Proofs and computation with infinite data

Helmut Schwichtenberg
(j.w.w. Nils Köpp)

Mathematisches Institut, LMU, München

Logic Colloquium 2021, Poznan, 21. July 2021

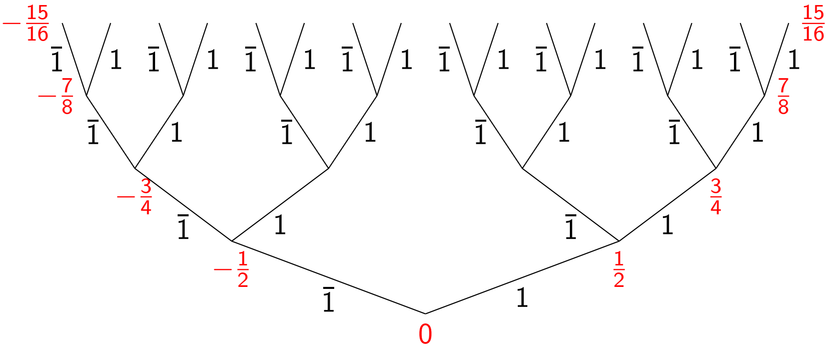
- Proofs may have computational content, which can be extracted (via realizability).
- Proofs (but not programs) can be checked for correctness.

Issues:

- Algorithms for exact real numbers extracted from proofs.
- Bounds for look-ahead formally verified.

For simplicity $x \in [-1, 1]$. Dyadic rationals:

$$\sum_{i<k} \frac{a_i}{2^{i+1}} \qquad \text{with } a_i \in \{-1, 1\}$$



with $\bar{1} := -1$.

Problem with productivity:

$$\bar{1}111 + 1\bar{1}\bar{1}\bar{1}\cdots = ?$$

What is the first digit? Cure: delay: add 0. Signed digit code

$$\sum_{i<k} \frac{d_i}{2^{i+1}} \qquad \text{with } d_i \in \{-1, 0, 1\}.$$

Widely used for real number computation. There is a lot of redundancy: $\bar{1}1$ and $0\bar{1}$ both denote $-\frac{1}{4}$.

Algorithms on stream-represented real numbers

We define an inductive predicate I by the single clause

$$\forall_{d,x',x}(d \in \mathrm{Sd} \to x' \in I \to x = \frac{x'+d}{2} \to x \in I).$$

The dual ${}^{\mathrm{co}}I$ of I is defined by its closure axiom ${}^{\mathrm{co}}I^{-}$:

$$\forall_x \Big(x \in {}^{\mathrm{co}}I \to \exists_{d,x'} \Big(d \in \mathrm{Sd} \wedge x' \in {}^{\mathrm{co}}I \wedge x = \frac{x'+d}{2} \Big) \Big)$$

and the coinduction (or greatest-fixed-point) axiom ${}^{\mathrm{co}}I^{+}$:

$$\forall_x \Big(x \in X \to \exists_{d,x'} \Big(d \in \mathrm{Sd} \wedge x' \in {}^{\mathrm{co}}I \cap X \wedge x = \frac{x'+d}{2} \Big) \Big) \to \\ \forall_x (x \in X \to x \in {}^{\mathrm{co}}I).$$

Goal: compute the average of two stream-coded reals. Prove

$$x, y \in {}^{\mathrm{co}}I \to \frac{x+y}{2} \in {}^{\mathrm{co}}I.$$

Computational content of this proof will be the desired algorithm.

Informal proof[1]. Define sets P, Q of averages, Q with a "carry" $i \in \mathbb{Z}$:

$$P := \{ \frac{x+y}{2} \mid x, y \in {}^{\mathrm{co}}I \},$$
$$Q := \{ \frac{x+y+i}{4} \mid x, y \in {}^{\mathrm{co}}I, i \in \mathrm{Sd}_2 \} \quad (\mathrm{Sd}_2 := \{-2, -1, 0, 1, 2\}).$$

Suffices: Q satisfies the clause coinductively defining ${}^{\mathrm{co}}I$. Then by the greatest-fixed-point axiom for ${}^{\mathrm{co}}I$ we have $Q \subseteq {}^{\mathrm{co}}I$. Since also $P \subseteq Q$ we obtain $P \subseteq {}^{\mathrm{co}}I$, which is our claim.

[1] U. Berger & M. Seisenberger, Proofs, programs, processes, 2012

$P \subseteq Q$:

$$\begin{aligned} & x, y \in {}^{\mathrm{co}}I \to \\ & \exists_{i,x',y'} \Big(i \in \mathrm{Sd}_2 \wedge x', y' \in {}^{\mathrm{co}}I \wedge \frac{x+y}{2} = \frac{x'+y'+i}{4} \Big) \end{aligned} \tag{1}$$

Proof.
From $x = \frac{x'+d}{2}$, $y = \frac{y'+e}{2}$ get $\frac{x+y}{2} = \frac{x'+y'+d+e}{4}$. □

Computational content:

$$\begin{aligned} & f_1 : \mathbb{S} \to \mathbb{S} \to \mathbb{D}_2 \times \mathbb{S} \times \mathbb{S} \\ & f_1(\mathrm{C}_d(u), \mathrm{C}_e(v)) = \langle d + e, u, v \rangle \end{aligned}$$

Q satisfies the clause coinductively defining ${}^{\mathrm{co}}I$:

$$i \in \mathrm{Sd}_2 \land x, y \in {}^{\mathrm{co}}I \to \exists_{d,j,x',y'} \Big(d \in \mathrm{Sd} \land j \in \mathrm{Sd}_2 \land x', y' \in {}^{\mathrm{co}}I \land \frac{x+y+i}{4} = \frac{\frac{x'+y'+j}{4} + d}{2} \Big). \tag{2}$$

Proof.

From $x = \frac{x'+d}{2}$, $y = \frac{y'+e}{2}$ get $\frac{x+y+i}{4} = \frac{x'+y'+k}{8}$ for $k := d+e+2i$.
Write $k = J(k) + 4D(k)$ with $|D(k)| \le 1$, $|J(k)| \le 2$ for $|k| \le 6$.

$$\frac{x+y+i}{4} = \frac{x'+y'+j+4d'}{8} = \frac{\frac{x'+y'+j}{4} + d'}{2}.$$

□

Computational content:

$$f_2 \colon \mathbb{D}_2 \times \mathbb{S} \times \mathbb{S} \to \mathbb{D} \times \mathbb{D}_2 \times \mathbb{S} \times \mathbb{S}$$
$$f_2\langle i, \mathrm{C}_d(u), \mathrm{C}_e(v)\rangle = \langle D(k), J(k), u, v\rangle \quad \text{with } k := d + e + 2i.$$

$P \subseteq {}^{\mathrm{co}}I$: The average of two real numbers x, y in ${}^{\mathrm{co}}I$ is in ${}^{\mathrm{co}}I$.

$$x, y \in {}^{\mathrm{co}}I \to \frac{x+y}{2} \in {}^{\mathrm{co}}I \tag{3}$$

Proof.

By coinduction from (1) and (2). □

Computational content: Uses corecursion.

- From $u, v \in \mathbb{S}$ form initial triple $f_1(u, v) \in \mathbb{D}_2 \times \mathbb{S} \times \mathbb{S}$.
- Iterate $f_2 \colon \mathbb{D}_2 \times \mathbb{S} \times \mathbb{S} \to \mathbb{D} \times \mathbb{D}_2 \times \mathbb{S} \times \mathbb{S}$ starting with $f_1(u, v)$.
- Return stream of generated $d \in \mathbb{D}$.

Bounds for the look-ahead

We replace the unary coinductive predicate ${}^{\mathrm{co}}I$ on reals by a binary inductive predicate I with the property that

a realizer of Ixn is a list of length n of signed digits approximating x with error bound $\frac{1}{2^n}$.

Below we will prove

$$n \in T_{\mathbb{N}} \to Ix(n+1) \to Iy(n+1) \to I\Big(\frac{x+y}{2}\Big)n,$$
$$n \in T_{\mathbb{N}} \to Ix(3n+3) \to Iy(3n+3) \to I(xy)n.$$

We inductively define a predicate I by the clauses

$$I_0^+ : \forall_x (x \in \mathbb{R} \to |x| \leq 1 \to Ix0),$$
$$I_1^+ : \forall_{d,x',x,n} \Big(d \in \mathrm{Sd} \to Ix'n \to x = \frac{x'+d}{2} \to Ix(n+1) \Big).$$

The elimination (induction, least-fixed-point) axiom is I^-:

$$\forall_x (x \in \mathbb{R} \to |x| \leq 1 \to Xx0) \to$$
$$\forall_{d,x',x,n} \Big(d \in \mathrm{Sd} \to Ix'n \to Xx'n \to x = \frac{x'+d}{2} \to Xx(n+1) \Big) \to$$
$$\forall_{x,n} (Ixn \to Xxn).$$

This axiom expresses that every “competitor” X satisfying the same clauses contains I. We take all substitution instances (w.r.t. the predicate variable X) of I_i^+, I^- as axioms.

Properties of I

Lemma (ICompat)

$\forall_{x,n}(x = y \to Ixn \to Iyn)$.

Proof.

Use I^- and properties of real equality. □

Lemma (IClosure)

$$\forall_{x,n}\Big(Ix(n+1) \to \exists_{d,x}\Big(d \in \mathrm{Sd} \wedge Ix'n \wedge x = \frac{x'+d}{2}\Big)\Big).$$

Proof.

Assume Ixm and $m = n+1$. Using I^- leaves us with two goals. The first one has a premise $0 = n+1$; we can use ex-falso. The second one has an existential conclusion which easily follows from what we have. □

Properties of I (continued)

Lemma (IUMinus)

$\forall_{x,n}(Ixn \to I(-x)n)$.

Proof.

Assume Ixn. Using I^- leaves us with two goals. The first one follows from I_0^+, and the second one from I_1^+. □

Lemma (ISdTimes)

$\forall_{d,x,n}(d \in \mathrm{Sd} \to Ixn \to I(dx)n)$.

Proof.

Cases on $d \in \mathrm{Sd}$, together with a Lemma IZero: $\forall_n I0n$, and IUMinus. In each case ICompat is applied. □

$$
\begin{aligned}
&Ix(n+1) \to Iy(n+1) \to \\
&\exists_{i,x',y'}\Big(i \in \mathrm{Sd}_2 \wedge Ix'n \wedge Iy'n \wedge \frac{x+y}{2} = \frac{x'+y'+i}{4}\Big)
\end{aligned} \tag{4}
$$

Proof.

From $x = \frac{x'+d}{2}$, $y = \frac{y'+e}{2}$ get $\frac{x+y}{2} = \frac{x'+y'+d+e}{4}$. $\square$

Computational content:

$$
\begin{aligned}
&f_4 \colon \mathbb{L} \to \mathbb{L} \to \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L} \\
&f_4(\mathrm{C}_d(u), \mathrm{C}_e(v)) = \langle d+e, u, v\rangle
\end{aligned}
$$

$$i \in \mathrm{Sd}_2 \wedge Ix(n+1) \wedge Iy(n+1) \to \exists_{d,j,x',y'} \Big(d \in \mathrm{Sd} \wedge j \in \mathrm{Sd}_2 \wedge Ix'n \wedge Iy'n \wedge \frac{x+y+i}{4} = \frac{\frac{x'+y'+j}{4}+d}{2} \Big). \tag{5}$$

Proof.

From $x = \frac{x'+d}{2}$, $y = \frac{y'+e}{2}$ get $\frac{x+y+i}{4} = \frac{x'+y'+k}{8}$ for $k := d+e+2i$.
Write $k = J(k) + 4D(k)$ with $|D(k)| \leq 1$, $|J(k)| \leq 2$ for $|k| \leq 6$.

$$\frac{x+y+i}{4} = \frac{x'+y'+j+4d'}{8} = \frac{\frac{x'+y'+j}{4}+d'}{2}. \qquad \square$$

Computational content:

$$f_5 \colon \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L} \to \mathbb{D} \times \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L}$$
$$f_5\langle i, \mathrm{C}_d(u), \mathrm{C}_e(v)\rangle = \langle D(k), J(k), u, v\rangle \quad \text{with } k := d+e+2i.$$

$$n \in T_{\mathbb{N}} \to Ix(n+1) \to Iy(n+1) \to I\left(\frac{x+y}{2}\right)n \tag{6}$$

Proof.

By induction from (4) and (5). □

Computational content: Uses recursion. Given n (wlog $0 < n$).

- From $u, v \in \mathbb{L}$ form initial triple $f_4(u, v) \in \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L}$.
- Iterate n times $f_5 \colon \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L} \to \mathbb{D} \times \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L}$, starting with $f_4(u, v)$.
- Return list of generated $d \in \mathbb{D}$.

$$n \in T_{\mathbb{N}} \to Ix(n+3) \to Iy(n+3) \to \exists_{i,x',y',z}\Big(\\ Iy'(n+2) \wedge i \in \mathrm{Sd}_2 \wedge Ix'(n+2) \wedge Izn \wedge xy = \frac{x'y'+z+i}{4}\Big). \tag{7}$$

Proof.

Assume $Ix(n+3)$ and $Iy(n+3)$. By IClosure: $x = \frac{x'+d}{2}$ and $y = \frac{y'+e}{2}$ with $Ix'(n+2)$ and $Iy'(n+2)$ and $d, e \in \mathrm{Sd}$. Using ISdTimes and (6) we obtain $I\big(\frac{ex'+dy'}{2}\big)(n+1)$. By IClosure: z, d_0 such that Izn, $d_0 \in \mathrm{Sd}$ and

$$\frac{ex'+dy'}{2} = \frac{z+d_0}{2}, \quad \text{hence}$$

$$\frac{(x'+d)(y'+e)}{4} = \frac{x'y'+(ex'+dy')+de}{4} = \frac{x'y'+z+(d_0+de)}{4},$$

which is of the required form. □

$$
\begin{aligned}
&Iy(n+2) \to i \in \mathrm{Sd}_2 \to Ix(m+1) \to Iz(n+3) \to \exists_{d,j,x',z'} \Big(\\
&d{\in}\mathrm{Sd} \wedge j{\in}\mathrm{Sd}_2 \wedge Ix'm \wedge Iz'n \wedge \frac{xy+z+i}{4} = \frac{\frac{x'y+z'+j}{4}+d}{2} \Big)
\end{aligned} \tag{8}
$$

Proof Let $Iy(n+2)$, $i \in \mathrm{Sd}_2$, $Ix(m+1)$ and $Iz(n+3)$. By IClosure

$$x = \frac{x_1+d_1}{2} \quad z = \frac{z_0+d_0}{2} \quad \text{with } Ix_1m,\ Iz_0(n+2) \text{ and } d_1, d_0 \in \mathrm{Sd}.$$

Then

$$
\begin{aligned}
\frac{xy+z+i}{4} &= \frac{(x_1+d_1)y+(z_0+d_0)+2i}{8} \\
&= \frac{x_1y+(z_0+d_1y+i)+d_0+i}{8}.
\end{aligned}
$$

Proof of (8) (continued)

Have $I(d_1 y)(n+2)$ by ISdTimes and $Iv(n+2)$ for $v := \frac{z_0+d_1 y+i}{4}$ by (4), (5). Using v we can continue the chain of equations by

$$= \frac{x_1 y + 4v + d_0 + i}{8}.$$

Because of $Iv(n+2)$ by IClosure we can write

$$v = \frac{z_1 + e_0}{2} = \frac{\frac{z_2+e}{2} + e_0}{2} \qquad \text{with } Iz_1(n+1),\ Iz_2 n \text{ and } e_0, e \in \mathrm{Sd}.$$

Therefore

$$= \frac{x_1 y + (z_2 + e + 2e_0) + d_0 + i}{8}.$$

Let $k := e + 2e_0 + d_0 + i$. Write $k = J(k) + 4D(k)$ Hence

$$= \frac{x_1 y + z_2 + j + 4d}{8} = \frac{\frac{x_1 y+z_2+j}{4} + d}{2} \quad \text{with } j := J(k),\ d := D(k).$$

$$n \in T_{\mathbb{N}} \to i \in \mathrm{Sd}_2 \to Iy(3n-1) \to Ix(3n) \to Iz(3n) \to I\Big(\frac{xy+z+i}{4}\Big)n \tag{9}$$

Proof Induction on n. We only consider the step case. Assume $i \in \mathrm{Sd}_2$, $Iy(3n+2)$, $Ix(3n+3)$, $Iz(3n+3)$. Get $I\big(\frac{xy+z+i}{4}\big)(n+1)$ by I_1^+. Need $d \in \mathrm{Sd}$ and x' with $Ix'n$ such that

$$\frac{xy+z+i}{4} = \frac{x'+d}{2}.$$

From (8) we obtain $d \in \mathrm{Sd}$, $j \in \mathrm{Sd}_2$, x'' and z'' such that $Ix''(3n+2)$, $Iz''(3n)$ and

$$\frac{xy+z+i}{4} = \frac{\frac{x''y+z''+j}{4}+d}{2}.$$

It suffices to show $I\big(\frac{x''y+z''+j}{4}\big)n$. To this end we use the IH. This requires $Iy(3n-1)$, $Ix''(3n)$ and $Iz''(3n)$. The latter we have, and the former two follow from $Iy(3n+2)$ and $Ix''(3n+2)$.

$$n \in T_{\mathbb{N}} \to Ix(3n+3) \to Iy(3n+3) \to I(xy)n \tag{10}$$

Proof.
Assume $Ix(3n+3)$ and $Iy(3n+3)$. Using (7) for x, y, $3n$ we obtain i, x', y', z s.t. $Iy'(3n+2)$, $i \in \mathrm{Sd}_2$, $Ix'(3n+2)$, $Iz(3n)$ and

$$xy = \frac{x'y' + z + i}{4}.$$

To prove $I\left(\frac{x'y'+z+i}{4}\right)n$ we apply (9). It suffices to prove $Iy'(3n-1)$ and $Ix'(3n)$, which follows from $Iy'(3n+2)$ and $Ix'(3n+2)$. $\square$

The term extracted from this proof is

```
[n,u,u0]
 [let utvw
   (cIMultToMultc(n+n+n)u u0)
   (cIMultcToI n
    (cIToIPred(n+n+n)
     (cISuccToI(n+n+n)(cISuccToI(Succ(n+n+n))clft utvw)))
    clft crht utvw
    (cISuccToI(n+n+n)
     (cISuccToI(Succ(n+n+n))clft crht crht utvw))
    crht crht crht utvw)]
```

Here `utvw` is a variable of type $\mathbb{L} \times \mathbb{D}_2 \times \mathbb{L} \times \mathbb{L}$.

Open questions

- Do the same with division[2].
- Can one obtain a bound for the look-ahead as a term read off from the proof?

[2] Wiesnet & S., LMCS 2021, gives an informal argument for a certain bound

Conclusion

- Constructive logic (and arithmetic) can and should be seen as an extension of the classical setup.
- Using the realizability interpretation of proofs one can extract computational content.
- Verification is automated: `add-sound` applied to a proof returns an internal proof of the soundness theorem.