Workshop on Proof Theory
July 2008
Bern

From coinductive proofs to exact real arithmetic

Ulrich Berger
Swansea

Outline

- Introduction
- Foundations
- Coinductive uniform continuity
- Towards a general theory of digital computation
- Conclusion and further work

Proving as programming

Programs are nearly always incorrect. It it is difficult, if not impossible, to decide whether or not a program is correct.

Proofs can easily be checked for correctness.

From proof theory we know that proofs contain correct programs.

Therefore, use proofs as programs.

More precisely, replace programming by proving.

In this talk I will try to argue that indeed this works in practice.

From proofs to programs

From a proof $T \vdash d : A$ read off a program $\mathrm{ep}(d)$ "realising" A.

- The theory T and the proof calculus $\vdash$ should be close to usual (informal) mathematics. The elegance of abstract mathematics should be retained and not be obscured by tedious encoding of mathematical objects.
- The proof d should be directly related to the program $\mathrm{ep}(d)$.
- Realisability should relate the program with the specification A in the expected way, and should also give meaning to all subprograms, so that we have a *full* understanding of the extracted program.

These criteria are well met by intuitionistic theories of inductive and coinductive data types and corresponding adaptations of Kleene/Kreisel realisability.

Foundations

- Induction and coinduction
- Initial algebras and terminal coalgebras
- Realisability

Induction

$\Phi : \mathcal{P}(U) \to \mathcal{P}(U)$ is monotone if $X \subseteq Y$ implies $\Phi(X) \subseteq \Phi(Y)$.

A set $X \subseteq U$ is Φ-*closed* if $\Phi(X) \subseteq X$.

$\mu\Phi$, the set *inductively* defined by Φ, is the least Φ-closed set.

Closure	$\Phi(\mu\Phi) \subseteq \mu\Phi$
Induction	if $\Phi(X) \subseteq X$, then $\mu\Phi \subseteq X$

Example

$\Phi : \mathcal{P}(\mathbb{R}) \to \mathcal{P}(\mathbb{R})$,

$\Phi(X) := \{0\} \cup \{x + 1 \mid x \in X\}$

$\mu\Phi = \mathbb{N} = \{0, 1, 2, \ldots\}$.

Induction:

If $X(0)$ and $\forall x\,(X(x) \to X(x + 1))$,

then $\forall x \in \mathbb{N}\, X(x)$.

Coinduction

A set $X \subseteq U$ is Φ-*coclosed* if $X \subseteq \Phi(X)$.

$\nu\Phi$, the set *coinductively* defined by Φ, is the largest Φ-coclosed set.

Coclosure	$\nu\Phi \subseteq \Phi(\nu\Phi)$
Coinduction	if $X \subseteq \Phi(X)$, then $X \subseteq \nu\Phi$

Example

$\mathbb{I} := [-1, 1] = \{x \mid |x| \leq 1\} \subseteq \mathbb{R}$

$\mathrm{SD} := \{-1, 0, 1\}$

$\mathrm{av}_d(x) := (x + d)/2 \quad (d \in \mathrm{SD})$

$\Phi : \mathcal{P}(\mathbb{R}) \to \mathcal{P}(\mathbb{R})$

$\Phi(X) := \{x \in \mathbb{I} \mid \exists d \in \mathrm{SD}\, \exists x' \in X \;\; x = \mathrm{av}_d(x')\}$

Lemma: $\nu\Phi = \mathbb{I}$.

Proof: $\nu\Phi \subseteq \Phi(\nu\Phi) \subseteq \mathbb{I}$.

$\mathbb{I} \subseteq \Phi(\nu\Phi)$ is shown by coinduction.

Need to show $\mathbb{I} \subseteq \Phi(\mathbb{I})$: Let $x \in \mathbb{I}$.

If $x \geq 0$, take $d := 1$, otherwise $d := -1$. $x' := 2 \cdot x - 1$

Initial algebras

$\varphi : \mathcal{C} \to \mathcal{C}$ functor (if $h : \alpha \to \beta$, then $\mathbf{map}_\varphi(h) : \varphi(\alpha) \to \varphi(\beta)$).

A *φ-algebra* is a pair (α, f) where $\alpha \in \mathcal{C}$ and $f : \varphi(\alpha) \to \alpha$ is a $\mathcal{C}$-morphism.

Algebras form a category $\mathrm{Alg}(\varphi)$. A morphism $h : (\alpha, f) \to (\beta, g)$ is a $\mathcal{C}$-morphism $h : \alpha \to \beta$ such that $h \circ f = g \circ \mathbf{map}_\varphi(h)$.

An *initial φ-algebra* is an initial object in $\mathrm{Alg}(\varphi)$. If it exists, it is denoted $(\mu\varphi, \mathrm{In}_\varphi)$.

Hence, for any φ-algebra (β, g) there exists a unique $\mathrm{Alg}(\varphi)$-morphism from $(\mu\varphi, \mathrm{In}_\varphi)$ to (α, f), i.e., a $\mathcal{C}$-morphism $\mathrm{It}_\varphi g : \mu\varphi \to \beta$ such that $\mathrm{It}_\varphi g \circ \mathrm{In}_\varphi = g \circ \mathbf{map}_\varphi(\mathrm{It}_\varphi g)$.

Example

$\varphi(\alpha) = 1 + \alpha.$

$\mu\varphi = \mathbb{N}, \quad \mathrm{In}_\varphi = [0, (+1)] : 1 + \mathbb{N} \to \mathbb{N}.$

Let $[a, s] : 1 + \alpha \to \alpha$ be a φ-algebra.

The unique algebra morphism $\mathrm{It}_\varphi[a, s] : \mathbb{N} \to \alpha$ satisfies

$\mathrm{It}_\varphi[a, s] \circ [0, (+1)] = [a, s] \circ \mathbf{map}_\varphi(\mathrm{It}_\varphi[a, s])$, i.e.

$$\begin{aligned} \mathrm{It}_\varphi[a, s](0) &= a \\ \mathrm{It}_\varphi[a, s](n+1) &= s(\mathrm{It}_\varphi[a, s](n)) \end{aligned}$$

Terminal coalgebras

A φ-*coalgebra* is a pair (α, f) where $A \in \mathcal{C}$ and $f : \alpha \to \varphi(\alpha)$ is a $\mathcal{C}$-morphism.

Colgebras form a category $\mathrm{Coalg}(\varphi)$. A morphism $h : (\alpha, f) \to (\beta, g)$ is a $\mathcal{C}$-morphism $h : \alpha \to \beta$ such that $g \circ h = \mathbf{map}_\varphi(h) \circ f$.

A *terminal* φ-*colgebra* is a terminal object in $\mathrm{Coalg}(\varphi)$. If it exists, it is denoted $(\nu\varphi, \mathrm{Out}_\varphi)$.

Hence, for any φ-coalgebra (α, f) there exists a unique $\mathrm{Coalg}(\varphi)$-morphism from (α, f) to $(\nu\varphi, \mathrm{Out}_\varphi)$, i.e., a $\mathcal{C}$-morphism $\mathrm{Coit}_\varphi f : \alpha \to \nu\varphi$ such that $\mathrm{Out}_\varphi \circ \mathrm{Coit}_\varphi f = \mathbf{map}_\varphi(\mathrm{Coit}_\varphi f) \circ f$.

Example

$\varphi(\alpha) = \mathrm{SD} \times \alpha$ (recall $\mathrm{SD} = \{-1, 0, 1\}$).

$\nu\varphi = \mathrm{SDS} = \mathrm{SD}^\omega$, $\mathrm{Out}_\varphi = \langle \mathrm{hd}, \mathrm{tl} \rangle : \mathrm{SDS} \to \mathrm{SD} \times \mathrm{SDS}$.

Let $\langle \mathrm{cur}, \mathrm{nxt} \rangle : \alpha \to \mathrm{SD} \times \alpha$ be a φ-algebra.

$\mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle : \alpha \to \mathrm{SDS}$ the unique coalgebra morphism.

$$\langle \mathrm{hd}, \mathrm{tl} \rangle \circ \mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle) = \mathbf{map}_\varphi(\mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle) \circ \langle \mathrm{cur}, \mathrm{nxt} \rangle$$

$$\begin{aligned} \mathrm{hd}(\mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle(x)) &= \mathrm{cur}(x) \\ \mathrm{tl}(\mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle(x)) &= \mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle(\mathrm{nxt}(x)) \quad \text{i.o.w.} \\ \mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle(x) &= \mathrm{cur}(x) : \mathrm{Coit}_\varphi \langle \mathrm{cur}, \mathrm{nxt} \rangle(\mathrm{nxt}(x)) \end{aligned}$$

Realisability

A: formula in the language of inductive and coinductive definitions.

r: term in the language of initial algebras and terminal coalgebras.

We define the set $\{x^{\tau(A)} \mid x\,\mathbf{mr}\,A\}$ of realisers of A. The type $\tau(A)$ represents the "propositional skeleton" of A:

$$
\begin{array}{lcl}
\tau(P(t)) & = & 1 \quad \text{(a singleton type)} \\
\tau(X(t)) & = & \alpha \quad \text{(a type variable assoc. with } X\text{)} \\
\tau(A \overset{\Rightarrow}{\underset{\wedge}{\vee}} B) & = & \tau(A) \overset{\rightarrow}{\underset{\times}{+}} \tau(B) \\
\tau(\overset{\forall}{\exists}\, x.A) & = & \tau(A) \\
\tau((\overset{\nu}{\mu}\, X.\{\vec{x} \mid A\})(\vec{t})) & = & \overset{\nu}{\mu}\, \alpha.\tau(A)
\end{array}
$$

Realisability

$$
\begin{array}{lcl}
x\,\mathbf{mr}\,P(t) & = & P(t) \\
x\,\mathbf{mr}\,X(t) & = & \tilde{X}(x,t)\ (\tilde{X}\ \text{a pred. var. assoc. with}\ X) \\
\mathrm{inl}(x)\,\mathbf{mr}\,(A \vee B) & = & x\,\mathbf{mr}\,A \\
\mathrm{inr}(y)\,\mathbf{mr}\,(A \vee B) & = & y\,\mathbf{mr}\,B \\
(x,y)\,\mathbf{mr}\,(A \wedge B) & = & x\,\mathbf{mr}\,A \wedge y\,\mathbf{mr}\,B \\
f\,\mathbf{mr}\,(A \to B) & = & \forall x\,(x\,\mathbf{mr}\,A \to f(x)\,\mathbf{mr}\,B) \\
z\,\mathbf{mr}\,(\overset{\forall}{\exists}\,x.A) & = & \overset{\forall}{\exists}\,x.(z\,\mathbf{mr}\,A) \\
z\,\mathbf{mr}\,(\mu X.\{\vec{x} \mid A\})(\vec{t}) & = & (\mu\tilde{X}.\{(\mathrm{In}_{\lambda\alpha.\tau(A)}(y),\vec{x}) \mid y\,\mathbf{mr}\,A\})(z,\vec{t}) \\
z\,\mathbf{mr}\,(\nu X.\{\vec{x} \mid A\})(\vec{t}) & = & (\nu\tilde{X}.\{(y,\vec{x}) \mid \mathrm{Out}_{\lambda\alpha.\tau(A)}(y)\,\mathbf{mr}\,A\})(z,\vec{t})
\end{array}
$$

Example

$$\mathrm{C}_0 := \nu X.\{x \in \mathbb{I} \mid \exists d \in \mathrm{SD}\, \exists x' \in X \;\; x = \mathrm{av}_d(x')\}$$

$$\tau(\mathrm{C}_0(x)) = \nu\alpha.(1+1+1) \times \alpha = \nu\alpha.\mathrm{SD} \times \alpha = \mathrm{SD}^\omega = \mathrm{SDS}.$$

· **mr** $\mathrm{C}_0(\cdot)$ is the largest subset of $\mathrm{SDS} \times \mathbb{R}$ such that

$$a\,\mathbf{mr}\,\mathrm{C}_0(x) \;\Rightarrow\; x \in \mathbb{I} \wedge \exists x' \,(x = \mathrm{av}_{\mathrm{hd}(a)}(x') \wedge \mathrm{tl}(a)\;\mathbf{mr}\;x'$$

Hence, if $a = a_0 : a_1 : a_2 : \ldots$, then

$$a\;\mathbf{mr}\;\mathrm{C}_0(x) \;\Leftrightarrow\; x = \mathrm{av}_{a_0}(\mathrm{av}_{a_1}(\mathrm{av}_{a_2}\ldots))) = \sum_{i=0}^{\infty} a_i 2^{-(i+1)} =: \sigma(a)$$

Formal system

Language: Any first-order language with predicate variables and (co)inductively defined predicates from strictly positive operators.

Axioms:

1. Any "true" $\vee$-free formulas (if B is $\vee$-free, then $r\,\mathbf{mr}\,B \equiv B$).
2. (Co)closure, (co)induction.

Proofs: Intuitionistic.

Soundness Theorem

From a constructive proof of A one can extract a term realising A.

Examples

Theorem: $\mathrm{C}_0(x) \Leftrightarrow |x| \leq 1 \wedge \forall n \in \mathbb{N}\, \exists q \in \mathbb{Q}\, |x - q| \leq 2^{-n}$.

Extracted program: translation from signed digit representation to Cauchy-sequence representation and back.

Theorem: $\mathrm{C}_0(x) \wedge \mathrm{C}_0(y) \Rightarrow \mathrm{C}_0(x \cdot y)$.

Extracted program: implementation of multiplication w.r.t. the signed digit representation.

Related work on coinduction, coalgebra and realisability

Jacobs, Rutten, Adamek, Aczel, . . . : LTS, bisimilarity, non-wellfouned sets, ...

Mendler, Geuvers, Matthes, Uustalu, Vene, . . . : monotone inductive types (with witness of monotonicity as extra input).

Bertot, Gianantonio, Ciaffaglione, Niqui, Konecni, O'Connor, . . . : coinductive definition and verification of real number algorithms in proof assistants.

Tatsuta: q-realisability for monotone coinductive definitions.

Uniformly continuous functions

Let ϵ, δ range over positive rational numbers.

$\mathrm{B}_\delta(x) := \{y \in \mathbb{R} \mid |x - y| \leq \delta\}$

Two definitions of $f : \mathbb{I} \to \mathbb{R}$ being uniformly continuous (u.c.):

1. $\forall \epsilon > 0\, \exists \delta > 0\, \forall x \in \mathbb{I} \quad f[\mathrm{B}_\delta(x)] \subseteq \mathrm{B}_\epsilon(f(x))$

2. $\forall \epsilon > 0\, \exists \delta > 0\, \forall p \in \mathbb{I} \cap \mathbb{Q}\, \exists q \in \mathbb{Q} \quad f[\mathrm{B}_\delta(p)] \subseteq \mathrm{B}_\epsilon(p)$

We adopt the second. Why?

Coinductive u.c.: the idea

Recall that for a signed digit stream $a = a_0 : a_1 : a_2 : \ldots \in \mathrm{SDS}$

$$\sigma(a) = \sum_{i \geq 0} a_i 2^{-(i+1)} = \mathrm{av}_{a_0}(\mathrm{av}_{a_1}(\mathrm{av}_{a_2} \ldots)))) \in \mathbb{I}$$

A function $f : \mathbb{I} \to \mathbb{I}$ is *represented* by a stream transformer $\hat{f} : \mathrm{SDS} \to \mathrm{SDS}$ if $f \circ \sigma = \sigma \circ \hat{f}$.

We give a coinductive definition of u.c. such that from a constructive proof of the u.c. of a function $f : \mathbb{I} \to \mathbb{I}$ one can extract an algorithm for a stream transformer representing f.

Writing and reading digits

$\mathrm{av}_d[\mathbb{I}] = \mathbb{I}_d := \{x \mid |x - d/2| \leq 1/2\}.$

Hence $f[\mathbb{I}] \subseteq \mathbb{I}_d$ iff $(\mathrm{va}_d^{-1} \circ f)[\mathbb{I}] \subseteq \mathbb{I}$, and in that case

$$f(\sigma(a_0 : a_1 : \ldots)) = \mathrm{av}_d((\mathrm{va}_d^{-1} \circ f)(\sigma(a_0 : a_1 : \ldots)))$$

i.e. we can **write** a digit, ignoring the input stream (the good case).

In the bad case we have to **read** a digit from the input:

$$f(\sigma(a_0 : a_1 : \ldots)) = (f \circ \mathrm{av}_{a_0})(\sigma(a_1 : \ldots)).$$

If f is continuous, then, after reading enough digits, we will be able to write again, and so on.

Coinductive u.c.: informal definition

For every $X \subseteq \mathbb{R}^{\mathbb{I}}$ we define inductively $\mathcal{J}(X) \subseteq \mathbb{R}^{\mathbb{I}}$ as the least set $Y \subseteq \mathbb{R}^{\mathbb{I}}$ such that for all $f \in \mathbb{R}^{\mathbb{I}}$

Write	$d \in \mathrm{SD} \wedge g \in X$	$\Rightarrow$	$\mathrm{av}_d \circ g \in Y$
Read	$\forall d\, f \circ \mathrm{av}_d \in Y$	$\Rightarrow$	$f \in Y$

Intuitively, $f \in \mathcal{J}(X)$ if after reading enough digits $a_0, \ldots, a_{n-1}$, we can write a digit d and $\mathrm{va}_d \circ f \circ \mathrm{av}_{a_0}^{-1} \circ \ldots \circ \mathrm{av}_{a_{n-1}} \in X$.

Clearly $\mathcal{J}$ is monotonic, i.e. $X \subseteq Y$ implies $\mathcal{J}(X) \subseteq \mathcal{J}(Y)$.

Therefore, we can define C coinductively as the largest subset of $\mathbb{R}^{\mathbb{I}}$ such that for all f

$$f \in \mathrm{C}_1 \Rightarrow f \in \mathcal{J}(\mathrm{C})$$

Coinductive u.c.: formal definition

$$\mathrm{C}_1 := \nu X.\mu Y.\{\ f : \mathbb{I} \to \mathbb{R} \quad \mid \quad \exists d \in \mathrm{SD}\, \exists f' \in X \ \ f = \mathrm{av}_d \circ f'$$
$$\vee \quad \forall d \in \mathrm{SD}\, f \circ \mathrm{av}_d \in Y\}$$

Related work by Hinze, Altenkirch, Ghani, Hancock, Pattinson.

Correctness

$f \in \mathrm{C}_1$ iff f is u.c. and $f[\mathbb{I}] \subseteq \mathbb{I}$.

Proof

"If": coinduction.

"Only if": Show

$$\forall k \, \forall f \, (f \in \mathrm{C}_1 \Rightarrow \exists \delta \, \forall p \in \mathbb{I} \cap \mathbb{Q} \, \exists q \in \mathbb{Q} \quad f[\mathrm{B}_\delta(p)] \subseteq \mathrm{B}_{2^{-k}}(p))$$

by induction on k.

Program extraction

The program extracted from the correctness proof provides translations between the Cauchy and the coinductive representation of continuous functions.

Intuitively, the coinductive data type for continuous functions consists of non-wellfounded trees with two kinds of nodes:

- writing nodes: labelled with a digit and one subtree,

- reading nodes: no labels, but three subtrees.

The inductive part of the definition ensures that trees are "productive", i.e. have infinitely many writing nodes on each branch.

Closure under composition

The coinductive predicates C_0 and C_1 can be generalised to C_n for functions of n arguments, $n \geq 0$.

Theorem
If $f \in \mathrm{C}_n$ and $g_1 \in \mathrm{C}_m, \ldots, g_n \in \mathrm{C}_m$, then $f \circ (g_1, \ldots, g_n) \in \mathrm{C}_m$.

Proof: coinduction.

The extracted program composes trees.

SD-systems

For a function $f : \mathbb{I} \to \mathbb{R}$ define its *potential*

$$\mathrm{Pot}(f) := \inf\{k \mid \forall p \in \mathbb{I} \cap \mathbb{Q}\, \exists q \in \mathbb{Q} \quad f[\mathrm{B}_{2^{-k}}(p)] \subseteq \mathrm{B}_{1/4}(p)\}$$

An SD-*system* is a set $\mathcal{F}$ of functions of finite potential from $\mathbb{I}$ to $\mathbb{I}$ such that for all $f \in \mathcal{F}$

$$\exists d\, (f[\mathbb{I}] \subseteq \mathbb{I}_d \wedge \mathrm{av}_d^{-1} \circ f \in \mathcal{F}) \vee (\mathrm{Pot}(f) > 0 \wedge \forall d\, f \circ \mathrm{av}_d \in \mathcal{F})$$

Theorem If $\mathcal{F}$ is an SD-system, then $\mathcal{F} \subseteq \mathrm{C}_1$.

(Can be generalised to functions of any arity.)

This theorem provides a practical method for generating SD-implementations of real runctions.

Example: Polynomials of degree ≤ 2 mapping $\mathbb{I}^n$ to $\mathbb{I}$.

The iterated logistic map

$f_a : [0,1] \to [0,1]$ $(0 \le a \le 4)$, $\quad f_a(x) := a \cdot x \cdot (1-x)$.

For many a, the iterates $f_a^n(x)$ behave chaotically and are difficult to compute as n grows (double precision arithmetic yields meaningless results for $n \ge 50$).

Experiment: Compute $f_a^n(x)$ for fixed large n (say $n = 500$), but with slightly varying x (say, in some interval $[b, b + 10^{-100}]$.

The computations are identical for many output digits, and our algorithm, working on a tree, memoizes previous computations.

deep n i m k

n digits of $f^i(x)$ for 2^k values of x all of which have the same first m digits.

deep 60 60 180 5	8 seconds
deepNM 60 60 180 5	7 minutes (no memoisation)

Integration

Theorem
If $f \in \mathrm{C}$, then $\forall k\, \exists p\, |\int f - p| \leq 2^{1-k}$

where $\int f := \int_{-1}^{1} f(x)\,\mathrm{d}x$

Proof: We show $\forall k\, \forall f\, (f \in \mathrm{C} \Rightarrow \exists p\, |\int f - p| \leq 2^{1-k})$
by induction on k.

$k = 0$: Since $f \in \mathrm{C}$ implies $f[\mathbb{I}] \subseteq \mathbb{I}$ it follows that $\int f \leq 2$. Hence we can take $p := 0$.

$k + 1$: Easy, using the equations

$$\int f = \tfrac{1}{2} \int (\mathrm{va}_d \circ f) + d = \tfrac{1}{2}(\int (f \circ \mathrm{va}_{-1}) + \int (f \circ \mathrm{va}_1)).$$

Further simple applicatons

- Inductive definition of $x < y$.
- Coinductive definition of $x \leq y$.
- Aproximate splitting:
  If $x < y$, then for all $z \in \mathrm{C}_0$, $x \leq z$ or $z \leq y$.
- Automatic functions.

Towards a general theory of digital computation

- Generalised digits
- π
- Analytic functions

Generalised digits

Fix a category $\mathcal{C}$ (e.g. the category of sets and functions).

If $X, Y \in \mathcal{C}$, then $X \to Y$ is the class of $\mathcal{C}$-morphism from X to Y.

$f : X \to Y$ means $f \in X \to Y$.

A *digit system* is a pair (X, D) consisting of an object $X \in \mathcal{C}$ and a set $D \subseteq X \to X$. The elements of D are called *digits*.

Digital morphisms

Let (X, D) and (Y, E) be digit systems. For any set $F \subseteq X \to Y$ we define (inductively) $\mathcal{J}(F)$ as the least $G \subseteq X \to Y$ such that

(W) if $e \in E$ and $f \in F$, then $e \circ f \in G$;

(R) if $f \circ d \in G$ for all $d \in D$, then $f \in G$.

The set C is (coinductively) defined as the largest $F \subseteq X \to Y$ such that

$$F \subseteq \mathcal{J}(F)$$

i.e. C is the largest fixed point of $\mathcal{J}$.

Formally,

$$\mathrm{C} = \nu F.\mu G.\{e \circ f \mid e \in E, f \in F\} \cup \{h : X \to Y \mid \forall d \in D\, h \circ d \in G\}$$

Identity

Lemma

Let (X, D) be a digit systems.

(a) $\mathrm{id}_X \in \mathrm{C}_{D,D}$.

(b) $D \subseteq \mathrm{C}_{D,D}$.

Composition

Theorem

Let (X_i, D_i) (i=1,2,3) be digit systems.

If $f \in \mathrm{C}_{D_1,D_2}$ and $g \in \mathrm{C}_{D_2,D_3}$, then $g \circ f \in \mathrm{C}_{D_1,D_3}$.

The category of digit systems

We can define a category $\mathcal{D}$ whose objects are digit systems and whose morphism sets are $\mathcal{D}((X, D), (Y, E)) := \mathrm{C}_{(X,D),(Y,E)}$. Composition is inherited from $\mathcal{S}$.

Theorem

The category $\mathcal{D}$ inherits finite products from the base category $\mathcal{C}$.

Remark: Probably $\mathcal{D}$ inherits all limits that exist in the base category.

But $\mathcal{D}$ seems to have almost never co-products.

Digital global elements

Let (X, D) be a digit system. We set

$$\mathrm{C}_D := \mathrm{C}_{\mathbf{1},(X,D)}, \qquad \mathcal{J}_D := \mathcal{J}_{\mathbf{1},(X,D)}$$

where $\mathbf{1}$ denotes the terminal object $(\mathbf{1}, \{\mathrm{id}_{\mathbf{1}}\})$ in $\mathcal{D}$. We identify C_D with a subset of X.

Lemma

C_D is the largest subset A of X such that

> if $x \in A$, then there exist $d \in D$ and $x' \in A$ such that $x = d(x')$, i.e.

$$\mathrm{C}_D = \nu A.\{x \in X \mid \exists d \in D\, \exists x' \in A\, (x = d(x'))\}$$

Metric digit spaces

A metric digit space $X = (X, \sigma, D)$ is called

contracting if there is a constant $c < 1$ such that for all $d \in D$ and $x, x' \in X$, $\sigma(d(x), d(x')) \leq c \cdot \sigma(x, x')$;

injective if all $d \in D$ are injective and have u.c. inverses $d^{-1} : d[X] \to X$;

uniformly covering if there is a $\epsilon > 0$ such that for all $x \in X$ there exists $d \in D$ with $\mathrm{B}_\epsilon(x) \subseteq d[X]$.

Characterisation of u.c.

Theorem

Let $X = (X, \sigma, P, D)$ and $Y = (Y, \tau, Q, E)$ be metric digit spaces. Set $\mathrm{U} := \{f : X \to Y \mid f \text{ u.c.}\}$ (w.r.t. σ and τ) and $\mathrm{C} := \mathrm{C}_{D,E}$.

(a) If X is bounded and contracting, and Y is injective and uniformly covering, then $\mathrm{U} \subseteq \mathrm{C}$.

(b) Assume D is finite. If X is injective and uniformly covering, and Y is bounded and contracting, then $\mathrm{C} \subseteq \mathrm{U}$.

Corollary

Let (X, σ) be a bounded metric space. Let $D, E \subseteq X \to X$. If (X, σ, D) is contracting and (X, σ, E) is injective and uniformly covering, then $\mathrm{C}_D \subseteq \mathrm{C}_E$.

π

Corollary

For the metric digit system $(\mathbb{I}, D)$ where $D := \{\mathrm{av}_1, \mathrm{av}_0, \mathrm{av}_{-1}\} \subseteq \mathbb{I} \to \mathbb{I}$ we have $\pi/4 \in \mathrm{C}_D$.

Proof We use the formula

$$\frac{\pi}{4} = \frac{1}{2} + \frac{1}{3}\left(\frac{1}{2} + \frac{2}{5}\left(\frac{1}{2} + \frac{3}{7}\left(\frac{1}{2} + \frac{4}{9}\left(\frac{1}{2} + \ldots\right)\right)\right)\right)$$

i.e. $\pi/4 = f_0(f_1(\ldots))$ where

$$f_n(x) := \frac{1}{2} + \frac{n\,x}{2n+1}.$$

Hence we have $\pi/4 \in \mathrm{C}_F$ where $F := \{f_n \mid n \in \mathbb{N}\}$. Since the f_n are all contracting (with common contraction factor $1/2$), and $(\mathbb{I}, D)$ is injective and uniformly covering, it follows $\pi/4 \in \mathrm{C}_D$.

Analytic functions

Theorem

Let $q > 0$ and $a_n \in \mathbb{R}$ $(n \in \mathbb{N})$ such that $|a_{n+1}| \leq q \cdot |a_n|$ for all $n \in \mathbb{N}$. Let $u, v \geq 0$ such that $|a_0|, u \leq q \cdot v^2$ and $q \cdot (u + v) < 1$. Set $X := \mathrm{B}_u(0)$ and $Y := \mathrm{B}_v(0)$. Then $f : X \to Y$,

$$f(x) := \sum_{n=0}^{\infty} a_n \cdot x^n$$

is well-defined, and $f \in \mathrm{C}_{\hat{P}}$ where

$\hat{P} := \{\hat{p_n} : (X \to Y) \to X \to Y \mid n \in \mathbb{N}\}$,
$\hat{p}(f, x) := p(x, f(x)) \quad (p : X \times Y \to Y)$,
$P := \{p_{b_n,q} : X \times Y \to Y \mid n \in \mathbb{N}\}$ with $b_n := a_n/q^n$,
$p_n(x, y) := b_n + q \cdot x \cdot y$ with $b_n := a_n/q^n$.

The Curry Lemma

In order to obtain a digital implementation of an analytic function f we need to show $f \in \mathrm{C}_{D,E}$ for suitable D, E.

But we only got $f \in \mathrm{C}_{\hat{P}}$.

Lemma

Let (X, D) and (Y, E) be digit systems, and assume that $A \subseteq (X \to Y) \to (X \to Y)$ is such that $\mathrm{uncurry}(A) \subseteq \mathrm{C}_{A \otimes D, E}$. Then $\mathrm{C}_A \subseteq \mathrm{C}_{D,E}$.

Hence it suffices to find a set $A \subseteq (X \to Y) \to (X \to Y)$ such that $\hat{P} \subseteq A$ (P defined as in the previous theorem) and $\mathrm{uncurry}(A) \subseteq \mathrm{C}_{A \otimes D, E}$.

The Contraction Theorem

Theorem
Let $D \subseteq X \to X$ uniformly contracting, $E \subseteq Y \to Y$ uniformly covering and s.t. all $e \in E$ are injective with a uniform Lipschitz constant for the inverses.

For $p : X \times Y \to Y$ and $q : X \to X$ define

$$\varphi_{p,q} : (X \to Y) \times X \to Y, \quad \varphi_{p,q}(f, x) := p(x, f(q(x)))$$

Let $\lambda < 1$ and $\gamma \geq 0$. Define

$$A := \{\varphi_{p,q} : p \ \lambda\text{-contracting}, \ q \ \gamma\text{-contracting} \} \subseteq (X \to Y) \times X \to Y$$

Then $A \subseteq \mathrm{C}_{\mathrm{curry}(A) \otimes D, E}$.

Further work

We would like to apply the the general theory to compute approximations to the compact subsets of a compact metric space, viewed as elemnts of the compact metric space of non-empty compact sets with the Hausdorff metric.

Unfortunately, on that space no finite system of contracting and uniformly covering digits exists.

This non-existence holds for a large class of metric spaces.

We are working on a further generalisation of digital computation that covers such situations.

Joint work with Dieter Spreen.

Conclusion

- Case studies show that "proofs as programs" works.
- New (correct!) programs extracted that would have been difficult to "guess".
- Using a fine tuning of realisability it is possible to do abstract mathematics as usual, and still get computational content.
- Generalisations to arbitrary digit streams and metric spaces? (see also Escardo, Scriven, Hutchinson)
- Related work by Edalat, Potts, Heckmann, Ciaffaglione, Gianantonio, Niqui.
- A lot of interesting work on program extraction has been done in the Coq community (Bertot, Russell O'Connor).