point},$$

$$col := \nu_{X} \Phi(X) \quad \text{greatest fixed point}.$$

Associated to $I$ is a (free) algebra $I$ with a single unary constructor

$$C : SD \rightarrow I \rightarrow I.$$ 

A realizer of an atomic proposition $colr$ is a stream of signed digits.
Induction and coinduction axioms:

$$\Phi(X) = X \rightarrow I \subseteq X \subseteq \text{co}I.$$ 

"Strengthened" form:

$$\Phi(I \cap X) \subseteq X \rightarrow I \subseteq X,$$
$$X \subseteq \Phi(\text{co}I \cup X) \rightarrow X \subseteq \text{co}I.$$
Similarly: simultaneous inductive definition of sets $G$ and $H$ of (abstract) reals. The clauses for $G$ are

$$\forall^\text{nc}_x \forall a \in \text{PSD}(G x \rightarrow G(-a\frac{x - 1}{2})), \quad \forall^\text{nc}_x (H x \rightarrow G(\frac{x}{2}))$$

and for $H$

$$\forall^\text{nc}_x \forall a \in \text{PSD}(G x \rightarrow H(a\frac{x + 1}{2})), \quad \forall^\text{nc}_x (H x \rightarrow H(\frac{x}{2})).$$

Associated to $G, H$ are algebras $G, H$ with constructors

- $\text{LR}: \text{PSD} \rightarrow G \rightarrow G$,
- $\text{U}: H \rightarrow G$ (for “undefined”),
- $\text{Fin}: \text{PSD} \rightarrow G \rightarrow H$,
- $\text{D}: H \rightarrow H$ (for “delay”).
The corresponding operators $\Gamma$, $\Delta$ are defined by

\[
y \in \Gamma(X, Y) \iff \exists_{x \in X} \exists_{a} (y = -a \frac{x - 1}{2}) \lor \exists_{x \in Y} (y = \frac{x}{2}),
\]
\[
y \in \Delta(X, Y) \iff \exists_{x \in X} \exists_{a} (y = a \frac{x + 1}{2}) \lor \exists_{x \in Y} (y = \frac{x}{2}).
\]

and $(\co G, \co H) := \nu_{X, Y}(\Gamma(X, Y), \Delta(X, Y))$.

Simultaneous coinduction axiom ($\subseteq$ meant component-wise)

\[
(X, Y) \subseteq (\Gamma(\co G \cup X, \co H \cup Y), \Delta(\co G \cup X, \co H \cup Y)) \rightarrow (X, Y) \subseteq (\co G, \co H).
\]

Witnesses for $\co Gr$, $\co Hr$ are “cototal ideals” in the algebras $G$, $H$. 
From e-Gray code to SD code

Axioms on the abstract reals:

\[ \forall x (1x = x), \quad \forall x ((-1)x = -x), \quad \forall a, x, d \left( \frac{ax + d}{2} = \frac{ax + ad}{2} \right). \]

Want \( \text{co}G \subseteq \text{co}I \). Proof requires generalization to

\[ \forall x^{nc} \left( \exists a (\text{co}G(ax) \lor \text{co}H(ax)) \rightarrow \text{co}I x \right). \]

Proof. For \( X := \{ x | \exists a (\text{co}G(ax) \lor \text{co}H(ax)) \} \) show \( X \subseteq \text{co}I \). Use coinduction, i.e., prove \( X \subseteq \Phi(\text{co}I \cup X) \). By cases.
\[ [\text{apq}] (\text{CoRec psd}@[@(ag \ ysum \ ah) => iv)apq} \\
( [\text{apq0}] \text{[case (right apq0)} \\
\text{(InL p -> [case (Des p)} \\
\text{(InL ap ->} \\
\text{PsdToSd(left apq0 times left ap)@} \\
\text{InR(inv(left apq0 times left ap)@InL right ap))} \\
\text{(InR q -> Mid@InR(left apq0@InR q)])} \\
\text{(InR q -> [case (Des q)} \\
\text{(InL ap ->} \\
\text{PsdToSd(left apq0 times left ap)@} \\
\text{InR(left apq0 times left ap@InL right ap))} \\
\text{(InR q0 -> Mid@InR(left apq0@InR q0)])}] \]

@@ / @ denotes product types / terms. iv, ag, ah denote I, G, H.

\[ p: G \quad \text{ap: PSD} \times G, \]
\[ q: H \quad \text{apq: PSD} \times (G + H). \]

\[ [\text{apq}] := \lambda_{\text{apq}}. \]
Type of the corecursion operator:

\[ \text{co}R^\tau_I : \tau \to (\tau \to \text{SD} \times (I + \tau)) \to I \quad \text{with} \quad \tau := \text{PSD} \times (G + H). \]

\( \text{SD} \times (I + \tau) \) appears since \( I \) has the constructor \( C : \text{SD} \to I \to I \).

The meaning of \( \text{co}R^\tau_I \) is defined by the conversion rule

\[
\text{co}R^\tau_I \text{NM} \mapsto C_{(MN)_1}(\text{id}^{I \to I}, \lambda_y(\text{co}R^\tau_I yM))(MN)_2).
\]

\( x_1, x_2 \) projections. For \( f : \rho \to \alpha \) and \( g : \sigma \to \alpha \) let \( [f, g] \) denote

\[
[f, g](z) := \begin{cases} 
  f(x) & \text{if } z = \text{inl}(x), \\
  g(y) & \text{if } z = \text{inr}(y).
\end{cases}
\]
By analyzing the particular step function $M$ (extracted from our proof) we can write $\lambda_y^{\text{co}R^*_y}yM$ as a function

$$[f, g]: \text{PSD} \times G + \text{PSD} \times H \rightarrow I$$

defined by

$$f(a, \text{LR}_b(p)) = C_{ab}(f(-ab, p)), \quad g(a, \text{Fin}_b(p)) = C_{ab}(f(ab, p)),
$$

$$f(a, \text{U}(q)) = C_0(g(a, q)), \quad g(a, \text{D}(q)) = C_0(g(a, q)).$$

Originally we wrote a Haskell program to convert e-Gray code to signed digit code, which consisted of 12 rules (cases for $a$ and $b$). Its correctness was not proved, and we did not realize that it can be compressed into 4 rules until we had the extracted term.
Converse: $\text{col} \subseteq \text{coG}$. Proof requires generalization to CoIToCoG:

\[
\forall_{\text{x}} (\exists_{\text{a}} \text{coI(ax)} \rightarrow \text{coGx}), \quad \forall_{\text{x}} (\exists_{\text{a}} \text{coI(ax)} \rightarrow \text{coHx}).
\]

Proof. For $X := \{ x | \exists_{a}(ax \in \text{col}) \}$ show $X \subseteq \text{coG}$ simultaneously with $X \subseteq \text{coH}$. By simultaneous coinduction it suffices to prove

\[
X \subseteq \Gamma(\text{coG} \cup X, \text{coH} \cup X),
\]

\[
X \subseteq \Delta(\text{coG} \cup X, \text{coH} \cup X).
\]

Use cases on the definition of $\Gamma$, $\Delta$. 

---

From SD code to e-Gray code
[bv](CoRec psd@iv=>ag psd@iv=>ah)bv

[[bv0][case (left Des right bv0)
  (Lft -> InL(inv left bv0@InR(PRht@right Des right bv0)))
  (Rht -> InL(left bv0@InR(PLft@right Des right bv0)))
  (Mid -> InR(InR(left bv0@right Des right bv0)))]]

[[bv0][case (left Des right bv0)
  (Lft -> InL(inv left bv0@InR(PLft@right Des right bv0)))
  (Rht -> InL(left bv0@InR(PRht@right Des right bv0)))
  (Mid -> InR(InR(left bv0@right Des right bv0)))]]

with bv: \textbf{PSD} \times I.
By analyzing the step function $M, M'$ (extracted from the proof) we can write the functions $\lambda_y^c R_G y^\alpha M M'$ and $\lambda_y^c R_H y^\alpha M M'$ as $g: \text{PSD} \times I \to G$ and $h: \text{PSD} \times I \to H$ defined by

\begin{align*}
g(b, C_{-1}(v)) &= LR_{-b}(g(1, v)), & h(b, C_{-1}(v)) &= \text{Fin}_{-b}(g(-1, v)), \\
g(b, C_1(v)) &= LR_{b}(g(-1, v)), & h(b, C_1(v)) &= \text{Fin}_{b}(g(1, v)), \\
g(b, C_0(v)) &= U(h(b, v)), & h(b, C_0(v)) &= D(h(b, v)).
\end{align*}
Postprocessing e-Gray code

- Given: $n$ and an e-Gray code of some $x \in [-1, 1]$.
- Want: Another e-Gray code of the same $x$, but without $\text{UD}^k \text{Fin}_{L/R}$ in its prefix of length $n$.

Have extracted an algorithm from a corresponding proof (by induction on $n$).
Conclusion

▶ “Code carrying proof” can be a reasonable alternative to “Proof carrying code” (Necula).


Open problems

▶ Direct construction of the average for e-Gray code.
▶ Extend linear two-sorted arithmetic (p-time) with coinduction.