LOGIC FOR EXACT REAL ARITHMETIC

HELMUT SCHWICHTENBERG

Abstract. The work reported in [7, 1] is extended in two directions. (1) Instead of viewing the real numbers as abstractly given objects with all the necessary properties assumed as axioms we now use concrete real numbers (Cauchy sequences with moduli) and provide formal proofs in the style of constructive analysis [3, 4]. Apart from being more complete, this resolves the delicate issue which equality (for reals) is to be used in the clauses of coinductively defined predicates. However, the choice of our model for the reals does not influence the extracted (Gray code based) algorithms, since quantifiers over real numbers are taken as non-computational. (2) Following [5], we extract a Gray code based algorithm for multiplication from a proof that the reals are closed under times.

Keywords: Gray code, real number computation, inductive and coinductive definitions, program extraction.

2010 Mathematics Subject Classification: 03D78, 03F60, 03B70, 03B35

1. Introduction

Real numbers in the exact (as opposed to floating-point) sense can be given in different formats, for instance

(i) as Cauchy sequences (of rationals, with Cauchy modulus), or else
(ii) as infinite sequences ("streams") of signed digits $\{-1, 0, 1\}$ or
(iii) $\{-1, 1, \bot\}$ containing at most one copy of $\bot$ (meaning undefinedness), so-called "Gray code" [6, 9].

We are interested in formally verified algorithms on real numbers given as streams. To this end we consider formal existence proofs M and apply a proof theoretic method ("realizability") to extract their computational content. We switch between different representations of reals by labelling universal quantifiers on reals x as "non-computational" and then relativising x to a predicate ${}^{\mathrm{co}}I$ coinductively defined in such a way that the computational content of $x \in {}^{\mathrm{co}}I$ is a stream representing x. The desired algorithm

This work was supported by the International Research Staff Exchange Scheme (IRSES) Nr. 612638 CORCON.

is obtained as the extracted term $\mathrm{et}(M)$ of the existence proof M, and the required verification is provided by a formal soundness proof of the realizability interpretation.

The work reported in [7, 1] is extended in two directions. (1) Instead of viewing the real numbers as abstractly given objects with all the necessary properties assumed as axioms we now use concrete real numbers (Cauchy sequences with moduli) and provide formal proofs in the style of constructive analysis [3, 4]. The equality used in the clauses of coinductively defined predicates then is the defined equality on concrete reals. However, the choice of our model for the reals does not influence the extracted (stream based) algorithms, since quantifiers over real numbers are taken as non-computational. (2) Following [5], we extract a stream based algorithm for multiplication from a proof that the reals are closed under times.

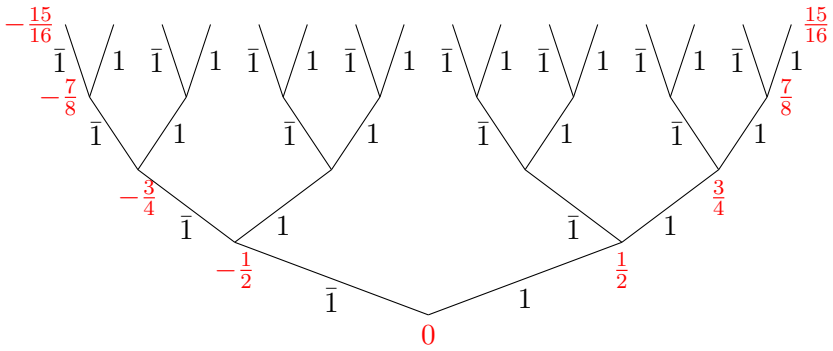
The rest of the paper is organized as follows. In Section 2 we recall signed digit and Gray code representations of real numbers. Sections 3 and 4 deal with the average and multiplication functions for signed digit streams, and Sections 5 and 6 do the same for pre-Gray code. In the final Sections 7 and 8 we give some details concerning formalization of the proofs in the Minlog system. We also present the realizers machine extracted from the formalized proofs and discuss the algorithms they represent.

2. Stream representations of real numbers

For simplicity we work in the interval $[-1, 1]$. Reals of the form

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

are called *dyadic rationals*:



where $\bar{1}$ means -1. Note that adjacent dyadics can differ in many digits:

$$\frac{7}{16} \sim 1\bar{1}11, \qquad \frac{9}{16} \sim 11\bar{1}\bar{1}.$$

A possible cure is to *flip* after an occurrence of 1; the result is called binary reflected (or Gray-) code.

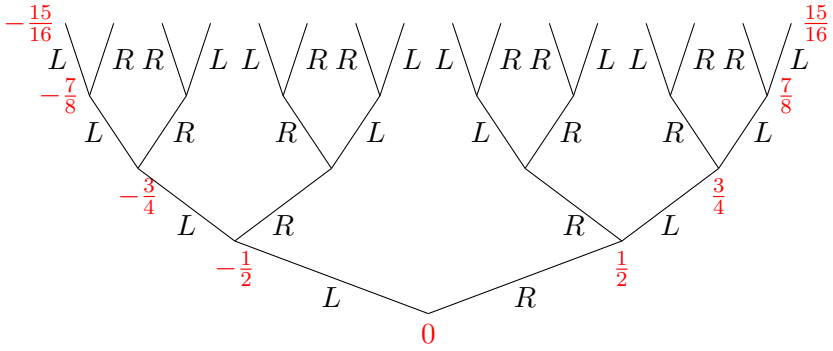


FIGURE 1. Binary reflected (or Gray-) code

Then we have

$$\frac{7}{16} \sim \mathrm{RRRL}, \qquad \frac{9}{16} \sim \mathrm{RLRL}.$$

However, a problem with "productivity" remains: we cannot determine what the first digit of $\bar{1}111\ldots + 1\bar{1}\bar{1}\bar{1}\ldots$ (or $\mathrm{LRLL}\ldots + \mathrm{RRRL}\ldots$) should be. The cure is to add *delay* digits. For dyadic rationals we add the digit 0 and obtain *signed digit code* widely used in numerical computation:

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

We have a lot of redundancy here: for instance $\bar{1}1$ and $0\bar{1}$ both denote $-\frac{1}{4}$.

For binary reflected code we first add digits U (for undefined), D (for delay), $\mathrm{Fin}_{L/R}$ (for finally left / right) and obtain *pre-Gray code*. Then a part of the last figure is expanded to Figure 2.

After computation in pre-Gray code, one can remove Fin_a by

$$\mathrm{U} \circ \mathrm{Fin}_a \mapsto a \circ \mathrm{R}, \qquad \mathrm{D} \circ \mathrm{Fin}_a \mapsto \mathrm{Fin}_a \circ \mathrm{L}.$$

If we now pass to infinite sequences, another source of non-uniqueness arises: $\mathrm{RRRLLL}\ldots$ and $\mathrm{RLRLLL}\ldots$ but also $\mathrm{RUDDDD}\ldots$ all denote $\frac{1}{2}$. From these three infinite sequences we can safely remove the former two and only keep $\mathrm{RUDDDD}\ldots$ to denote $\frac{1}{2}$. Then, generally,

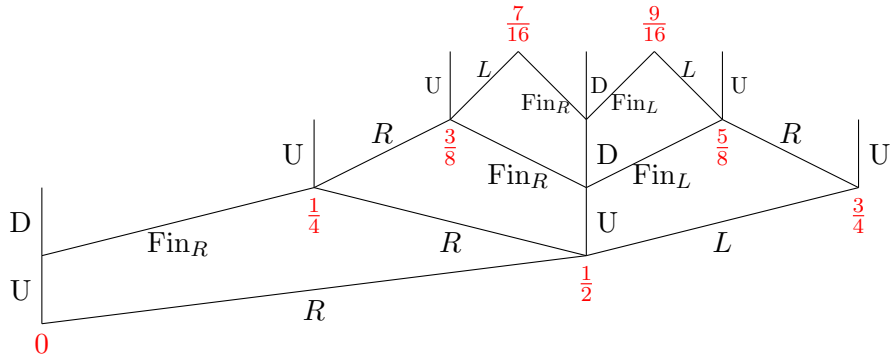


FIGURE 2. Pre-Gray code

(i) U occurs in a context UDDDD... only, and
(ii) such an occurrence of U appears exactly in the representation of dyadic rationals.

In this way we obtain a *unique* representation of real numbers by infinite sequences (or streams), which we call *pure Gray code*.

3. Average for signed digit streams

We now tackle our goal to extract stream algorithms from proofs, and as an example we consider a proof that the average of two real numbers in $[-1,1]$ is in $[-1,1]$ again. We first deal with the representation of reals as signed digit streams; in Section 5 we solve the corresponding problem for pre-Gray code. To start, we need to accomodate streams in our logical framework.

3.1. **The predicates I and ${}^{\mathrm{co}}I$.** We model infinite sequences of signed digits (or streams for short) as objects[1] in the (free) algebra $\mathbf{I}$ given by just one constructor $\mathrm{C}\colon \mathbf{Sd} \to \mathbf{I} \to \mathbf{I}$, where $\mathbf{Sd} := \{\mathrm{SdR}, \mathrm{SdM}, \mathrm{SdL}\}$ is a formal representation of signed digits. Each such object can be decomposed into its head (an object in $\mathbf{Sd}$) and tail (another stream). Intuitively, the stream $d_0, d_1, d_2 \ldots$ represents the real number

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

[1] More precisely, cototal ideals; cf. [8]

We inductively define a predicate I by the single clause

$$\forall^{\mathrm{nc}}_{d,x',x}(d \in \mathrm{Sd} \to x' \in I \to x = \frac{x'+d}{2} \to x \in I). \tag{1}$$

Here (and later) x ranges over real numbers and d over integers. Sd is a (formally inductive) predicate expressing that the integer d is a signed digit, i.e., $|d| \le 1$.

We have chosen (1) rather than the simpler

$$\forall^{\mathrm{nc}}_{d,x}(d \in \mathrm{Sd} \to x \in I \to \frac{x+d}{2} \in I), \tag{2}$$

since we want I to be compatible with the defined equality $=$ on real numbers

$$\forall^{\mathrm{nc}}_{x,y}(x = y \to x \in I \to y \in I), \tag{3}$$

which easily follows from (1) (with reflexivity, symmetry and transitivity of $=$). Using (3) we then obtain (2) from (1) as a lemma, called IClosure.

At this point the present paper deviates from [1]. In the latter work real numbers including their equality were viewed as given axiomatically. Desired well-known properties of reals were just taken as axioms, provided they have no computational content and hence do not influence the terms extracted from proofs. However, one also needs compatibilities like (3), which do have computational content (in fact, identities). — In the formal development the present paper is reporting on such axioms have been replaced by proofs. This required a full develoment of the number systems (unary and binary natural numbers, integers, rationals and reals). A real number x is taken as a pair $((a_n)_{n\in\mathbb{N}}, M)$ with $a_n \in \mathbb{Q}$ and $M\colon \mathbb{Z}^+ \to \mathbb{N}$ such that $(a_n)_n$ is a Cauchy sequence with modulus M, that is

$$|a_n - a_m| \le \frac{1}{2^p} \quad \text{for } n, m \ge M(p).$$

Two reals $x := ((a_n)_n, M)$, $y := ((b_n)_n, N)$ are equal (written $x = y$) if

$$|a_{M(p+1)} - b_{N(p+1)}| \le \frac{1}{2^p} \quad \text{for all } p \in \mathbb{Z}^+.$$

The "non-computational" (n.c.) universal quantifier $\forall^{\mathrm{nc}}_{d,x,y}$ deserves a special comment. It has the effect that the type of the computational content of this formula is independent of d, x, y and hence in particular of the concrete representations of integers and real numbers in the underlying theory. Computational content *only* arises from inductive (and coinductive, see below) predicates, here Sd and I. Therefore the type of I's single clause is $\mathbf{Sd} \to \mathbf{I} \to \mathbf{I}$, i.e., the type of $\mathbf{I}$'s constructor C.

Dually to I we coinductively define a predicate ${}^{\mathrm{co}}I$ whose clause is

$$(4) \qquad \forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}I \to \exists^{\mathrm{r}}_{d,x',y}(d \in \mathrm{Sd} \land x' \in {}^{\mathrm{co}}I \land y = \frac{x'+d}{2} \land x = y))$$

Here $\exists^{\mathrm{r}}_{d,x'}$ is an (inductively defined) version of the existential quantifier with the effect that again the computational content of the formula is independent of d, x'.

Similar to what was done for I above we can simplify the original (automatically obtained) CoIClause (4) to

Lemma 3.1 (CoIClosure)**.**

$$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}I \to \exists^{\mathrm{r}}_{d,x'}(d \in \mathrm{Sd} \land x' \in {}^{\mathrm{co}}I \land x = \frac{x'+d}{2})).$$

Since the type of $x \in {}^{\mathrm{co}}I$ is the same as the type of I, the type of ${}^{\mathrm{co}}I$'s clause is $\mathbf{I} \to \mathbf{Sd} \times \mathbf{I}$, i.e., the type of an operator *destructing* a stream into its head and its tail.

More formally, both I and ${}^{\mathrm{co}}I$ are defined as fixed points of an operator

$$\Phi(X) := \{\, x \mid \exists^{\mathrm{r}}_{d,x'}(d \in \mathrm{Sd} \land x' \in X \land x = \frac{x'+d}{2}) \,\}.$$

Then

$$\begin{aligned} I &:= \mu_X \Phi(X) && \text{least fixed point} \\ {}^{\mathrm{co}}I &:= \nu_X \Phi(X) && \text{greatest fixed point} \end{aligned}$$

satisfy the (strengthened) axioms

$$\begin{aligned} \Phi(I \cap X) \subseteq X \to I \subseteq X && \quad \text{induction} \\ X \subseteq \Phi({}^{\mathrm{co}}I \cup X) \to X \subseteq {}^{\mathrm{co}}I && \quad \text{coinduction} \end{aligned}$$

(they are called "strengthened" because their hypotheses are weaker than the fixed point property $\Phi(X) = X$).

3.2. **Realizability.** The realizability extensions $I^{\mathbf{r}}$ and $({}^{\mathrm{co}}I)^{\mathbf{r}}$ are binary predicates on streams v of signed digits (coming from $d \in \mathrm{Sd}$ in the definition of $\Phi(X)$) and real numbers x. Consider the operator

$$\Phi^{\mathbf{r}}(Y) := \{\, (v,x) \mid \exists^{\mathrm{nc}}_{v',d,x'}(d \in \mathrm{Sd} \land (v',x') \in Y \land x = \frac{x'+d}{2} \land v = \mathrm{C}_d(v')) \,\}$$

(the ${}^{\mathrm{nc}}$ in $\exists^{\mathrm{nc}}$ indicates that neither the quantified variables nor the kernel has computational significance). Since $\Phi^{\mathbf{r}}(Y)$ is strictly positive in Y, again

our underlying theory provides us with binary predicates (or relations) $I^{\mathbf{r}}$ and $({}^{\mathrm{co}}I)^{\mathbf{r}}$ for the least and greatest fixed point of $\Phi^{\mathbf{r}}$:

$$\begin{aligned} I^{\mathbf{r}} &:= \mu_Y \Phi^{\mathbf{r}}(Y) && \text{least fixed point} \\ ({}^{\mathrm{co}}I)^{\mathbf{r}} &:= \nu_Y \Phi^{\mathbf{r}}(Y) && \text{greatest fixed point} \end{aligned}$$

satisfying the (strengthened) axioms

$$\begin{aligned} \Phi^{\mathbf{r}}(I^{\mathbf{r}} \cap Y) \subseteq Y \to I^{\mathbf{r}} \subseteq Y && \text{induction} \\ Y \subseteq \Phi^{\mathbf{r}}(({}^{\mathrm{co}}I)^{\mathbf{r}} \cup Y) \to Y \subseteq ({}^{\mathrm{co}}I)^{\mathbf{r}} && \text{coinduction.} \end{aligned}$$

3.3. **Informal proof.** Consider the problem to compute the average of two real numbers coded by streams. To this end we will prove

$$\forall^{\mathrm{nc}}_{x,x'}(x, x' \in {}^{\mathrm{co}}I \to \frac{x+x'}{2} \in {}^{\mathrm{co}}I), \tag{5}$$

and the computational content of this proof will be the desired algorithm.

We give an informal proof, following [2]. Consider two sets of averages, the second one with a "carry" $i \in \mathbb{Z}$

$$P := \{\, \frac{x+y}{2} \mid x, y \in {}^{\mathrm{co}}I \,\}, \qquad Q := \{\, \frac{x+y+i}{4} \mid x, y \in {}^{\mathrm{co}}I, i \in \mathrm{Sd}_2 \,\},$$

where Sd_2 is a (formally inductive) predicate expressing that the integer i is an extended signed digit, i.e., $|i| \le 2$.

Recall that ${}^{\mathrm{co}}I$ is a fixed point of Φ. Hence ${}^{\mathrm{co}}I \subseteq \Phi({}^{\mathrm{co}}I)$, which is Lemma 3.1 (CoIClosure). It suffices to show that Q satisfies CoIClosure

$$\forall^{\mathrm{nc}}_{x}(x \in Q \to \exists^{\mathrm{r}}_{d,x'}(d \in \mathrm{Sd} \wedge x' \in Q \wedge x = \frac{x'+d}{2})),$$

for then by the greatest-fixed-point axiom for ${}^{\mathrm{co}}I$ we have $Q \subseteq {}^{\mathrm{co}}I$. Since we also have $P \subseteq Q$ we then obtain $P \subseteq {}^{\mathrm{co}}I$, which is our claim.

Lemma 3.2 (CoIAvToAvc)**.**

$$\forall^{\mathrm{nc}}_{x,y}(x, y \in {}^{\mathrm{co}}I \to \exists^{\mathrm{r}}_{i,x',y'}(i \in \mathrm{Sd}_2 \wedge x', y' \in {}^{\mathrm{co}}I \wedge \frac{x+y}{2} = \frac{x'+y'+i}{4})).$$

Proof. By Lemma 3.1 (CoIClosure) we can write $x = \frac{x'+d}{2}$ and $y = \frac{y'+e}{2}$ with $d, e \in \mathrm{Sd}$ and $x', y' \in {}^{\mathrm{co}}I$. Then

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

Implicit algorithm. $f_{\mathrm{init}}\colon \mathbf{I} \to \mathbf{I} \to \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$ defined by[2]

$$f_{\mathrm{init}}(\mathrm{C}_d(u), \mathrm{C}_e(v)) = (d + e, u, v).$$

Throughout this paper we will use functions $J, K\colon \mathbb{Z} \to \mathbb{Z}$ such that

(6) $\forall_i(i = J(i) + 4K(i))$ (with $\forall_i(|J(i)| \le 2)$, $\forall_i(|i| \le 6 \to |K(i)| \le 1)$).

Lemma 3.3 (CoIAvcSatCoICl)**.**

$$\forall^{\mathrm{nc}}_{i,x,y}(i \in \mathrm{Sd}_2 \to x, y \in {}^{\mathrm{co}}I \to$$

$$\exists^{\mathrm{r}}_{j,d,x',y'}(j \in \mathrm{Sd}_2 \land d \in \mathrm{Sd} \land x', y' \in {}^{\mathrm{co}}I \land \frac{x+y+i}{4} = \frac{\frac{x'+y'+j}{4} + d}{2}).$$

Proof. By Lemma 3.1 (CoIClosure) we can write $x = \frac{x'+d}{2}$ and $y = \frac{y'+e}{2}$ with $d, e \in \mathrm{Sd}$ and $x', y' \in {}^{\mathrm{co}}I$. Then

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

Since $|d + e + 2i| \le 6$ we can write $d + e + 2i = j + 4k$ with $|j| \le 2$ and $|k| \le 1$ by the JK-property (6). Therefore

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

Implicit algorithm. $f\colon \mathbf{Sd}_2 \to \mathbf{I} \to \mathbf{I} \to \mathbf{Sd}_2 \times \mathbf{Sd} \times \mathbf{I} \times \mathbf{I}$ defined by

$$f(i, \mathrm{C}_d(u), \mathrm{C}_e(v)) = (J(d + e + 2i), K(d + e + 2i), u, v).$$

By coinduction from Lemma 3.3 we obtain

Lemma 3.4 (CoIAvcToCoI)**.**

$$\forall^{\mathrm{nc}}_z(\exists^{\mathrm{r}}_{i,x,y}(i \in \mathrm{Sd}_2 \land x, y \in {}^{\mathrm{co}}I \land z = \frac{x+y+i}{4}) \to z \in {}^{\mathrm{co}}I).$$

Proposition 3.5 (CoIAverage)**.**

$$\forall^{\mathrm{nc}}_{x,y}(x, y \in {}^{\mathrm{co}}I \to \frac{x+y}{2} \in {}^{\mathrm{co}}I).$$

Proof. Immediate from Lemmata 3.2 and 3.4. $\square$

[2]We use $\mathbf{Sd}_2 := \{\mathrm{RT}, \mathrm{RR}, \mathrm{MT}, \mathrm{LT}, \mathrm{LL}\}$ as a formal representation of the set Sd_2 of extended signed digits. – Formally $\mathrm{C}_d(u)$ ($:= \mathrm{C}(d, u)$) is not correct: we have $d \in \mathbb{Z}$, but the constructor C has type $\mathbf{Sd} \to \mathbf{I} \to \mathbf{I}$. First d has to be converted (by IntToSd) into an element of $\mathbf{Sd} := \{\mathrm{SdR}, \mathrm{SdM}, \mathrm{SdL}\}$. For readability such conversions are suppressed. However, they will show up in the extracted terms in Section 7.

Implicit algorithm. $f\colon \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I} \to \mathbf{I}$ defined corecursively by

$$f(i, \mathrm{C}_d(v), \mathrm{C}_e(w)) = \mathrm{C}_{K(d+e+2i)}(f(J(d+e+2i), v, w)).$$

More precisely, f_{init} (from Lemma 3.2 (CoIAvToAvc)) computes the first "carry" $i \in \mathrm{Sd}_2$ and the tails of the inputs. Then f is called repeatedly, computing the average step by step.

4. Multiplication for signed digit streams

Next we consider a proof that $[-1, 1]$ is closed under multiplication, w.r.t. the representation of reals as signed digit streams. Here we follow Ciaffaglione and Di Gianantonio [5], who found a nice way to reduce this problem to the one for the average function. Correspondingly our treatment uses material from Section 3.

4.1. **Informal proof.** Our goal this time is to prove

$$\forall^{\mathrm{nc}}_{x,x'}(x, x' \in {}^{\mathrm{co}}I \to x \cdot x' \in {}^{\mathrm{co}}I). \tag{7}$$

Again the computational content of this proof will be the desired algorithm. Consider the two sets

$$P := \{\, x \cdot y \mid x, y \in {}^{\mathrm{co}}I \,\}, \qquad Q := \{\, \frac{x \cdot y + z + i}{4} \mid x, y, z \in {}^{\mathrm{co}}I, i \in \mathrm{Sd}_2 \,\}.$$

It again suffices to show that Q satisfies Lemma 3.1 (CoIClosure)

$$\forall^{\mathrm{nc}}_x(x \in Q \to \exists^{\mathrm{r}}_{d,x'}(d \in \mathrm{Sd} \land x' \in Q \land x = \frac{x' + d}{2})),$$

for then by the greatest-fixed-point axiom for ${}^{\mathrm{co}}I$ we have $Q \subseteq {}^{\mathrm{co}}I$. Since we also have $P \subseteq Q$ we obtain $P \subseteq {}^{\mathrm{co}}I$, which is our claim.

For $P \subseteq Q$ — which is Lemma 4.4 (CoIMultToMultc) below — we need some auxiliary lemmata.

Lemma 4.1 (CoIUMinus)**.** $\forall^{\mathrm{nc}}_x(-x \in {}^{\mathrm{co}}I \to x \in {}^{\mathrm{co}}I)$.

Proof. By coinduction, using properties of the unary minus functions. □

Lemma 4.2 (CoIZero)**.** $0 \in {}^{\mathrm{co}}I$.

Proof. Let $P := \{\, x \mid x = 0 \,\}$. It suffices to show that P satisfies Lemma 3.1 (CoIClosure)

$$\forall^{\mathrm{nc}}_x(x \in P \to \exists^{\mathrm{r}}_{d,x'}(d \in \mathrm{Sd} \land x' \in P \land x = \frac{x' + d}{2})),$$

for then by the greatest-fixed-point axiom for ${}^{\mathrm{co}}I$ we have $P \subseteq {}^{\mathrm{co}}I$ and hence $0 \in {}^{\mathrm{co}}I$. Assume $x \in P$. Choose $d = 0$ and $x' = 0$. Then $d \in \mathrm{Sd}$, $x' \in P$ and also $x = \frac{x'+d}{2}$, since 0 is the only element of P. □

Lemma 4.3 (CoISdTimes)**.** $\forall^{\mathrm{nc}}_{x,d}(d \in \mathrm{Sd} \to x \in {}^{\mathrm{co}}I \to dx \in {}^{\mathrm{co}}I)$.

Proof. By the definition of Sd, using Lemma 4.1 (CoIUMinus) and in the zero case Lemma 4.2 (CoIZero). □

Lemma 4.4 (CoIMultToMultc)**.**

$$\forall^{\mathrm{nc}}_{x,y\in{}^{\mathrm{co}}I}\exists^{\mathrm{r}}_{y'\in{}^{\mathrm{co}}I}\exists^{\mathrm{r}}_{i\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x',z\in{}^{\mathrm{co}}I}(xy = \frac{x'y' + z + i}{4}).$$

Proof. By Lemma 3.1 (CoIClosure) we can write

$$x = \frac{x' + d}{2} \qquad y = \frac{y' + e}{2} \qquad \text{with } x', y' \in {}^{\mathrm{co}}I \text{ and } d, e \in \mathrm{Sd}.$$

Then

$$\frac{ex' + dy'}{2} \in {}^{\mathrm{co}}I$$

using Proposition 3.5 (CoIAverage) and Lemma 4.3 (CoISdTimes). Now again from Lemma 3.1 (CoIClosure) we obtain z, d_0 such that

$$\frac{ex' + dy'}{2} = \frac{z + d_0}{2}.$$

Therefore

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

which is of the required form. □

Lemma 4.5 (CoIMultcSatCoICl)**.**

$$\forall^{\mathrm{nc}}_{y\in{}^{\mathrm{co}}I}\forall^{\mathrm{nc}}_{i\in\mathrm{Sd}_2}\forall^{\mathrm{nc}}_{x,z\in{}^{\mathrm{co}}I}\exists^{\mathrm{r}}_{d\in\mathrm{Sd}}\exists^{\mathrm{r}}_{j\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x',z'\in{}^{\mathrm{co}}I}(\frac{xy + z + i}{4} = \frac{\frac{x'y+z'+j}{4} + d}{2}).$$

Proof. By Lemma 3.1 (CoIClosure) we can write

$$x = \frac{x_1 + d_1}{2} \qquad z = \frac{z_0 + d_0}{2} \qquad \text{with } x_1, z_1 \in {}^{\mathrm{co}}I \text{ and } d_1, d_0 \in \mathrm{Sd}.$$

Then

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

We have $d_1y \in {}^{\mathrm{co}}I$ by Lemma 4.3 (CoISdTimes) and $\frac{z_0+d_1y+i}{4} := v \in {}^{\mathrm{co}}I$ by Lemma 3.4 (CoIAvcToCoI). Hence we can continue the chain of equations by

$$= \frac{x_1y + 4v + d_0 + i}{8}.$$

Because of $v \in {}^{\mathrm{co}}I$ we can write

$$v = \frac{z_1 + e_0}{2} = \frac{\frac{z_2+e}{2} + e_0}{2} \qquad \text{with } z_1, z_2 \in {}^{\mathrm{co}}I \text{ and } e_0, e \in \mathrm{Sd}.$$

Therefore

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

Using again the functions J, K with the JK-property (6) we can write $e + 2e_0 + d_0 + i$ as $j + 4d$ with $j := J(e+2e_0+d_0+i)$ and $d := K(e+2e_0+d_0+i)$. Hence

$$= \frac{x_1 y + z_2 + j + 4d}{8} = \frac{\frac{x_1 y + z_2 + j}{4} + d}{2}. \qquad \square$$

Implicit algorithm. $f\colon \mathbf{I} \to \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I} \to \mathbf{Sd} \times \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$ defined by

$$f_v(i, \mathrm{C}_{d_1}(u), \mathrm{C}_{d_0}(w)) = (K(e + 2e_0 + d_0 + i), J(e + 2e_0 + d_0 + i), u, w')$$

where e, e_0, w' are obtained as follows. Let $g := \mathrm{cCoIAvcToCoI}$ and $f^* := \mathrm{cCoISdTimes}$. Then $g(i, f^*(d, v), w) \in {}^{\mathrm{co}}I$ by Lemma 3.4 (CoIAvcToCoI), and hence can be destructed into $\mathrm{C}_{e_0}(\mathrm{C}_e(w'))$.

By coinduction from Lemma 4.5 we obtain

Lemma 4.6 (CoIMultcToCoI)**.**

$$\forall_z^{\mathrm{nc}}(\exists_{i,x,y,z_0}^{\mathrm{r}}(i \in \mathrm{Sd}_2 \wedge x, y, z_0 \in {}^{\mathrm{co}}I \wedge z = \frac{xy + z_0 + i}{4}) \to z \in {}^{\mathrm{co}}I).$$

Proposition 4.7 (CoIMult)**.**

$$\forall_{x,y}^{\mathrm{nc}}(x, y \in {}^{\mathrm{co}}I \to xy \in {}^{\mathrm{co}}I).$$

Proof. Immediate from Lemmata 4.4 and 4.6. $\square$

Implicit algorithm. Lemma 4.4 (CoIMultToMultc) computes from the two inputs an initial quadruple i, x, y, z such that $\frac{xy+z+i}{4}$ is the product of the inputs. Fix y. Then $f_y\colon \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I} \to \mathbf{I}$ is defined corecursively by

$$f_y(i, \mathrm{C}_{d_1}(u), \mathrm{C}_{d_0}(w)) = \mathrm{C}_{K(e+2e_0+d_0+i)}(f_y(J(e + 2e_0 + d_0 + i), u, w')).$$

Here w' ($\sim z_2$) and e, e_0 are computed from the stream representation of y and d_1, w as described in the proof. Then f_y is called repeatedly, computing step by step the digits representing the product of the original inputs.

5. Average for pre-Gray code

We now consider the problem to compute the average of two real numbers given in pre-Gray code. The method is essentially the same as for signed digit streams; we only need to insert a different computational content to the predicates expressing how a real x is given. Instead of ${}^{\mathrm{co}}I$ for signed digit streams we now need two such predicates ${}^{\mathrm{co}}G$ and ${}^{\mathrm{co}}H$, corresponding to the two "modes" we have in pre-Gray codes.

5.1. **The predicates G, H and ${}^{\mathrm{co}}G, {}^{\mathrm{co}}H$.** We model pre-Gray codes as objects in the (simultaneously defined free) algebras $\mathbf{G}$ and $\mathbf{H}$ given by the constructors $\mathrm{Lr}\colon \mathbf{B} \to \mathbf{G} \to \mathbf{G}$, $\mathrm{U}\colon \mathbf{H} \to \mathbf{G}$ for $\mathbf{G}$ and $\mathrm{Fin}\colon \mathbf{B} \to \mathbf{G} \to \mathbf{H}$, $\mathrm{D}\colon \mathbf{H} \to \mathbf{H}$ for $\mathbf{H}$, with $\mathbf{B} = \{\mathtt{tt}, \mathtt{ff}\}$. We write $\mathrm{Lr}_1(p)$ for $\mathrm{Lr}(\mathtt{tt}, p)$ and $\mathrm{Lr}_{-1}(p)$ for $\mathrm{Lr}(\mathtt{ff}, p)$, and similarly for Fin. The predicates G, H and ${}^{\mathrm{co}}G, {}^{\mathrm{co}}H$ are defined as fixed points of the operators

$$\Gamma(X,Y) := \{\, x \mid \exists^{\mathrm{r}}_{a,x'}(a \in \mathrm{Psd} \land x' \in X \land x = -a\frac{x'-1}{2}) \lor \exists^{\mathrm{r}}_{x'}(x' \in Y \land x = \frac{x'}{2})\,\},$$

$$\Delta(X,Y) := \{\, x \mid \exists^{\mathrm{r}}_{a,x'}(a \in \mathrm{Psd} \land x' \in X \land x = a\frac{x'+1}{2}) \lor \exists^{\mathrm{r}}_{x'}(x' \in Y \land x = \frac{x'}{2})\,\}$$

Psd is a (formally inductive) predicate expressing that the integer a is a proper signed digit, i.e., $|a| = 1$. We will only need the greatest fixed point

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

which is expressed by the (strengthened) simultaneous coinduction axiom

$$(X,Y) \subseteq (\Gamma({}^{\mathrm{co}}G \cup X, {}^{\mathrm{co}}H \cup Y), \Delta({}^{\mathrm{co}}G \cup X, {}^{\mathrm{co}}H \cup Y)) \to (X,Y) \subseteq ({}^{\mathrm{co}}G, {}^{\mathrm{co}}H),$$

where inclusion $\subseteq$ is meant component-wise.

Similarly to what was done for ${}^{\mathrm{co}}I$ above we have compatibility of ${}^{\mathrm{co}}G$ and ${}^{\mathrm{co}}H$ with $=$, and can simplify the their clauses to

Lemma 5.1 (CoGClosure, CoHClosure)**.**

$$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}G \to \exists^{\mathrm{r}}_{a,x'}(a \in \mathrm{Psd} \land x' \in {}^{\mathrm{co}}G \land x = -a\frac{x'-1}{2}) \lor \exists^{\mathrm{r}}_{x'}(x' \in {}^{\mathrm{co}}H \land x = \frac{x'}{2})),$$

$$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}H \to \exists^{\mathrm{r}}_{a,x'}(a \in \mathrm{Psd} \wedge x' \in {}^{\mathrm{co}}G \wedge x = a\frac{x'+1}{2}) \vee$$
$$\exists^{\mathrm{r}}_{x'}(x' \in {}^{\mathrm{co}}H \wedge x = \frac{x'}{2})).$$

We also have their inverses

Lemma 5.2.

CoGClosureInvLr	$\forall^{\mathrm{nc}}_{a,x}(a \in \mathrm{Psd} \to x \in {}^{\mathrm{co}}G \to -a\frac{x-1}{2} \in {}^{\mathrm{co}}G)$,
CoGClosureInvU	$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}H \to \frac{x}{2} \in {}^{\mathrm{co}}G)$,
CoHClosureInvFin	$\forall^{\mathrm{nc}}_{a,x}(a \in \mathrm{Psd} \to x \in {}^{\mathrm{co}}G \to a\frac{x+1}{2} \in {}^{\mathrm{co}}H)$,
CoHClosureInvD	$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}H \to \frac{x}{2} \in {}^{\mathrm{co}}H)$.

5.2. **Realizability.** The realizability extensions $({}^{\mathrm{co}}G)^{\mathbf{r}}$ and $({}^{\mathrm{co}}H)^{\mathbf{r}}$ are binary predicates on cototal ideals p in $\mathbf{G}$ or q in $\mathbf{H}$ (respectively) and real numbers x. Consider the operators

$$\Gamma^{\mathbf{r}}(Z,W) := \{\,(p,x) \mid$$
$$\exists_{a,p',x'}(a \in \mathrm{Psd} \wedge (p',x') \in Z \wedge x = -a\frac{x'-1}{2} \wedge p = \mathrm{Lr}_a(p'))$$
$$\vee^{\mathrm{nc}} \exists_{q',x'}((q',x') \in W \wedge x = \frac{x'}{2} \wedge p = \mathrm{U}(q'))\,\},$$
$$\Delta^{\mathbf{r}}(Z,W) := \{\,(q,x) \mid$$
$$\exists_{a,p',x'}(a \in \mathrm{Psd} \wedge (p',x') \in Z \wedge x = a\frac{x'+1}{2} \wedge q = \mathrm{Fin}_a(p'))$$
$$\vee^{\mathrm{nc}} \exists_{q',x'}((q',x') \in W \wedge x = \frac{x'}{2} \wedge q = \mathrm{D}(q'))\,\}$$

(the $^{\mathrm{nc}}$ in $\vee^{\mathrm{nc}}$ indicates that the disjunction has no computational content). Since both $\Gamma^{\mathbf{r}}(Z,W)$ and $\Delta^{\mathbf{r}}(Z,W)$ are strictly positive in Z,W, our underlying theory provides us with a pair of binary predicates $({}^{\mathrm{co}}G)^{\mathbf{r}}$, $({}^{\mathrm{co}}H)^{\mathbf{r}}$ for the greatest fixed point of $(\Gamma^{\mathbf{r}},\Delta^{\mathbf{r}})$:

$$(({}^{\mathrm{co}}G)^{\mathbf{r}},({}^{\mathrm{co}}H)^{\mathbf{r}}) := \nu_{(Z,W)}(\Gamma^{\mathbf{r}}(Z,W),\Delta^{\mathbf{r}}(Z,W))$$

satisfying the (strengthened) simultaneous coinduction axiom

$$(Z,W) \subseteq (\Gamma^{\mathbf{r}}(({}^{\mathrm{co}}G)^{\mathbf{r}} \cup Z,({}^{\mathrm{co}}H)^{\mathbf{r}} \cup W),\Delta^{\mathbf{r}}(({}^{\mathrm{co}}G)^{\mathbf{r}} \cup Z,({}^{\mathrm{co}}H)^{\mathbf{r}} \cup W)) \to$$
$$(Z,W) \subseteq (({}^{\mathrm{co}}G)^{\mathbf{r}},({}^{\mathrm{co}}H)^{\mathbf{r}})$$

where again inclusion $\subseteq$ is meant component-wise.

5.3. **Informal proof.** We now consider the problem to compute the average of two real numbers given in pre-Gray code.

As a preparation we treat the unary minus function. Here we make use of the fact that our coinduction axioms are in strengthened form (that is $X \subseteq \Phi({}^{\mathrm{co}}I \cup X) \to X \subseteq {}^{\mathrm{co}}I$ instead of $X \subseteq \Phi(X) \to X \subseteq {}^{\mathrm{co}}I$, for example).

Lemma 5.3 (CoGUMinus, CoHUMinus)**.**

$$\forall^{\mathrm{nc}}_x(-x \in {}^{\mathrm{co}}G \to x \in {}^{\mathrm{co}}G),$$
$$\forall^{\mathrm{nc}}_x(-x \in {}^{\mathrm{co}}H \to x \in {}^{\mathrm{co}}H).$$

Proof. For $P := \{\, x \mid -x \in {}^{\mathrm{co}}G) \,\}$ and $Q := \{\, x \mid -x \in {}^{\mathrm{co}}H) \,\}$ we show $P \subseteq {}^{\mathrm{co}}G$ simultaneously with $Q \subseteq {}^{\mathrm{co}}H$. By coinduction it suffices to prove (i) $P \subseteq \Gamma({}^{\mathrm{co}}G \cup P, {}^{\mathrm{co}}H \cup Q)$ and (ii) $Q \subseteq \Delta({}^{\mathrm{co}}G \cup P, {}^{\mathrm{co}}H \cup Q)$. For (i), let $x_1 \in P$. We show $x_1 \in \Gamma({}^{\mathrm{co}}G \cup P, {}^{\mathrm{co}}H \cup Q)$:

$$\exists^{\mathrm{r}}_{x \in {}^{\mathrm{co}}G \cup P} \exists_a (x_1 = -a\frac{x-1}{2}) \vee \exists^{\mathrm{r}}_{x \in {}^{\mathrm{co}}H \cup Q}(x_1 = \frac{x}{2}). \tag{8}$$

The ${}^{\mathrm{co}}G$-clause applied to $-x_1 \in {}^{\mathrm{co}}G$ gives us

$$\exists^{\mathrm{r}}_{x \in {}^{\mathrm{co}}G} \exists_a (-x_1 = -a\frac{x-1}{2}) \vee \exists^{\mathrm{r}}_{x \in {}^{\mathrm{co}}H}(-x_1 = \frac{x}{2}).$$

In the first case we have $x_2 \in {}^{\mathrm{co}}G$ and a with $-x_1 = -a\frac{x_2-1}{2}$. Then the left hand side of (8) holds for x_2 and $-a$ (here we use that our coinduction axiom is in strengthened form). In the second case we have $x_2 \in {}^{\mathrm{co}}H$ with $-x_1 = \frac{x_2}{2}$. Then the right hand side of (8) holds for $-x_2$. This finishes the proof of (i). The proof of (ii) is similar, and we omit it. □

Implicit algorithm. $f\colon \mathbf{G} \to \mathbf{G}$ and $f'\colon \mathbf{H} \to \mathbf{H}$ defined by

$$f(\mathrm{Lr}_a(u)) = \mathrm{Lr}_{-a}(u), \qquad f'(\mathrm{Fin}_a(u)) = \mathrm{Fin}_{-a}(u),$$
$$f(\mathrm{U}(v)) = \mathrm{U}(f'(v)), \qquad f'(\mathrm{D}(v)) = \mathrm{D}(f'(v)).$$

Using Lemma 5.3 we prove that ${}^{\mathrm{co}}G$ and ${}^{\mathrm{co}}H$ are in fact equivalent.

Lemma 5.4 (CoHToCoG, CoGToCoH)**.**

$$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}H \to x \in {}^{\mathrm{co}}G),$$
$$\forall^{\mathrm{nc}}_x(x \in {}^{\mathrm{co}}G \to x \in {}^{\mathrm{co}}H).$$

Proof. We show ${}^{\mathrm{co}}H \subseteq {}^{\mathrm{co}}G$ simultaneously with ${}^{\mathrm{co}}G \subseteq {}^{\mathrm{co}}H$. By coinduction it suffices to prove (i) ${}^{\mathrm{co}}H \subseteq \Gamma({}^{\mathrm{co}}G \cup {}^{\mathrm{co}}H, {}^{\mathrm{co}}H \cup {}^{\mathrm{co}}G)$ and (ii) ${}^{\mathrm{co}}G \subseteq \Delta({}^{\mathrm{co}}G \cup {}^{\mathrm{co}}H, {}^{\mathrm{co}}H \cup {}^{\mathrm{co}}G)$. For (i), let $x_1 \in {}^{\mathrm{co}}H$. We show $x_1 \in \Gamma({}^{\mathrm{co}}G \cup {}^{\mathrm{co}}H, {}^{\mathrm{co}}H \cup {}^{\mathrm{co}}G)$:

$$\exists^{\mathrm{r}}_{x \in {}^{\mathrm{co}}G \cup {}^{\mathrm{co}}H} \exists_a (x_1 = -a\frac{x-1}{2}) \vee \exists^{\mathrm{r}}_{x \in {}^{\mathrm{co}}H \cup {}^{\mathrm{co}}G}(x_1 = \frac{x}{2}). \tag{9}$$

The ${}^{\mathrm{co}}H$-clause applied to $x_1 \in {}^{\mathrm{co}}H$ gives us

$$\exists^{\mathrm{r}}_{x\in{}^{\mathrm{co}}G}\exists_a(x_1 = a\frac{x+1}{2}) \vee \exists^{\mathrm{r}}_{x\in{}^{\mathrm{co}}H}(x_1 = \frac{x}{2}).$$

In the first case we have $x_2 \in {}^{\mathrm{co}}G$ and a with $x_1 = a\frac{x_2+1}{2}$. Then the left hand side of (9) holds for $-x_2$ and a, using Lemma 5.3 and (again) that our coinduction axiom is in strengthened form. In the second case we have $x_2 \in {}^{\mathrm{co}}H$ with $x_1 = \frac{x_2}{2}$. Then the right hand side of (8) holds for x_2. This finishes the proof of (i). The proof of (ii) is similar, and we omit it. □

Implicit algorithm. $g\colon \mathbf{H} \to \mathbf{G}$ and $h\colon \mathbf{G} \to \mathbf{H}$:

$$\begin{aligned} g(\mathrm{Fin}_a(u)) &= \mathrm{Lr}_a(f^-(u)), & h(\mathrm{Lr}_a(u)) &= \mathrm{Fin}_a(f^-(u)),\\ g(\mathrm{D}(v)) &= \mathrm{U}(v), & h(\mathrm{U}(v)) &= \mathrm{D}(v) \end{aligned}$$

where $f^- :=$ cCoGUMinus (cL denotes the function extracted from the proof of a lemma L). Notice that no corecursive call is involved.

The proof of the existence of the average w.r.t. Gray-coded reals is similar to the proof in Section 3.3 of the existence of the average w.r.t. signed digit stream coded reals. It proceeds as follows. To prove

$$\forall^{\mathrm{nc}}_{x,y}(x \in {}^{\mathrm{co}}G \to y \in {}^{\mathrm{co}}G \to \frac{x+y}{2} \in {}^{\mathrm{co}}G)$$

consider again two sets of averages, the second one with a "carry":

$$P := \{\, \frac{x+y}{2} \mid x, y \in {}^{\mathrm{co}}G \,\}, \qquad Q := \{\, \frac{x+y+i}{4} \mid x, y \in {}^{\mathrm{co}}G,\ i \in \mathrm{Sd}_2 \,\}.$$

It suffices to show that Q satisfies CoGClosure in Lemma 5.1, for then by the greatest-fixed-point axiom for ${}^{\mathrm{co}}G$ we have $Q \subseteq {}^{\mathrm{co}}G$. Since we also have $P \subseteq Q$ we obtain $P \subseteq {}^{\mathrm{co}}G$, which is our claim.

For $P \subseteq Q$ — which is Lemma 5.6 (CoGAvToAvc) below — we need

Lemma 5.5 (CoGPsdTimes)**.** $\forall^{\mathrm{nc}}_{a,x}(a \in \mathrm{Psd} \to x \in {}^{\mathrm{co}}G \to ax \in {}^{\mathrm{co}}G)$.

Proof. By the definition of Psd, using Lemma 5.3 (CoGUMinus). □

Lemma 5.6 (CoGAvToAvc)**.**

$$\forall^{\mathrm{nc}}_{x,y\in{}^{\mathrm{co}}G}\exists^{\mathrm{r}}_{i\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x',y'\in{}^{\mathrm{co}}G}(\frac{x+y}{2} = \frac{x'+y'+i}{4}).$$

Proof. Let $x, y \in {}^{\mathrm{co}}G$. By Lemma 5.1 (CoGClosure) there are two cases (Lr and U) for each of x, y. With CoHToCoG the argument is easy. □

Implicit algorithm. We use f^* for cCoGPsdTimes and s for cCoHToCoG.

$$\begin{aligned}
f(\mathrm{Lr}_a(u),\mathrm{Lr}_{a'}(u')) &= (a+a', f^*(-a,u), f^*(-a',u')),\\
f(\mathrm{Lr}_a(u),\mathrm{U}(v)) &= (a, f^*(-a,u), s(v)),\\
f(\mathrm{U}(v),\mathrm{Lr}_a(u)) &= (a, s(v), f^*(-a,u)),\\
f(\mathrm{U}(v),\mathrm{U}(v')) &= (0, s(v), s(v')).
\end{aligned}$$

Lemma 5.7 (CoGAvcSatCoICl)**.**

$$\forall^{\mathrm{nc}}_{i\in\mathrm{Sd}_2}\forall^{\mathrm{nc}}_{x,y\in{}^{\mathrm{co}}G}\exists^{\mathrm{r}}_{j\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{d\in\mathrm{Sd}}\exists^{\mathrm{r}}_{x',y'\in{}^{\mathrm{co}}G}(\frac{x+y+i}{4}=\frac{\frac{x'+y'+j}{4}+d}{2}).$$

Proof. Let $x,y\in{}^{\mathrm{co}}G$. Again by Lemma 5.1 (CoGClosure) there are two cases (Lr and U) for each of x, y. As in the proof of Lemma 3.3 we need the functions J,K defined there. □

Implicit algorithm.

$$\begin{aligned}
f(i,\mathrm{Lr}_a(u),\mathrm{Lr}_{a'}(u')) &= (J(a+a'+2i), K(a+a'+2i), f^*(-a,u), f^*(-a',u')),\\
f(i,\mathrm{Lr}_a(u),\mathrm{U}(v)) &= (J(a+2i), K(a+2i), f^*(-a,u), s(v)),\\
f(i,\mathrm{U}(v),\mathrm{Lr}_a(u)) &= (J(a+2i), K(a+2i), s(v), f^*(-a,u)),\\
f(i,\mathrm{U}(v),\mathrm{U}(v')) &= (J(2i), K(2i), s(v), s(v')).
\end{aligned}$$

By coinduction from Lemma 5.7 we obtain

Lemma 5.8 (CoGAvcToCoG)**.**

$$\begin{aligned}
&\forall^{\mathrm{nc}}_z(\exists^{\mathrm{r}}_{i\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x,y\in{}^{\mathrm{co}}G}(z=\frac{x+y+i}{4})\to z\in{}^{\mathrm{co}}G),\\
&\forall^{\mathrm{nc}}_z(\exists^{\mathrm{r}}_{i\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x,y\in{}^{\mathrm{co}}G}(z=\frac{x+y+i}{4})\to z\in{}^{\mathrm{co}}H).
\end{aligned}$$

Proof. We show $Q\subseteq{}^{\mathrm{co}}G$ simultaneously with $Q\subseteq{}^{\mathrm{co}}H$. By coinduction it suffices to prove (i) $Q\subseteq\Gamma({}^{\mathrm{co}}G\cup Q,{}^{\mathrm{co}}H\cup Q)$ and (ii) $Q\subseteq\Delta({}^{\mathrm{co}}G\cup Q,{}^{\mathrm{co}}H\cup Q)$. For (i), let $z_1\in Q$. We show $z_1\in\Gamma({}^{\mathrm{co}}G\cup Q,{}^{\mathrm{co}}H\cup Q)$:

$$\exists^{\mathrm{r}}_{z\in{}^{\mathrm{co}}G\cup Q}\exists^{\mathrm{r}}_{a\in\mathrm{Psd}}(z_1=-a\frac{z-1}{2})\vee\exists^{\mathrm{r}}_{z\in{}^{\mathrm{co}}H\cup Q}(z_1=\frac{z}{2}).\tag{10}$$

Lemma 5.7 applied to $z_1\in Q$ gives us $x_1,y_1\in{}^{\mathrm{co}}G$ and i_1,d_1 such that

$$z_1=\frac{\frac{x_1+y_1+i_1}{4}+d_1}{2}.$$

Case $d_1=0$. Go for the right hand side of (10) with $z:=(x_1+y_1+i_1)/4\in Q$. *Case* $d_1=\pm1$. Go for the left hand side of (10) with $a:=d_1$ and

$z := (-ax_1 - ay_1 - ai_1)/4 \in Q$. Then

$$-a\frac{z-1}{2} = -a\frac{4z-4}{8} = \frac{x_1+y_1+i_1+4a}{8} = z_1.$$

This finishes the proof of (i). The proof of (ii) is similar, and we omit it. □

Implicit algorithm. In the proof we used a lemma:

$$\mathrm{SdDisj}\colon \forall^{\mathrm{nc}}_{d\in\mathrm{Sd}}(d = 0 \vee^{\mathrm{r}} \exists^{\mathrm{r}}_{a\in\mathrm{Psd}}(d = a)).$$

Here $\vee^{\mathrm{r}}$ is an (inductively defined) variant of $\vee$ where only the content of the right hand side is kept.

$$\begin{aligned}
g(i,u,u') &= \text{let } (i_1,d,u_1,u_1') = \mathrm{cCoGAvcSatCoICl}(i,u,u') \text{ in}\\
&\quad \text{case } \mathrm{cSdDisj}(d) \text{ of}\\
&\quad\quad 0 \to \mathrm{U}(h(i,u_1,u_1'))\\
&\quad\quad a \to \mathrm{Lr}_a(g(-ai, f^*(-a,u_1), f^*(-a,u_1'))),\\
h(i,u,u') &= \text{let } (i_1,d,u_1,u_1') = \mathrm{cCoGAvcSatCoICl}(i,u,u') \text{ in}\\
&\quad \text{case } \mathrm{cSdDisj}(d) \text{ of}\\
&\quad\quad 0 \to \mathrm{D}(h(i,u_1,u_1'))\\
&\quad\quad a \to \mathrm{Fin}_a(g(-ai, f^*(-a,u_1), f^*(-a,u_1'))).
\end{aligned}$$

Proposition 5.9 (CoGAverage)**.**

$$\forall^{\mathrm{nc}}_{x,y}(x \in {}^{\mathrm{co}}G \to y \in {}^{\mathrm{co}}G \to \frac{x+y}{2} \in {}^{\mathrm{co}}G).$$

Proof. Compose Lemmata 5.6 and 5.8. □

6. Multiplication for pre-Gray code

Finally we consider a proof that $[-1,1]$ is closed under multiplication, w.r.t. the representation of reals in pre-Gray code. We will make use of material from Section 5.

6.1. **Informal proof.** Our goal is to find a proof of

$$\forall^{\mathrm{nc}}_{x,x'}(x, x' \in {}^{\mathrm{co}}G \to x \cdot x' \in {}^{\mathrm{co}}G), \tag{11}$$

which will give us the desired algorithm. Consider the two sets

$$P := \{\, x \cdot y \mid x, y \in {}^{\mathrm{co}}G \,\}, \qquad Q := \{\, \frac{x\cdot y + z + i}{4} \mid x, y, z \in {}^{\mathrm{co}}G, i \in \mathrm{Sd}_2 \,\}.$$

It again suffices to show that Q satisfies Lemma 5.1 (CoGClosure), for then by the greatest-fixed-point axiom for ${}^{\mathrm{co}}G$ we have $Q \subseteq {}^{\mathrm{co}}G$. Since we also have $P \subseteq Q$ we obtain $P \subseteq {}^{\mathrm{co}}G$, which is our claim.

We need an auxiliary lemma

Lemma 6.1 (CoGZero)**.** $0 \in {}^{\mathrm{co}}G$.

Proof. By coinduction. The proof is similar to the one for Lemma 4.2 (CoIZero), but we must prove the claim $0 \in {}^{\mathrm{co}}G$ simultaneously with $0 \in {}^{\mathrm{co}}H$. However, since in both cases we can use the second alternative in Lemma 5.1 (CoGClosure and CoHClosure), the proof is even simpler. □

Lemma 6.2 (CoGMultToMultc)**.**

$$\forall^{\mathrm{nc}}_{x,y\in{}^{\mathrm{co}}G}\exists^{\mathrm{r}}_{i\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x',y',z\in{}^{\mathrm{co}}G}(xy = \frac{x'y' + z + i}{4}).$$

Proof. We distinguish cases on $x \in {}^{\mathrm{co}}G$ and $y \in {}^{\mathrm{co}}G$ according to the disjunction in Lemma 5.1 (CoGClosure).

Case Lrx, Lry. Then we can write

$$x = -a\frac{x' - 1}{2} \qquad y = -b\frac{y' - 1}{2} \qquad \text{with } x', y' \in {}^{\mathrm{co}}G \text{ and } a, b \in \mathrm{Psd}.$$

By Proposition 5.9 (CoGAverage) and Lemma 5.5 (CoGPsdTimes) we have

$$\frac{-abx' - aby'}{2} =: z \in {}^{\mathrm{co}}G$$

We again need to distinguish cases according to the disjunction in Lemma 5.1 (CoGClosure). *Subcase* Lrz. Then we can write

$$z = -c\frac{z' - 1}{2} \qquad \text{with } z' \in {}^{\mathrm{co}}G \text{ and } c \in \mathrm{Psd}.$$

Therefore $xy =$

$$\frac{(ax' - a)(by' - b)}{4} = \frac{ax'by' - ab(x' + y') + ab}{4} = \frac{ax'by' - cz' + (c + ab)}{4},$$

which is of the required form. *Subcase* Uz. Then we can write

$$z = \frac{z'}{2} \qquad \text{with } z' \in {}^{\mathrm{co}}H.$$

Therefore

$$xy = \frac{(ax' - a)(by' - b)}{4} = \frac{ax'by' - ab(x' + y') + ab}{4} = \frac{ax'by' + z' + ab}{4},$$

which is of the required form because of Lemma 5.4 (CoHToCoG).

Case Lrx, Uy. Then we can write

$$x = -a\frac{x'-1}{2} \qquad y = \frac{y'}{2} \qquad \text{with } x' \in {}^{\mathrm{co}}G,\ y' \in {}^{\mathrm{co}}H \text{ and } a \in \mathrm{Psd}.$$

Therefore

$$xy = \frac{(-ax'+a)y'}{4} = \frac{-ax'y'+ay'}{4},$$

which is of the required form because of Lemma 5.4 (CoHToCoG).

Case Ux, Lry. Then we can write

$$x = \frac{x'}{2} \qquad y = -b\frac{y'-1}{2} \qquad \text{with } x' \in {}^{\mathrm{co}}H,\ y' \in {}^{\mathrm{co}}G \text{ and } b \in \mathrm{Psd}.$$

Therefore

$$xy = \frac{x'(-by'+b)}{4} = \frac{-bx'y'+bx'}{4},$$

which is of the required form because of Lemma 5.4 (CoHToCoG).

Case Ux, Uy. Then we can write

$$x = \frac{x'}{2} \qquad y = \frac{y'}{2} \qquad \text{with } x', y' \in {}^{\mathrm{co}}H.$$

Therefore

$$xy = \frac{x'y'}{4},$$

which is of the required form because of Lemma 6.1 (CoGZero). □

Implicit algorithm. We use s for cCoHToCoG. For brevity from now on we omit f^* (for cCoGPsdTimes) and simply write au for $f^*(a,u)$.

$$\begin{aligned}
g(\mathrm{Lr}_a(u), \mathrm{Lr}_b(v)) &= \text{case cCoGAverage}(-abu, -abv) \text{ of}\\
&\qquad \mathrm{Lr}_c(w) \to (c+ab, au, bv, -cw)\\
&\qquad \mathrm{U}(w) \to (ab, au, bv, s(w))\\
g(\mathrm{Lr}_a(u), \mathrm{U}(v)))\ &= (0, -au, s(v), as(v))\\
g(\mathrm{U}(u), \mathrm{Lr}_b(v))\ &= (0, s(u), -bv, bs(u))\\
g(\mathrm{U}(u), \mathrm{U}(v))\ &= (0, s(u), s(v), \text{cCoGZero}).
\end{aligned}$$

Lemma 6.3 (CoGMultcSatCoICl)**.**

$$\forall^{\mathrm{nc}}_{y\in{}^{\mathrm{co}}G}\forall^{\mathrm{nc}}_{i\in\mathrm{Sd}_2}\forall^{\mathrm{nc}}_{x,z\in{}^{\mathrm{co}}G}\exists^{\mathrm{r}}_{d\in\mathrm{Sd}}\exists^{\mathrm{r}}_{j\in\mathrm{Sd}_2}\exists^{\mathrm{r}}_{x',z'\in{}^{\mathrm{co}}G}\Big(\frac{xy+z+i}{4} = \frac{\frac{x'y+z'+j}{4}+d}{2}\Big).$$

Proof. We distinguish cases on $x \in {}^{\mathrm{co}}G$ and $z \in {}^{\mathrm{co}}G$ according to the disjunction in Lemma 5.1 (CoGClosure).

Case Lrx, Lrz. We can write

$$x = -a_1 \frac{x_1 - 1}{2} \qquad z = -a_0 \frac{z_1 - 1}{2} \qquad \text{with } x_1, z_1 \in {}^{\mathrm{co}}G \text{ and } a_1, a_0 \in \mathrm{Psd}.$$

Then

$$\begin{aligned} \frac{xy + z + i}{4} &= \frac{(-a_1 x_1 + a_1)y - a_0 z_1 + a_0 + 2i}{8} \\ &= \frac{-a_1 x_1 y + (a_1 y - a_0 z_1 + i) + a_0 + i}{8}. \end{aligned}$$

We have $a_1 y, -a_0 z_1 \in {}^{\mathrm{co}}G$ by Lemma 5.5 (CoGPsdTimes) and $\frac{a_1 y - a_0 z_1 + i}{4} =: v \in {}^{\mathrm{co}}G$ by Lemma 5.8 (CoGAvcToCoG). Hence we can continue

$$= \frac{-a_1 x_1 y + 4v + a_0 + i}{8} = \frac{\frac{-a_1 x_1 y}{4} + v + \frac{a_0 + i}{4}}{2} = \frac{\frac{-a_1 x_1 y + z + j}{4} + d}{2}$$

with $z \in {}^{\mathrm{co}}G$, $j \in \mathrm{Sd}_2$ and $d \in \mathrm{Sd}$, by Lemma 6.4 (JKLr).

Case Lrx, Uz. We can write

$$x = -a_1 \frac{x_1 - 1}{2} \qquad z = \frac{z_1}{2} \qquad \text{with } x_1 \in {}^{\mathrm{co}}G,\ z_1 \in {}^{\mathrm{co}}H \text{ and } a_1 \in \mathrm{Psd}.$$

Then

$$\frac{xy + z + i}{4} = \frac{(-a_1 x_1 + a_1)y + z_1 + 2i}{8} = \frac{-a_1 x_1 y + (a_1 y + z_1 + i) + i}{8}.$$

We have $a_1 y \in {}^{\mathrm{co}}G$ by Lemma 5.5 (CoGPsdTimes), $z_1 \in {}^{\mathrm{co}}G$ by Lemma 5.4 (CoHToCoG) and $\frac{a_1 y + z_1 + i}{4} =: v \in {}^{\mathrm{co}}G$ by Lemma 5.8 (CoGAvcToCoG). Hence we can continue

$$= \frac{-a_1 x_1 y + 4v + i}{8} = \frac{\frac{-a_1 x_1 y}{4} + v + \frac{i}{4}}{2} = \frac{\frac{-a_1 x_1 y + z + j}{4} + d}{2}$$

with $z \in {}^{\mathrm{co}}G$, $j \in \mathrm{Sd}_2$ and $d \in \mathrm{Sd}$, by Lemma 6.5 (JKU).

Case Ux, Lrz. We can write

$$x = \frac{x_1}{2} \qquad z = -a_0 \frac{z_1 - 1}{2} \qquad \text{with } x_1 \in {}^{\mathrm{co}}H,\ z_1 \in {}^{\mathrm{co}}G \text{ and } a_0 \in \mathrm{Psd}.$$

Then

$$\frac{xy + z + i}{4} = \frac{x_1 y - a_0 z_1 + a_0 + 2i}{8} = \frac{x_1 y + (0 - a_0 z_1 + i) + a_0 + i}{8}.$$

We have $0 \in {}^{\mathrm{co}}G$ by Lemma 6.1 (CoGZero), $-a_0 z_1 \in {}^{\mathrm{co}}G$ by Lemma 5.5 (CoGPsdTimes) and $\frac{0-a_0 z_1+i}{4} =: v \in {}^{\mathrm{co}}G$ by Lemma 5.8 (CoGAvcToCoG). Hence we can continue

$$= \frac{x_1 y + 4v + a_0 + i}{8} = \frac{\frac{x_1 y}{4} + v + \frac{a_0+i}{4}}{2} = \frac{\frac{x_1 y + z + j}{4} + d}{2}$$

with $z \in {}^{\mathrm{co}}G$, $j \in \mathrm{Sd}_2$ and $d \in \mathrm{Sd}$, by Lemma 6.4 (JKLr).

Case Ux, Uz. We can write

$$x = \frac{x_1}{2} \qquad z = \frac{z_1}{2} \qquad \text{with } x_1, z_1 \in {}^{\mathrm{co}}H.$$

Then

$$\frac{xy + z + i}{4} = \frac{x_1 y + z_1 + 2i}{8} = \frac{x_1 y + (0 + z_1 + i) + i}{8}.$$

We have $0 \in {}^{\mathrm{co}}G$ by Lemma 6.1 (CoGZero), $z_1 \in {}^{\mathrm{co}}G$ by Lemma 5.4 (CoHToCoG) and $\frac{0+z_1+i}{4} =: v \in {}^{\mathrm{co}}G$ by Lemma 5.8 (CoGAvcToCoG). Hence we can continue

$$= \frac{x_1 y + 4v + i}{8} = \frac{\frac{x_1 y}{4} + v + \frac{i}{4}}{2} = \frac{\frac{x_1 y + z + j}{4} + d}{2}$$

with $z \in {}^{\mathrm{co}}G$, $j \in \mathrm{Sd}_2$ and $d \in \mathrm{Sd}$, by Lemma 6.5 (JKU). □

Implicit algorithm. We use w_0 for cCoGZero and s for cCoHToCoG.

$g(v, i, \mathrm{Lr}_a(u), \mathrm{Lr}_c(w_1)) =$
  let $(j, d, w) = \mathrm{cJKLr}(i, c, \mathrm{cCoGAvcToCoG}(i, av, -cw_1))$ in $(d, j, -au, w)$
$g(v, i, \mathrm{Lr}_a(u), \mathrm{U}(w_1)) =$
  let $(j, d, w) = \mathrm{cJKU}(i, \mathrm{cCoGAvcToCoG}(i, av, s(w_1)))$ in $(d, j, -au, w)$
$g(v, i, \mathrm{U}(u), \mathrm{Lr}_c(w_1)) =$
  let $(j, d, w) = \mathrm{cJKLr}(i, c, \mathrm{cCoGAvcToCoG}(i, w_0, -cw_1))$ in $(d, j, s(u), w)$
$g(v, i, \mathrm{U}(u), \mathrm{U}(w_1)) =$
  let $(j, d, w) = \mathrm{cJKU}(i, \mathrm{cCoGAvcToCoG}(i, w_0, s(w_1)))$ in $(d, j, s(u), w)$

Lemma 6.4 (JKLr)**.**

$$\forall^{\mathrm{nc}}_{i \in \mathrm{Sd}_2} \forall^{\mathrm{nc}}_{a \in \mathrm{Psd}} \forall^{\mathrm{nc}}_{v \in {}^{\mathrm{co}}G} \exists^{\mathrm{r}}_{j \in \mathrm{Sd}_2} \exists^{\mathrm{r}}_{d \in \mathrm{Sd}} \exists^{\mathrm{r}}_{z \in {}^{\mathrm{co}}G} (v + \frac{a+i}{4} = \frac{z+j}{4} + d).$$

Proof. We distinguish cases on $v \in {}^{\mathrm{co}}G$ according to the disjunction in Lemma 5.1 (CoGClosure).

Case Lrv. Then $v = -b_0 \frac{w-1}{2}$ with $w \in {}^{\mathrm{co}}G$ and $b_0 \in \mathrm{Psd}$.

Subcase Lrw. Then we can write

$$v = -b_0\frac{w-1}{2} = -b_0\frac{-b\frac{z-1}{2}-1}{2} \qquad \text{with } w, z \in {}^{\text{co}}G \text{ and } b_0, b \in \text{Psd}.$$

Hence

$$v + \frac{a+i}{4} = \frac{b_0bz - b_0b + 2b_0 + a + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $-b_0b+2b_0+a+i$ as $j+4d$ with $j := J(-b_0b+2b_0+a+i)$ and $d := K(-b_0b+2b_0+a+i)$. Hence

$$= \frac{b_0bz + j + 4d}{4} = \frac{b_0bz + j}{4} + d.$$

Subcase Lrv, Uw. Then we can write

$$v = -b_0\frac{w-1}{2} = -b_0\frac{\frac{z}{2}-1}{2} \qquad \text{with } w \in {}^{\text{co}}G,\ z \in {}^{\text{co}}H \text{ and } b_0 \in \text{Psd}.$$

Hence

$$v + \frac{a+i}{4} = \frac{-b_0z + 2b_0 + a + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $2b_0 + a + i$ as $j + 4d$ with $j := J(2b_0 + a + i)$ and $d := K(2b_0 + a + i)$. Hence

$$= \frac{-b_0z + j + 4d}{4} = \frac{-b_0z + j}{4} + d.$$

Because of Lemma 5.4 (CoHToCoG) we obtain the required form.

Subcase Uv, Finw. Then we can write

$$v = \frac{w}{2} = \frac{b\frac{z+1}{2}}{2} \qquad \text{with } w \in {}^{\text{co}}H,\ z \in {}^{\text{co}}G \text{ and } b \in \text{Psd}.$$

Hence

$$v + \frac{a+i}{4} = \frac{bz + b + a + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $b + a + i$ as $j + 4d$ with $j := J(b + a + i)$ and $d := K(b + a + i)$. Hence

$$= \frac{bz + j + 4d}{4} = \frac{bz + j}{4} + d.$$

Subcase Uv, Dw. Then we can write

$$v = \frac{w}{2} = \frac{\frac{z}{2}}{2} \qquad \text{with } w, z \in {}^{\text{co}}H.$$

Hence

$$v + \frac{a+i}{4} = \frac{z + a + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $a + i$ as $j + 4d$ with $j := J(a + i)$ and $d := K(a + i)$. Hence

$$= \frac{z + j + 4d}{4} = \frac{z + j}{4} + d.$$

Because of Lemma 5.4 (CoHToCoG) we obtain the required form. □

Implicit algorithm. We use s for cCoHToCoG.

$$\begin{aligned}
g(i, a, \mathrm{Lr}_{b_0}(\mathrm{Lr}_b(w))) &= (J(-b_0 b + 2b_0 + a + i), K(-b_0 b + 2b_0 + a + i), b_0 b w) \\
g(i, a, \mathrm{Lr}_{b_0}(\mathrm{U}(w))) &= (J(2b_0 + a + i), K(2b_0 + a + i), -b_0 s(w)) \\
g(i, a, \mathrm{U}(\mathrm{Lr}_b(w))) &= (J(b + a + i), K(b + a + i), bw) \\
g(i, a, \mathrm{U}(\mathrm{U}(w))) &= (J(a + i), K(a + i), s(w))
\end{aligned}$$

Lemma 6.5 (JKU)**.**

$$\forall^{\mathrm{nc}}_{i \in \mathrm{Sd}_2} \forall^{\mathrm{nc}}_{v \in {}^{\mathrm{co}}G} \exists^{\mathrm{r}}_{j \in \mathrm{Sd}_2} \exists^{\mathrm{r}}_{d \in \mathrm{Sd}} \exists^{\mathrm{r}}_{z \in {}^{\mathrm{co}}G} (v + \frac{i}{4} = \frac{z + j}{4} + d)$$

Proof. As in the previous lemma we need to consider four subcases, this time with a missing. In detail: we distinguish cases on $v \in {}^{\mathrm{co}}G$ according to the disjunction in Lemma 5.1 (CoGClosure).

Case Lrv. Then $v = -b_0 \frac{w-1}{2}$ with $w \in {}^{\mathrm{co}}G$ and $b_0 \in \mathrm{Psd}$.

Subcase Lrw. Then we can write

$$v = -b_0 \frac{w - 1}{2} = -b_0 \frac{-b \frac{z-1}{2} - 1}{2} \qquad \text{with } w, z \in {}^{\mathrm{co}}G \text{ and } b_0, b \in \mathrm{Psd}.$$

Hence

$$v + \frac{i}{4} = \frac{b_0 b z - b_0 b + 2b_0 + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $-b_0 b + 2b_0 + i$ as $j + 4d$ with $j := J(-b_0 b + 2b_0 + i)$ and $d := K(-b_0 b + 2b_0 + i)$. Hence

$$= \frac{b_0 b z + j + 4d}{4} = \frac{b_0 b z + j}{4} + d.$$

Subcase Lrv, Uw. Then we can write

$$v = -b_0 \frac{w - 1}{2} = -b_0 \frac{\frac{z}{2} - 1}{2} \qquad \text{with } w \in {}^{\mathrm{co}}G,\ z \in {}^{\mathrm{co}}H \text{ and } b_0 \in \mathrm{Psd}.$$

Hence

$$v + \frac{i}{4} = \frac{-b_0 z + 2b_0 + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $2b_0 + i$ as $j + 4d$ with $j := J(2b_0 + i)$ and $d := K(2b_0 + i)$. Hence

$$= \frac{-b_0 z + j + 4d}{4} = \frac{-b_0 z + j}{4} + d.$$

Because of Lemma 5.4 (CoHToCoG) we obtain the required form.

Subcase Uv, Finw. Then we can write

$$v = \frac{w}{2} = \frac{b^{\frac{z+1}{2}}}{2} \qquad \text{with } w \in {}^{\mathrm{co}}H,\ z \in {}^{\mathrm{co}}G \text{ and } b \in \mathrm{Psd}.$$

Hence

$$v + \frac{i}{4} = \frac{bz + b + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write $b + i$ as $j + 4d$ with $j := J(b + i)$ and $d := K(b + i)$. Hence

$$= \frac{bz + j + 4d}{4} = \frac{bz + j}{4} + d.$$

Subcase Uv, Dw. Then we can write

$$v = \frac{w}{2} = \frac{\frac{z}{2}}{2} \qquad \text{with } w, z \in {}^{\mathrm{co}}H.$$

Hence

$$v + \frac{i}{4} = \frac{z + i}{4}.$$

Using the functions J, K with the JK-property (6) we can write i as $j + 4d$ with $j := J(i)$ and $d := K(i)$. Hence

$$= \frac{z + j + 4d}{4} = \frac{z + j}{4} + d.$$

Because of Lemma 5.4 (CoHToCoG) we obtain the required form. □

Implicit algorithm. We use s for cCoHToCoG.

$$\begin{aligned}
g(i, \mathrm{Lr}_{b_0}(\mathrm{Lr}_b(w))) &= (J(-b_0 b + 2b_0 + i), K(-b_0 b + 2b_0 + i), b_0 b w)\\
g(i, \mathrm{Lr}_{b_0}(\mathrm{U}(w))) &= (J(2b_0 + i), K(2b_0 + i), -b_0 s(w))\\
g(i, \mathrm{U}(\mathrm{Lr}_b(w))) &= (J(b + i), K(b + i), bw)\\
g(i, \mathrm{U}(\mathrm{U}(w))) &= (J(i), K(i), s(w))
\end{aligned}$$

By coinduction from Lemma 6.3 (CoGMultcSatCoICl) we obtain

Lemma 6.6 (CoGMultcToCoG)**.**

$$\forall^{\mathrm{nc}}_{z_0}(\exists^{\mathrm{r}}_{i\in \mathrm{Sd}_2}\exists^{\mathrm{r}}_{x,y,z\in {}^{\mathrm{co}}G}(z_0 = \frac{xy+z+i}{4}) \to z_0 \in {}^{\mathrm{co}}G),$$
$$\forall^{\mathrm{nc}}_{z_0}(\exists^{\mathrm{r}}_{i\in \mathrm{Sd}_2}\exists^{\mathrm{r}}_{x,y,z\in {}^{\mathrm{co}}G}(z_0 = \frac{xy+z+i}{4}) \to z_0 \in {}^{\mathrm{co}}H).$$

Proof. We show $Q \subseteq {}^{\mathrm{co}}G$ simultaneously with $Q \subseteq {}^{\mathrm{co}}H$. By coinduction it suffices to prove (i) $Q \subseteq \Gamma({}^{\mathrm{co}}G \cup Q, {}^{\mathrm{co}}H \cup Q)$ and (ii) $Q \subseteq \Delta({}^{\mathrm{co}}G \cup Q, {}^{\mathrm{co}}H \cup Q)$. For (i), let $z_0 \in Q$. We show $z_0 \in \Gamma({}^{\mathrm{co}}G \cup Q, {}^{\mathrm{co}}H \cup Q)$:

$$\exists^{\mathrm{r}}_{z\in {}^{\mathrm{co}}G\cup Q}\exists^{\mathrm{r}}_{a\in \mathrm{Psd}}(z_0 = -a\frac{z-1}{2}) \vee \exists^{\mathrm{r}}_{z\in {}^{\mathrm{co}}H\cup Q}(z_0 = \frac{z}{2}). \tag{12}$$

Lemma 6.3 applied to $z_0 \in Q$ gives us $x_1, y_1, z_1 \in {}^{\mathrm{co}}G$ and i_1, d_1 such that

$$z_0 = \frac{\frac{x_1y_1+z_1+i_1}{4} + d_1}{2}.$$

Case $d_1 = 0$. Go for the right hand side of (12) with $z := (x_1y_1+z_1+i_1)/4 \in Q$. *Case* $d_1 = \pm 1$. Go for the left hand side of (12) with $a := d_1$ and $z := (-ax_1y_1 - az_1 - ai_1)/4 \in Q$. Then

$$-a\frac{z-1}{2} = -a\frac{4z-4}{8} = \frac{x_1y_1 + z_1 + i_1 + 4a}{8} = z_0.$$

This finishes the proof of (i). The proof of (ii) is similar, and we omit it. □

Implicit algorithm.

$g(i, u, u', u'') = \mathrm{let}\ (d, j, u_1, u_1') = \mathrm{cCoGMultcSatCoICl}(u', i, u, u'')\ \mathrm{in}$
  $\mathrm{case}\ \mathrm{cSdDisj}(d)\ \mathrm{of}$
    $0 \to \mathrm{U}(h(j, u_1, u', u_1'))$
    $a \to \mathrm{Lr}_a(g(-aj, u_1, f^*(-a, u'), f^*(-a, u_1')))$,

$h(i, u, u', u'') = \mathrm{let}\ (d, j, u_1, u_1') = \mathrm{cCoGMultcSatCoICl}(u', i, u, u'')\ \mathrm{in}$
  $\mathrm{case}\ \mathrm{cSdDisj}(d)\ \mathrm{of}$
    $0 \to \mathrm{D}(h(j, u_1, u', u_1'))$
    $a \to \mathrm{Fin}_a(g(aj, u_1, f^*(a, u'), f^*(a, u_1')))$.

Proposition 6.7 (CoGMult)**.**

$$\forall^{\mathrm{nc}}_{x,y}(x \in {}^{\mathrm{co}}G \to y \in {}^{\mathrm{co}}G \to xy \in {}^{\mathrm{co}}G).$$

Proof. Compose Lemmata 6.2 and 6.6. □

7. Formalization and extraction for signed digits

All proofs in the previous sections have been formalized in the proof assistant Minlog. The formalization closely follows the informal proofs, and is not spelled out in detail here[3]. We present the realizers machine extracted from the formalized proofs and discuss the algorithms they represent. They involve recursion and corecursion operators where the original proofs used induction or coinduction axioms, and the conversion rules for these operators determine how the extracted terms can be used as programs. The results of such an analysis have been shown in the previous sections under the label "implicit algorithm".

7.1. **Corecursion.** Recall the type of the corecursion operator for $\mathbf{I}$:

$$ {}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{I}} \colon \tau \to (\tau \to \mathbf{Sd} \times (\mathbf{I} + \tau)) \to \mathbf{I}. \tag{13} $$

The type $\mathbf{Sd} \times (\mathbf{I} + \tau)$ appears since $\mathbf{I}$ has the single constructor C of type $\mathbf{Sd} \to \mathbf{I} \to \mathbf{I}$. The meaning of ${}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{I}} NM$ is defined by the conversion rule

$$ {}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{I}} NM \mapsto \mathrm{C}_{\pi_1(MN)}([\mathrm{id}^{\mathbf{I}\to\mathbf{I}}, \lambda_y({}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{I}} yM)]\pi_2(MN)). $$

We have used π_1, π_2 for the two projections of type $\rho \times \sigma$, and the notation $[f, g] \colon \rho + \sigma \to \tau$ (for $f \colon \rho \to \tau$ and $g \colon \sigma \to \tau$) defined by

$$ [f, g](z) := \begin{cases} f(x) & \text{if } z = \mathrm{inl}(x), \\ g(y) & \text{if } z = \mathrm{inr}(y). \end{cases} $$

7.2. **Notational conventions of Minlog.** Types:

`iv`	base type for the algebra $\mathbf{I}$
`rho=>sigma`	function type
`rho yprod sigma`	product type
`rho ysum sigma`	sum type

Variables (with fixed types)

`d, e`	of type $\mathbb{Z}$
`v`	of type $\mathbf{I}$
`dv`	of type $\mathbf{Sd} \times \mathbf{I}$
`ivw`	of type $\mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$

[3] See `http://www.minlog-system.de/`, which gives instructions on how to download (or clone) the system and the necessary software (Scheme in this case). The formalizations can be found in the directory `minlog/examples/analysis/grayrealeq.scm`.

`viuw`	of type $\mathbf{I} \times \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$
`diuw`	of type $\mathbf{Sd} \times \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$

If for a given type no specific variable names are provided, Minlog uses the name of the type as default variable name. For instance, `ag` and `ah` are variable names for the types **G** and **H**, respectively. Constants

`Rec, CoRec`	recursion, corecursion
`DesYprod`	destructor for products
`cL`	realizer for lemma L

Terms

`[x]r`	lambda abstraction $\lambda_x r$
`r pair s`	product term
`clft r, crht r`	components (prefix, binding strongest)
`InL, InR`	injections into a sum type

```
[v,v0][let ivw
   (IntToSdtwo(SdToInt clft(cCoIClosure v)+
               SdToInt clft(cCoIClosure v0))pair
   crht(cCoIClosure v)pair crht(cCoIClosure v0))
   ((CoRec sdtwo yprod iv yprod iv=>iv)ivw
   ([ivw0][let jdvw
       (IntToSdtwo
       (J(SdToInt clft(cCoIClosure clft crht ivw0)+
          SdToInt clft(cCoIClosure crht crht ivw0)+
          SdtwoToInt clft ivw0*2))pair
       IntToSd
       (K(SdToInt clft(cCoIClosure clft crht ivw0)+
          SdToInt clft(cCoIClosure crht crht ivw0)+
          SdtwoToInt clft ivw0*2))pair
       crht(cCoIClosure clft crht ivw0)pair
       crht(cCoIClosure crht crht ivw0))
     (clft crht jdvw pair
      InR(clft jdvw pair crht crht jdvw))]))]
```

FIGURE 3. Extracted term for CoIAverage.

7.3. **Average for signed digit streams.** We analyze the term in Figure 3 extracted from CoIAverage. First the arguments `v`, `v0` are destructed into their components (d, v), (e, w) and from these the triple `ivw` $:= (d{+}e, v, w)$ is formed, which is the first argument N of the corecursion operator. The step function M, when applied to an argument `ivw0` of type $\tau = \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$, operates as follows. It destructs `ivw0` into the form $(i, (d, v), (e, w))$, and builds `jdvw` as the quadruple $(J(d + e + 2i), K(d + e + 2i), v, w)$. Then it returns d (the first digit), and continues with a corecursive call to (j, v, w).

```
[v,v0][let viuw (cCoIMultToMultc v v0)
   ((CoRec iv yprod sdtwo yprod iv yprod iv=>iv)viuw
   ([viuw0][let diuw
             (cCoIMultcSatCoICl clft viuw0 crht viuw0)
     (clft diuw pair
      InR(clft viuw0 pair crht diuw))]))]
```

FIGURE 4. Extracted term for CoIMult.

```
[v,v0][let dv (cCoIClosure
 (cCoIAverage
  (cCoISdTimes clft(cCoIClosure v0)crht(cCoIClosure v))
  (cCoISdTimes clft(cCoIClosure v)crht(cCoIClosure v0))))
 (crht(cCoIClosure v0)pair
  cIntPlusSdToSdtwo clft dv(cIntTimesSdToSd
                            clft(cCoIClosure v)
                            clft(cCoIClosure v0))pair
  crht(cCoIClosure v)pair
  crht dv)]
```

FIGURE 5. Extracted term for CoIMultToMultc.

7.4. **Multiplication for signed digit streams.** We analyze the term in Figure 4 extracted from CoIMult. First the arguments `v`, `v0` are destructed into their components (d, v), (e, w) and from these the initial quadruple `viuw` is formed such that $\frac{xy+z+i}{4}$ is the product of the inputs, with u, v, w representing x, y, z. This is done by Lemma 4.4 (CoIMultToMultc) whose extracted term is shown in Figure 5. It is here where the we make use of Proposition 3.5 (CoIAverage), whose extracted term shows up as

```
[v,ivw]
 [let dv (cCoIClosure clft crht ivw)
 [let dv0 (cCoIClosure crht crht ivw)
 [let vde [let dv1 (cCoIClosure
  (cCoIAvcToCoI(clft ivw pair
                crht dv0 pair
                cCoISdTimes clft dv v)))
            (crht(cCoIClosure crht dv1)pair
             clft(cCoIClosure crht dv1)pair
             clft dv1)]
   (IntToSd(K(SdToInt clft crht vde+
              2*SdToInt crht crht vde+
              SdToInt clft dv0+
              SdtwoToInt clft ivw))pair
    IntToSdtwo(J(SdToInt clft crht vde+
                 2*SdToInt crht crht vde+
                 SdToInt clft dv0+
                 SdtwoToInt clft ivw))pair
    crht dv pair
    clft vde)]]]
```

FIGURE 6. Extracted term for CoIMultcSatCoICl.

`cCoIAverage`. This initial quadruple is the first argument N of the corecursion operator in Figure 4. The step function M, when applied to an argument `viuw0` of type $\tau = \mathbf{I} \times \mathbf{Sd}_2 \times \mathbf{I} \times \mathbf{I}$, operates as follows. Let `viuw0` represent (y, i, x, z). Destruct x, z and build $(i, (d_1, x_1), y, (d_0, z_0))$. Then $\frac{z_0+d_1y+i}{4} \in {}^{\mathrm{co}}I$ by Lemma 3.4 (CoIAvcToCoI), hence can be destructed into $(e_0, (e, z_2))$. This is done via Lemma 4.5 (CoIMultcSatCoICl), which yields a quadruple (d, j, u, w) with $d := K(e+2e_0+d_0+i)$ and $j := J(e+2e_0+d_0+i)$. Then it returns d (the first digit), and continues with a corecursive call to (j, u, w). The way CoIMultcSatCoICl operates can be seen from its extracted term in Figure 6. Note that it essentially relies on Proposition 3.5 (CoIAverage) again, since it calls its main auxiliary Lemma 3.3 (CoIAvcSatCoICl).

8. Formalization and extraction for Gray code

We extend what was done in Section 7 for Gray code. Here we essentially restrict ourselves to a display of the extracted terms.

8.1. **Simultaneous corecursion.** We will now need the simultaneous corecursion operators ${}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}}^{(\mathbf{G},\mathbf{H}),(\sigma,\tau)}$ and ${}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}}^{(\mathbf{G},\mathbf{H}),(\sigma,\tau)}$ for $\mathbf{G}$, $\mathbf{H}$, of type

$$\begin{aligned} {}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}}^{(\mathbf{G},\mathbf{H}),(\sigma,\tau)} &: \sigma \to \delta_{\mathbf{G}} \to \delta_{\mathbf{H}} \to \mathbf{G} \\ {}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}}^{(\mathbf{G},\mathbf{H}),(\sigma,\tau)} &: \tau \to \delta_{\mathbf{G}} \to \delta_{\mathbf{H}} \to \mathbf{H} \end{aligned} \tag{14}$$

with step types

$$\begin{aligned} \delta_{\mathbf{G}} &:= \sigma \to \mathbf{B} \times (\mathbf{G} + \sigma) + (\mathbf{H} + \tau), \\ \delta_{\mathbf{H}} &:= \tau \to \mathbf{B} \times (\mathbf{G} + \sigma) + (\mathbf{H} + \tau). \end{aligned}$$

The type $\mathbf{B} \times (\mathbf{G} + \sigma) + (\mathbf{H} + \tau)$ appears since $\mathbf{G}$ has the two constructors $\mathrm{Lr}\colon \mathbf{B} \to \mathbf{G} \to \mathbf{G}$ and $\mathrm{U}\colon \mathbf{H} \to \mathbf{G}$, and $\mathbf{H}$ has the two constructors $\mathrm{Fin}\colon \mathbf{B} \to \mathbf{G} \to \mathbf{H}$ and $\mathrm{D}\colon \mathbf{H} \to \mathbf{H}$. Omitting the upper indices of ${}^{\mathrm{co}}\mathcal{R}$, the terms ${}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}}NMM'$ and ${}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}}N'MM'$ are defined by the conversion rules

$${}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}}NMM' \mapsto \begin{cases} \mathrm{Lr}_{\pi_1(u)}([\mathrm{id}, \lambda_y({}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}}yMM')]\pi_2(u)) & \text{if } MN = \mathrm{inl}(u) \\ \mathrm{U}([\mathrm{id}, \lambda_z({}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}}zMM')]v) & \text{if } MN = \mathrm{inr}(v) \end{cases}$$

$${}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}}N'MM' \mapsto \begin{cases} \mathrm{Fin}_{\pi_1(u)}([\mathrm{id}, \lambda_y({}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}}yMM')]\pi_2(u)) & \text{if } M'N' = \mathrm{inl}(u) \\ \mathrm{D}([\mathrm{id}, \lambda_z({}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}}zMM')]v) & \text{if } M'N' = \mathrm{inr}(v) \end{cases}$$

8.2. **Notational conventions of Minlog (continued).** Types:

`ag`, `ah`	base types for the algebras $\mathbf{G}$, $\mathbf{H}$

Variables (with fixed types)

`btgh`	of type $\mathbf{B} \times \mathbf{G} + \mathbf{H}$
`bg`	of type $\mathbf{B} \times \mathbf{G}$
`gg`	of type $\mathbf{G} \times \mathbf{G}$
`igg`	of type $\mathbf{Sd}_2 \times \mathbf{G} \times \mathbf{G}$
`dgg`	of type $\mathbf{Sd} \times \mathbf{G} \times \mathbf{G}$
`jdgg`	of type $\mathbf{Sd}_2 \times \mathbf{Sd} \times \mathbf{G} \times \mathbf{G}$
`idg`	of type $\mathbf{Sd}_2 \times \mathbf{Sd} \times \mathbf{G}$
`iggg`	of type $\mathbf{Sd}_2 \times \mathbf{G} \times \mathbf{G} \times \mathbf{G}$
`djgg`	of type $\mathbf{Sd} \times \mathbf{Sd}_2 \times \mathbf{G} \times \mathbf{G}$

8.3. **Average for pre-Gray code.** From the proof of Lemma 5.3 (CoGU-Minus) we obtain the extracted term shown in Figure 7. It uses the simultaneous corecursion operators ${}^{\mathrm{co}}\mathcal{R}^{(\mathbf{G},\mathbf{H}),(\sigma,\tau)}_{\mathbf{G}}$, ${}^{\mathrm{co}}\mathcal{R}^{(\mathbf{G},\mathbf{H}),(\sigma,\tau)}_{\mathbf{H}}$. By analyzing the particular step functions M, M' extracted from our proof we see that we can write $\lambda_y {}^{\mathrm{co}}\mathcal{R}_{\mathbf{G}} y M M'$ and $\lambda_z {}^{\mathrm{co}}\mathcal{R}_{\mathbf{H}} z M M'$ as the two functions $f\colon \sigma \to \mathbf{G}$ and $f'\colon \tau \to \mathbf{H}$ shown as informal algorithm in Section 5.3. Note that the content `cCoHCompat` of Lemma `CoHCompat`: $\forall^{\mathrm{nc}}_{x,y}(x = y \to x \in {}^{\mathrm{co}}H \to y \in {}^{\mathrm{co}}H)$ is (essentially) the identity and can be ignored. – An easy consequence of Lemma 5.3 (CoGUMinus) is a lemma `CoGPsdTimes`: $\forall^{\mathrm{nc}}_{x,d}(x \in {}^{\mathrm{co}}G \to d \in \mathrm{Psd} \to xd \in {}^{\mathrm{co}}G)$ whose extracted term in shown in Figure 8. Here again `cCoGCompat` is the identity and can be ignored.

Another consequence of CoGUMinus was the equivalence of ${}^{\mathrm{co}}G$ and ${}^{\mathrm{co}}H$ proved in Lemma 5.4 (CoHToCoG); its extracted term is shown in Figure 9. Its operation clearly is described by the informal algorithm above.

Next Lemma 5.6 (CoGAvToAvc) gives us the term in Figure 10. Again its operation is described by the informal algorithm above.

For Lemma 5.7 (CoGAvcSatCoICl) we get the term in Figure 11, operation is described by the informal algorithm above.

By coinduction from Lemma 5.7 we obtained Lemma 5.8 (CoGAvcToCoG), whose extracted term is shown in Figure 12. It clearly describes the implicit algorithm above, using the Lemma SdDisj: $\forall^{\mathrm{nc}}_{d\in\mathrm{Sd}}(d = 0 \vee^{\mathrm{r}} \exists^{\mathrm{r}}_{a\in\mathrm{Psd}}(d = a))$.

Finally for Proposition 5.9 (CoGAverage) we just need to compose the extracted terms for Lemma 5.6 (CoGAvToAvc) and Lemma 5.8 (CoGAvc-ToCoG). The extracted term is

```
[ag,ag0]cCoGAvcToCoG(cCoGAvToAvc ag ag0).
```

8.4. **Multiplication for pre-Gray code.** Here we restrict ourselves to just printing the extracted terms.

References

[1] U. Berger, K. Miyamoto, H. Schwichtenberg, and H. Tsuiki. Logic for Gray-code computation. In D. Probst and P. Schuster, editors, *Concepts of Proof in Mathematics, Philosophy, and Computer Science*, pages 69–110. De Gruyter, 2016.

[2] U. Berger and M. Seisenberger. Proofs, programs, processes. In F. Ferreira et al., editors, *Proceedings CiE 2010*, volume 6158 of *LNCS*, pages 39–48. Springer Verlag, Berlin, Heidelberg, New York, 2010.

[3] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill, New York, 1967.

[4] E. Bishop and D. Bridges. *Constructive Analysis*, volume 279 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag, Berlin, Heidelberg, New York, 1985.

[5] A. Ciaffaglione and P. D. Gianantonio. A certified, corecursive implementation of exact real numbers. *Theoretical Computer Science*, 351:39–51, 2006.
[6] P. D. Gianantonio. An abstract data type for real numbers. *Theoretical Computer Science*, 221(1-2):295–326, 1999.
[7] K. Miyamoto and H. Schwichtenberg. Program extraction in exact real arithmetic. *Mathematical Structures in Computer Science*, 25:1692–1704, 2015.
[8] H. Schwichtenberg and S. S. Wainer. *Proofs and Computations*. Perspectives in Logic. Association for Symbolic Logic and Cambridge University Press, 2012.
[9] H. Tsuiki. Real number computation through Gray code embedding. *Theoretical Computer Science*, 284:467–485, 2002.
[10] T. Überrück Fries. Program extraction for exact real numbers: Stream multiplication. Bachelorarbeit, Mathematisches Institut der LMU, 2016.

```
[ag](CoRec ag=>ag ah=>ah)ag
 ([ag0][case (DesYprod ag0)
     (InL bg -> [case bg (boole pair ag1 ->
                 InL(cPsdUMinus boole pair InL ag1))])
     (InR ah -> InR(InR(cCoHCompat ah)))])
 ([ah][case (cCoHClosure ah)
     (InL bg -> [case bg (boole pair ag0 ->
                 InL(cPsdUMinus boole pair InL ag0))])
     (InR ah0 -> InR(InR(cCoHCompat ah0)))])
```

FIGURE 7. Extracted term for CoGUMinus.

```
[ag,boole][if boole
   (cCoGCompat ag)
   (cCoGCompat(cCoGUMinus(cCoGCompat(cCoGCompat ag))))]
```

FIGURE 8. Extracted term for CoGPsdTimes.

```
[ah](CoRec ah=>ag ag=>ah)ah
 ([ah0][case (DesYprod ah0)
     (InL bg -> InL(clft bg pair
                    InL(cCoGUMinus(cCoGCompat crht bg))))
     (InR ah1 -> InR(InL ah1))])
 ([ag][case (DesYprod ag)
     (InL bg -> InL(clft bg pair
                    InL(cCoGUMinus(cCoGCompat crht bg))))
     (InR ah0 -> InR(InL ah0))])
```

FIGURE 9. Extracted term for CoHToCoG.

```
[ag,ag0][case (DesYprod ag)
 (InL bg -> [case (DesYprod ag0)
   (InL bg0 ->
   cIntPlusPsdToSdtwo clft bg clft bg0 pair
   cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
   cCoGPsdTimes crht bg0(cPsdUMinus clft bg0))
   (InR ah ->
   cPsdToSdtwo clft bg pair
   cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair cCoHToCoG ah)])
 (InR ah -> [case (DesYprod ag0)
   (InL bg ->
   cPsdToSdtwo clft bg pair
   cCoHToCoG ah pair cCoGPsdTimes crht bg(cPsdUMinus clft bg))
   (InR ah0 -> MT pair cCoHToCoG ah pair cCoHToCoG ah0)])]
```

FIGURE 10. Extracted term for CoGAvToAvc.

```
[sdtwo,ag,ag0][case (DesYprod ag)
   (InL bg -> [case (DesYprod ag0)
     (InL bg0 -> IntToSdtwo(J(BooleToInt clft bg+
                             BooleToInt clft bg0+
                             SdtwoToInt sdtwo*2))pair
                 IntToSd(K(BooleToInt clft bg+
                           BooleToInt clft bg0+
                           SdtwoToInt sdtwo*2))pair
     cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
     cCoGPsdTimes crht bg0(cPsdUMinus clft bg0))
     (InR ah -> cSdtwoPsdToSdtwoJ sdtwo clft bg pair
                cSdtwoPsdToSdK sdtwo clft bg pair
                cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
                cCoHToCoG ah)])
   (InR ah -> [case (DesYprod ag0)
     (InL bg -> cSdtwoPsdToSdtwoJ sdtwo clft bg pair
                cSdtwoPsdToSdK sdtwo clft bg pair
                cCoHToCoG ah pair
                cCoGPsdTimes crht bg(cPsdUMinus clft bg))
     (InR ah0 -> cSdtwoToSdtwoJ sdtwo pair
                 cSdtwoToSdK sdtwo pair
                 cCoHToCoG ah pair
                 cCoHToCoG ah0)])]
```

FIGURE 11. Extracted term for CoGAvcSatCoICl.

```
[igg](CoRec sdtwo yprod ag yprod ag=>ag
           sdtwo yprod ag yprod ag=>ah)igg
 ([igg0][let jdgg
  (cCoGAvcSatCoICl clft igg0 clft crht igg0 crht crht igg0)
  [case (cSdDisj clft crht jdgg)
   (DummyL -> InR(InR(clft jdgg pair crht crht jdgg)))
   (Inr boole -> InL(boole pair InR
    (cIntTimesSdtwoPsdToSdtwo clft jdgg(cPsdUMinus boole)pair
     cCoGPsdTimes clft crht crht jdgg(cPsdUMinus boole)pair
     cCoGPsdTimes crht crht crht jdgg(cPsdUMinus boole))))]])
 ([igg0][let jdgg
  (cCoGAvcSatCoICl clft igg0 clft crht igg0 crht crht igg0)
  [case (cSdDisj clft crht jdgg)
   (DummyL -> InR(InR(clft jdgg pair crht crht jdgg)))
   (Inr boole ->
   InL(boole pair InR
    (cIntTimesSdtwoPsdToSdtwo clft jdgg boole pair
     cCoGPsdTimes clft crht crht jdgg boole pair
     cCoGPsdTimes crht crht crht jdgg boole)))]])
```

FIGURE 12. Extracted term for CoGAvcToCoG.

```
[ag,ag0][case (DesYprod ag)
   (InL bg -> [case (DesYprod ag0)
     (InL bg0 -> [case (DesYprod(cCoGAverage
       (cCoGPsdTimes crht bg
        (cPsdUMinus(cIntTimesPsdToPsd clft bg clft bg0)))
       (cCoGPsdTimes crht bg0
        (cPsdUMinus(cIntTimesPsdToPsd clft bg clft bg0)))))
       (InL bg1 ->
       cIntPlusPsdToSdtwo
         clft bg1(cIntTimesPsdToPsd clft bg clft bg0)pair
       cCoGPsdTimes crht bg clft bg pair
       cCoGPsdTimes crht bg0 clft bg0 pair
       cCoGPsdTimes crht bg1(cPsdUMinus clft bg1))
       (InR ah ->
       cPsdToSdtwo(cIntTimesPsdToPsd clft bg clft bg0)pair
       cCoGPsdTimes crht bg clft bg pair
       cCoGPsdTimes crht bg0 clft bg0 pair
       cCoHToCoG ah)])
     (InR ah -> MT pair
                cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
                cCoHToCoG ah pair
                cCoGPsdTimes(cCoHToCoG ah)clft bg)])
   (InR ah -> [case (DesYprod ag0)
     (InL bg -> MT pair
                cCoHToCoG ah pair
                cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
                cCoGPsdTimes(cCoHToCoG ah)clft bg)
     (InR ah0 -> MT pair
                 cCoHToCoG ah pair
                 cCoHToCoG ah0 pair
                 cCoGZero)])]
```

FIGURE 13. Extracted term for CoGMultToMultc.

```
[sdtwo,boole,ag]
 [case (DesYprod ag)
  (InL bg -> [case (DesYprod crht bg)
   (InL bg0 -> IntToSdtwo(J(~(BooleToInt clft bg0*
                              BooleToInt clft bg)+
                             2*BooleToInt clft bg+
                             BooleToInt boole+
                             SdtwoToInt sdtwo))pair
               IntToSd(K(~(BooleToInt clft bg0*
                           BooleToInt clft bg)+
                          2*BooleToInt clft bg+
                          BooleToInt boole+
                          SdtwoToInt sdtwo))pair
               cCoGPsdTimes(cCoGPsdTimes crht bg0 clft bg0)
                           clft bg)
    (InR ah -> IntToSdtwo(J(2*BooleToInt clft bg+
                            BooleToInt boole+
                            SdtwoToInt sdtwo))pair
                IntToSd(K(2*BooleToInt clft bg+
                          BooleToInt boole+
                          SdtwoToInt sdtwo))pair
                cCoGPsdTimes(cCoHToCoG ah)
                            (cPsdUMinus clft bg))])
   (InR ah -> [case (DesYprod ah)
    (InL bg -> IntToSdtwo(J(BooleToInt clft bg+
                            BooleToInt boole+
                            SdtwoToInt sdtwo))pair
                IntToSd(K(BooleToInt clft bg+
                          BooleToInt boole+
                          SdtwoToInt sdtwo))pair
                cCoGPsdTimes crht bg clft bg)
    (InR ah0 -> IntToSdtwo(J(BooleToInt boole+
                             SdtwoToInt sdtwo))pair
                 IntToSd(K(BooleToInt boole+
                           SdtwoToInt sdtwo))pair
                 cCoHToCoG ah0)])]
```

FIGURE 14. Extracted term for JKLr.

```
[sdtwo,ag][case (DesYprod ag)
 (InL bg -> [case (DesYprod crht bg)
  (InL bg0 -> IntToSdtwo(J(~(BooleToInt clft bg0*
                              BooleToInt clft bg)+
                            2*BooleToInt clft bg+
                            SdtwoToInt sdtwo))pair
              IntToSd(K(~(BooleToInt clft bg0*
                          BooleToInt clft bg)+
                        2*BooleToInt clft bg+
                        SdtwoToInt sdtwo))pair
              cCoGPsdTimes
               (cCoGPsdTimes crht bg0 clft bg0)clft bg)
  (InR ah -> IntToSdtwo(J(2*BooleToInt clft bg+
                          SdtwoToInt sdtwo))pair
             IntToSd(K(2*BooleToInt clft bg+
                       SdtwoToInt sdtwo))pair
             cCoGPsdTimes(cCoHToCoG ah)
                         (cPsdUMinus clft bg))])
 (InR ah -> [case (DesYprod ah)
  (InL bg -> IntToSdtwo(J(BooleToInt clft bg+
                          SdtwoToInt sdtwo))pair
             IntToSd(K(BooleToInt clft bg+
                       SdtwoToInt sdtwo))pair
             cCoGPsdTimes crht bg clft bg)
  (InR ah0 -> IntToSdtwo(J(SdtwoToInt sdtwo))pair
              IntToSd(K(SdtwoToInt sdtwo))pair
              cCoHToCoG ah0)])]
```

FIGURE 15. Extracted term for JKU.

```
[ag,sdtwo,ag0,ag1][case (DesYprod ag0)
   (InL bg -> [case (DesYprod ag1)
     (InL bg0 -> [let idg (cJKLr sdtwo clft bg0
        (cCoGAvcToCoG(sdtwo pair
                      cCoGPsdTimes ag clft bg pair
                      cCoGPsdTimes crht bg0(cPsdUMinus clft bg0))))
       (clft crht idg pair
        clft idg pair
        cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
        crht crht idg)])
     (InR ah -> [let idg
       (cJKU sdtwo
             (cCoGAvcToCoG(sdtwo pair
                           cCoGPsdTimes ag clft bg pair
                           cCoHToCoG ah)))
       (clft crht idg pair
        clft idg pair
        cCoGPsdTimes crht bg(cPsdUMinus clft bg)pair
        crht crht idg)])])
   (InR ah -> [case (DesYprod ag1)
     (InL bg -> [let idg (cJKLr sdtwo clft bg
       (cCoGAvcToCoG(sdtwo pair
                     cCoGZero pair
                     cCoGPsdTimes crht bg(cPsdUMinus clft bg))))
       (clft crht idg pair
        clft idg pair
        cCoHToCoG ah pair
        crht crht idg)])
     (InR ah0 -> [let idg
       (cJKU sdtwo(cCoGAvcToCoG(sdtwo pair
                                cCoGZero pair
                                cCoHToCoG ah0)))
       (clft crht idg pair
        clft idg pair
        cCoHToCoG ah pair
        crht crht idg)])])]
```

FIGURE 16. Extracted term for CoGMultcSatCoICl.

```
[iggg](CoRec sdtwo yprod ag yprod ag yprod ag=>ag
            sdtwo yprod ag yprod ag yprod ag=>ah)iggg
 ([iggg0][let djgg (cCoGMultcSatCoICl
                    clft crht crht iggg0 clft iggg0
                    clft crht iggg0 crht crht crht iggg0)
     [case (cSdDisj clft djgg)
      (DummyL -> InR(InR(clft crht djgg pair
                         clft crht crht djgg pair
                         clft crht crht iggg0 pair
                         crht crht crht djgg)))
      (Inr boole -> InL(boole pair
        InR(cIntTimesSdtwoPsdToSdtwo
            clft crht djgg(cPsdUMinus boole)pair
            clft crht crht djgg pair
            cCoGPsdTimes clft crht crht iggg0
                         (cPsdUMinus boole)pair
            cCoGPsdTimes crht crht crht djgg
                         (cPsdUMinus boole))))]])
 ([iggg0][let djgg (cCoGMultcSatCoICl
                    clft crht crht iggg0 clft iggg0
                    clft crht iggg0 crht crht crht iggg0)
     [case (cSdDisj clft djgg)
      (DummyL -> InR(InR(clft crht djgg pair
                         clft crht crht djgg pair
                         clft crht crht iggg0 pair
                         crht crht crht djgg)))
      (Inr boole -> InL(boole pair
        InR(cIntTimesSdtwoPsdToSdtwo clft crht djgg boole pair
            clft crht crht djgg pair
            cCoGPsdTimes clft crht crht iggg0 boole pair
            cCoGPsdTimes crht crht crht djgg boole)))]])
```

FIGURE 17. Extracted term for CoGMultcToCoG.