Domains IX
Brighton, 22-24 September 2008

A coinductive approach to digital computation

Ulrich Berger
Swansea

Outline

- Introduction
- Induction and coinduction
- Digit spaces
- Metric digit spaces
- Program extraction
- Case studies
- Conclusion

The aims of this talk

- to outline a constructive theory of digital computation;
- to show that program extraction from proofs is a practical method for obtaining certified programs for digital computation.

Background

- Exact Real Arithmetic via infinite streams of Signed Digits and Linear Fractional Transformations
- Coalgebraic modelling of infinite data types
- Proof Theory: Program extraction from proofs via realisability
- Domain-theoretic modelling and termination proofs

Example: computing with signed digits

$\mathbb{I} := [-1, 1] \subseteq \mathbb{R}$
$\mathrm{SD} := \{-1, 0, 1\}$
$x \in \mathbb{I}$
$a = (a_n)_{n \in \mathbb{N}} \in \mathrm{SD}^\omega$

$$x \sim a \quad :\Leftrightarrow \quad x = \sum_{n=0}^{\infty} a_n \cdot 2^{-(n+1)}$$

A function $f : \mathbb{I} \to \mathbb{I}$ is *represented* by a function $\hat{f} : \mathrm{SD}^\omega \to \mathrm{SD}^\omega$ if

$$\forall x, a \; (\, x \sim a \Rightarrow f(x) \sim \hat{f}(a) \,)$$

Power series as infinite composition

$$\sum_{n=0}^{\infty} a_n \cdot 2^{-(n+1)} = \frac{1}{2}(a_0 + \frac{1}{2}(a_1 + \ldots))$$

$\mathrm{av}_d : \mathbb{I} \to \mathbb{I}, \quad \mathrm{av}_d(x) := \frac{1}{2}(d + x) \quad (d \in \mathrm{SD}).$

$$\sum_{n=0}^{\infty} a_n \cdot 2^{-(n+1)} = \mathrm{av}_{a_0}(\mathrm{av}_{a_1}(\ldots)) = \mathrm{av}_{a_0} \circ \mathrm{av}_{a_1} \circ \ldots$$

Therefore, $x \sim a \;\Leftrightarrow\; x = \mathrm{av}_{a_0} \circ \mathrm{av}_{a_1} \circ \ldots$.

$\mathrm{AV} := \{\mathrm{av}_{-1}, \mathrm{av}_0, \mathrm{av}_1\} \subseteq \mathbb{I} \to \mathbb{I}.$

$(\mathbb{I}, \mathrm{AV})$ is an example of a *digit space*.

Digit spaces

We study digit spaces (X, D), where X is a set and $D \subseteq X \rightarrow X$, and characterise the functions $f : X \rightarrow Y$ that have a continuous digital representation $\hat{f} : D^\omega \rightarrow E^\omega$.

The characterisation does not refer to infinite objects (like streams of digits), but uses a combined inductive/coinductive definition.

Program extraction yields implementations of $\hat{f}$ by finitely branching non-wellfounded trees.

We also consider *metric digit spaces* (X, σ, P, D), where σ is a metric on X and $P \subseteq X$ is dense, and study the relation between digital representability and uniform continuity.

Induction

$\Phi \colon \mathcal{P}(U) \to \mathcal{P}(U)$ 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 $\mu\Phi \subseteq X$

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 $X \subseteq \nu\Phi$

Digital maps (motivation)

Let $x = \mathrm{av}_{a_0} \circ \mathrm{av}_{a_1} \circ \ldots$. Consider $h\colon \mathbb{I} \to \mathbb{I}$ (e.g. a polynomial)

Writing a digit

$$\begin{aligned} h(x) &= h \circ \mathrm{av}_{a_0} \circ \mathrm{av}_{a_1} \circ \ldots \\ &= \mathrm{av}_b \circ ((\mathrm{av}_b^{-1} \circ h) \circ \mathrm{av}_{a_0} \circ \mathrm{av}_{a_1} \circ \ldots) \qquad \text{if } h[\mathbb{I}] \subseteq \mathrm{av}_b[\mathbb{I}] \end{aligned}$$

Reading a digit

$$\begin{aligned} h(x) &= h \circ \mathrm{av}_{a_0} \circ \mathrm{av}_{a_1} \circ \ldots \\ &= (h \circ \mathrm{av}_{a_0}) \circ \mathrm{av}_{a_1} \circ \ldots \end{aligned}$$

After reading finitely many digits writing must be possible again.

Digital maps (definition)

Let (X, D) and (Y, E) be digit spaces.
We define the set $\mathrm{C}_{D,E} \subseteq X \to Y$ of *digital maps* as follows.

Let F, G range over subsets of $X \to Y$
and let $\nu F. \ldots$ stand for $\nu \lambda F \ldots$ e.t.c.

$\mathrm{C}_{D,E} :=$

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

Identity and composition

Identity Lemma

Let (X, D) be a digit spaces.

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

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

Composition Lemma

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

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 spaces

By the Identity Lemma and the Composition Lemma, digit spaces and digital maps form a category.

Product Lemma

The category $\mathcal{D}$ has finite products.

Digital global elements

The set of global elements of a digit space (X, D) is

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

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.

Global Element Lemma

$$\mathrm{C}_D = \nu A.\{d(x) \mid d \in D, x \in A\} \overset{\text{roughly}}{=} \{d_0 \circ d_1 \circ \ldots \mid (d_n)_{n\in\mathbb{N}} \in D^\omega\}$$

Application Lemma

If $f \in \mathrm{C}_{D,E}$ and $x \in \mathrm{C}_D$, then $f(x) \in \mathrm{C}_E$.

Proof: Composition Lemma.

Metric spaces

A *metric space* $X = (X, \sigma, P)$ consists of

- a set X,
- a metric σ on X,
- a dense set $P \subseteq X$ (of *concrete* elements).

For a rational number $\epsilon > 0$ and $p \in P$ we define

$$\mathrm{B}_\epsilon(p) := \{x \in X \mid \sigma(p, x) \leq \epsilon\}$$

For convenience, we only work with metric spaces that are *bounded*, i.e. $X \subseteq \mathrm{B}_M(p)$ for some $M > 0$ and $p \in P$.

Uniform continuity

Let $X = (X, P, \sigma)$ and $Y = (Y, Q, \tau)$ be metric spaces.

In the following: $p \in P$, $q \in Q$, $a, b \in \mathbb{R}^+$, $\delta, \epsilon \in \mathbb{Q}^+$,
$\hat{a} := \{\delta \in \mathbb{Q}^+ \mid \delta \leq a\} =]0, a]$.

A *modulus* is a relation $m \subseteq \mathbb{Q}^+ \times \mathbb{Q}^+$ such that

$$\forall b \, \exists a \;\; m[\hat{a}] \subseteq \hat{b}$$

A relation $f \subseteq X \times Y$ is *m-continuous*, if

$$\forall \delta, p \, \exists \epsilon \in m(\delta) \, \exists q \;\; f[\mathrm{B}_\delta(p)] \subseteq \mathrm{B}_\epsilon(q)$$

A relation f is *uniformly continuous (u.c.)* if it is m-continuous for some modulus m.

Properties of uniform continuity

Lemma

A relation $f \subseteq X \times Y$ is u.c. iff it is a partial function which is uniformly continuous on its domain,
$\mathrm{dom}(f) := \{x \in X \mid \exists y \in Y\, (x, y) \in f\}$, in the usual sense, i.e.

$$\forall \epsilon\, \exists \delta\, \forall x, x' \in \mathrm{dom}(F)$$

$$\sigma(x, x') \leq \delta \Rightarrow \tau(f(x), f(x')) \leq \epsilon$$

Composition Lemma

If $g \subseteq Y \times Z$ and $f \subseteq X \times Y$ are uniformly continuous, so is $g \circ f \subseteq X \times Z$.

Lipschitz conditions and contractivity

For $\lambda \geq 0$ we define

$$m_\lambda := \{(\delta, \epsilon) \mid \epsilon \leq \lambda\delta\}$$

Clearly, m_λ is a modulus and $m_\lambda \circ m_\gamma = m_{\lambda\gamma}$.

A relation $f \subseteq X \times Y$ is called *Lipschitz* if it is m_λ-continuous for some $\lambda \geq 0$. If $\lambda < 1$, we call f *contracting*.

Lemma A relation $f \subseteq X \times Y$ is λ-continuous iff it is a partial function and $\tau(f(x), f(x')) \leq \lambda \cdot \sigma(x, x')$ for all $x, x' \in \mathrm{dom}(f)$. Hence a function is Lipschitz iff it is Lipschitz in the usual sense.

Metric digit spaces

A *metric digit space* $X = (X, \sigma, P, D)$ is a metric space (X, σ, P) together with a set of digits $D \subseteq X \to X$.

(X, σ, P, D) is called

- *contracting* if there is $\lambda < 1$ such that all $d \in D$ are contracting with factor λ.
- *invertible* if d^{-1} is u.c. for all $d \in D$.
- *covering* if there is an $\epsilon_0 > 0$ such that, for all $\epsilon > 0$ and $p \in P$, either there exists $d \in D$ with $\mathrm{B}_\epsilon(p) \subseteq d[X]$, or $\epsilon > \epsilon_0$.
- *finitely covering* if there is a finite subset of D which is uniformly covering.

Example: $(\mathbb{I}, \mathrm{AV})$ has all these properties.

Characterisation of uniform continuity

Characterisation Lemma

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

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

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

Corollary (change of digits)

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

Proof

The identity function on X is u.c. and hence in $\mathrm{C}_{D,E}$, by (a).

The type of a formula

To every formula A we assign the type $\tau(A)$ of its *realisers*, i.e. the type a program extracted from a proof of A will have:

- $\tau(A)$ is the unit type if A contains neither $\vee$ nor predicate variables (A may contain predicate constants like "$=$", "$\leq$" and "$\in \mathbb{R}$").
- The propositional connectives $\wedge$, $\vee$, $\Rightarrow$ are translated into the type constructors $\times$, $+$, $\to$.
- Quantifiers and terms are ignored.
- Predicate variables are translated into type variables.
- Inductive and coinductive definitions are translated into initial algebras and terminal coalgebras, respectively.

Example: τ("f is uniformly continuous")

Recall that $f : \mathbb{I} \to \mathbb{I}$ is uniformly continuous if there is some modulus m such that

$$\forall \delta, p \, \exists \epsilon \in m(\delta) \, \exists q \;\; f[\mathrm{B}_\delta(p)] \subseteq \mathrm{B}_\epsilon(q)$$

Since $\delta, \epsilon, p, q \in \mathbb{Q}$ and $\tau(\epsilon \in \mathbb{Q}) = \mathbb{Q}$, and furthermore $\tau(\epsilon \in m(\delta))$ is the unit type, we have

$$\tau(f \text{ u.c}) = \mathbb{Q} \times \mathbb{Q} \to \mathbb{Q} \times \mathbb{Q}$$

Example: $\tau(\mathrm{C}_{\mathrm{AV}})$

Recall the definition of $\mathrm{C}_{\mathrm{AV}} \subseteq \mathbb{I}$:

$$
\begin{aligned}
\mathrm{C}_{\mathrm{AV}} &= \nu A \,.\, \{d(x) \in \mathbb{I} \mid d \in \mathrm{AV}, x \in A\} \\
&= \nu A \,.\, \{y \in \mathbb{R} \mid -1 \leq y \leq 1 \land \\
&\qquad\qquad \exists d, x\, (d \in \mathrm{AV} \land x \in A \land y = \mathrm{av}_a(x))\}
\end{aligned}
$$

where

$\mathrm{AV} = \{\mathrm{av}_a \mid a \in \mathrm{SD}\} = \{d : \mathbb{R} \to \mathbb{R} \mid \exists a \in \mathrm{SD}\, d = \mathrm{av}_a\}$

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

Therefore

$$
\begin{aligned}
\tau(\mathrm{C}_{\mathrm{AV}}) &= \nu\alpha \,.\, \mathrm{SD} \times \alpha \\
&= \mathrm{SD}^{\omega}
\end{aligned}
$$

Example: $\tau(\mathrm{C}_{\mathrm{AV},\mathrm{AV}})$

Recall the definition of $\mathrm{C}_{\mathrm{AV},\mathrm{AV}} \subseteq \mathbb{I} \to \mathbb{I}$:

$$
\begin{aligned}
\mathrm{C}_{\mathrm{AV},\mathrm{AV}} &= \nu F \,.\, \mu G \,. \\
&\quad \{e \circ f : \mathbb{I} \to \mathbb{I} \mid e \in \mathrm{AV}, f \in F\} \cup \\
&\quad \{h : \mathbb{I} \to \mathbb{I} \mid \forall d \in \mathrm{AV}\; h \circ \mathrm{av}_d \in G\} \\
&= \nu F \,.\, \mu G \,. \\
&\quad \{h : \mathbb{R} \to \mathbb{R} \mid h[\mathbb{I}] \subseteq \mathbb{I} \wedge \\
&\qquad\qquad (\exists e, f\, (e \in \mathrm{AV} \wedge f \in F \wedge h = e \circ f) \vee \\
&\qquad\qquad (h \circ d_{-1} \in G \wedge h \circ d_0 \in G \wedge h \circ d_1 \in G))\}
\end{aligned}
$$

Therefore

$$
\tau(\mathrm{C}_{\mathrm{AV},\mathrm{AV}}) = \nu\alpha \,.\, \mu\beta \,.\, \mathrm{SD} \times \alpha + \beta^3
$$

See also [Hancock, Pattinson, Ghani]

Understanding $\tau(\mathrm{C}_{\mathrm{AV},\mathrm{AV}}) = \nu\alpha\,.\,\mu\beta\,.\,\mathrm{SD}\times\alpha+\beta^3$

Define T as the largest solution of the domain equation

$$T = \mathrm{SD} \times T + T^3$$

i.e. the elements of T are non-wellfounded trees with two kinds of nodes:

- **Writing nodes:** $W(d, t)$ where $d \in \mathrm{SD}$ and $t \in T$.
- **Reading nodes:** $R(t_{-1}, t_0, t_1)$ where $t_i \in T$.

$\tau(\mathrm{C}_{\mathrm{AV},\mathrm{AV}})$ is the set of those trees in T that have on every infinite path infinitely many writing nodes.

Realising inductive definitions

Assume the set operator Φ corresponds to the type operator φ.

Then, the inductively defined set $\mu\Phi$ together with the axioms

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

are realised by the initial algebra $(\mu\varphi, \mathrm{In}_\varphi)$
and the family It_φ of universal arrows, i.e.

$$\begin{aligned} \mathrm{In}_\varphi &: \varphi(\mu\varphi) \to \mu\varphi \\ \mathrm{It}_\varphi[s] &: \mu\varphi \to \alpha \quad (s : \varphi(\alpha) \to \alpha) \end{aligned}$$

with the defining recursion equation expressing that $\mathrm{It}_\varphi[s]$ is an algebra morphism

$$\mathrm{It}_\varphi[s] \circ \mathrm{In}_\varphi = s \circ \mathbf{map}_\varphi(\mathrm{It}_\varphi[s])$$

Hence, the function $\mathrm{It}_\varphi[s]$ is defind by *structural recursion* with step function s.

Realising coinductive definitions

For coinductive definitions the situation is dual.

The coinductively defined set $\nu\Phi$ and its axioms

Coclosure $\nu\Phi \subseteq \Phi(\nu\Phi)$

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

are realised by the terminal coalgebra $(\nu\varphi, \mathrm{Out}_\varphi)$
and the family $\mathrm{Coit}_\varphi[s]$ of universal arrows

$$\begin{array}{rcl} \mathrm{Out}_\varphi & : & \nu\varphi \to \varphi(\nu\varphi) \\ \mathrm{Coit}_\varphi[s] & : & \alpha \to \nu\varphi \quad (s : \alpha \to \varphi(\alpha)) \end{array}$$

with the equation expressing that $\mathrm{Coit}_\varphi[s]$ is a coalgebra morphism

$$\mathrm{Out}_\varphi \circ \mathrm{Coit}_\varphi[s] = \mathbf{map}_\varphi(\mathrm{Coit}_\varphi[s]) \circ s$$

Hence $\mathrm{Coit}_\varphi[s]$ is the function defined by *guarded recursion* with "state transition" function s.

Iterated maps

The family of logistic maps (transformed from $[0,1]$ to $\mathbb{I} = [-1,1]$):

$$f_a : \mathbb{I} \to \mathbb{I}, \quad f(x) = a * (1 - x^2) - 1 \qquad (0 \leq a \leq 2).$$

f_a is $2a$-Lipschitz, hence uniformly continuous (Lipschitz Lemma), hence in $\mathrm{C} := \mathrm{C}_{\mathrm{AV},\mathrm{AV}} \subseteq \mathbb{I} \to \mathbb{I}$ (Characterisation Lemma (a)).

Hence the iterated maps f_a^n are in C (Composition Lemma) and therefore define signed digit stream transformers (Application Lemma).

The main point of this example is to demonstrate the **memoizing effect** of the tree representation of u.c. functions (see also Hinze, Altenkirch).

$f_2(x) = 2 * (1 - x^2) - 1$. Computing $f_2^n(1/3)$

```
n = 1
Exact SD:         PPZZPZZNZZPZZNZZPZZNZZPZZNZZPZZNZZPZZNZZPZZNZZPZZNZZP
                  ZZNZZPZ
... as Float:     0.7777777777777778
Float:            0.7777777777777778
E. R. as Float:   0.7777777777777778
Exact Rat:        7%9


n = 5
Exact SD:         ZZPZZZZNZNPNPZZPZZPNZZPNNPNPNZPZZZZPZZPZZNPNZPZNPZZP
                  NZZNPZZN
... as Float:     0.12049965049830123
Float:            0.120499650498301
E. R. as Float:   0.12049965049830123
Exact Rat:        223288285122943%1853020188851841
```

$f_2(x) = 2 * (1 - x^2) - 1$. Computing $f_2^n(1/3)$

```
n = 10
Exact SD:         PPZZZZZZZNZPZPNZPNZZNPPNPZNZNPZNPNZZZPZZNZZZZZZNPZZZ
                  PZZZZZZP
... as Float:     0.7493017528354341
Float:            0.7493017528354383
E. R. as Float:   0.7493017528354341
Exact Rat:        2797831667561095955203291549538860747228990633859021
77740551732553042997330746919549126577046636631596857447826125531004
55275469619973310064456547818090396304560929400688366970738212885462
64844417999225202069134402085116597175924067307663489082904387928038
65580294776553932679091174750985548564347963457895727062471618250343
99787779443351588523431266046450103423936416024727327712311056145767
80329653408043140886531065840850302727959404399911873974992591272020
98801908525573383524124995583%37339184874102004353295975418486658822
54097767837340077506369317220790406172652512299936889388039772204687
65065431475158108727054592160858581351336982809187314191748594262580
93880701995195640428557181804104668128879740292551766801234061729839
65747316191523867230462351259348960585905882846547935405059362023765
47807442730582144527058988756251452817793413352141920744623027518729
18543286237573706398548531947641692626381997288700690701389925652429
```

$f_2(x) = 2 * (1 - x^2) - 1$. Computing $f_2^n(1/3)$

```
n = 30
Exact SD:       PZZZZZPNPZPNPPNPNPZZZZZNZPNZZZZZZPNPZPNPNPNPZZNPZNZP
                ZNZPZZZN
... as Float:   0.5062674954994535
Float:          0.5062674897488822


n = 60
Exact SD:       NNNZPNPZZNZNPZPNZPNZZPZZNZZNPZZPNPZPNZNZPZZNZNZZNZZP
                NZNZZZZN
... as Float:   -0.8526437597407311
Float:          -0.9469915748606237


n = 600
Exact SD:       NNZZZNPZZNZZZZZNPNZPZZPNPPNPNZZNPZZZPNZZZNZPNPZNPZNZ
                ZZZNPZNZ
... as Float:   -0.7587994543102519
Float:          0.33218504745590244
```

π

For the metric digit space $(\mathbb{I}, \mathrm{AV})$ we have $\pi/4 \in \mathrm{C}_D$.

Proof We use the formula

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

i.e. $\pi/4 = f_0(f_1(\dots))$ 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 F is contracting and AV is invertible and covering, it follows, by the Change of Digits L., $\pi/4 \in \mathrm{C}_D$.

Integration

For a continuous function $f : \mathbb{I} \to \mathbb{R}$ we set
$\int f := \int_{-1}^{1} f = \int_{-1}^{1} f(t)\,\mathrm{d}t \in \mathbb{R}$.

Lemma

(a) $\int(\mathrm{av}_i \circ f) = \mathrm{av}_{2\cdot i}(\int f)$

(b) $\int f = \frac{1}{2}(\int(f \circ \mathrm{av}_{-1}) + \int(f \circ \mathrm{av}_1))$.

(see also Simpson, Scriven)

Integration Lemma

Let (X, σ, P, D) be a covering and invertible metric digit system and $f \in \mathrm{C}_{D \otimes \mathrm{AV}, \mathrm{AV}}$. Then the function mapping $(a, b, x) \in \mathbb{I}^2 \times X$ to $\int_a^b f(x, t)\,\mathrm{d}t$ is well-defined and uniformly continuous.

Conclusion

- "Proofs as programs" deserves a "last chance".
- 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.

Further work

- Clarify connections with related work by Edalat, Heckmann, Potts, Escardo, Simpson, Bauer, Taylor, Hutchinson, Scriven, Hancock, Pattinson, Buchholz, Bertot, Niqui, O'Connor, Spitters, ...
- Implement and automate (joint work with the Munich logic group (Minlog) and Anton Setzer (Agda)).
- Overcome limitations: no finite system of contracting and uniformly covering digits exists on the compact metric space of non-empty compact sets with the Hausdorff metric (joint work with Dieter Spreen).
- Power series via higher type digits.
- "Realiser sensitive" logic:
  $\forall x\,(A(x) \to B(x) \to C(x))$ vs.
  $\forall x\,(A(x) \to \neg B(x) \lor C(x))$ for decidable $B(x)$.

References

T. Altenkirch.
Representations of first order function types as terminal coalgebras. TLCA 2001. LNCS 2044, 8–21, 2001.

A. Abel.
Syntactical normalization for intersection types with term rewriting rules. In *Fourth International Workshop on Higher-Order Rewriting, HOR'07, Paris, France, 25 June 2007*, 2007.

B.
Strong normalization for applied lambda calculi. *Logical Methods in Computer Science*, 1(2):1–14, 2005.

T. Hou, B.
Coinduction for Exact Real Number Computation. *Theory of Computing Systems*, 2007.

References

Y. Bertot.
Coinduction in Coq. In Lecture Notes of TYPES Summer School 2005, August 15-26 2005, Sweden, vol. II (2005).

W. Buchholz.
A term calculus for (co-)recursive definitions on streamlike data-structures. Annals of Pure and Applied Logic, Volume **136** (2005) 75–90.

A. Ciaffaglione, P. Di Gianantonio, Di P.
A certified, corecursive implementation of exact real numbers. Theoretical Computer Science, Volume **351** (2006) 39–51.

References

- T. Coquand, A. Spiwack.
  Proof of strong normalisation using domain theory. *Lics 2006*.

- A. Edalat, R. Heckmann.
  Computing with real numbers - I. The LFT approach to real number computation - II. A domain framework for computational geometry. In: Barthe G, Dybjer P, Pinto L, Saraiva J, editors, International summer school on applied semantics, Caminha, Portugal, Berlin, Springer-Verlag (2002) 193–267.

- M. Escardo, A. Simpson, A.
  A universal characterization of the closed Euclidean interval (extended abstract). Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science, Boston, Massachusetts (2001) 115–125.

References

P. Hancock, D. Pattinson,N. Ghani.
Representation of stream processors using nested fixed points. *unpublished*. 2008.

R. Hinze.
Memo functions, polytypically!". Proceedings of the Second Workshop on Generic Programming, WGP 2000, Ponte de Lima, Portugal. 2000.

B. Jacobs, J. Rutten.
A Tutorial on (Co)Algebras and (Co)Induction. EATCS Bulletin 62, 222–259, 1997.

References

R. O'Connor.
Certified Exact Transcendental Real Number Computation in Coq. Unpublished. 2008

R. O'Connor, B. Spitters.
A computer verified monadic, functional implementation of the integral. Unpublished. 2008

M. Niqui.
Formalising exact arithmetic in type theory. In S. B. Cooper, B. Lowe, and L. Torenvliet, editors, CiE 2005: New Computational Paradigms. Amsterdam, The netherlands. June 8 – 12, 2005. LNCS **3526** (2005) 368–377.

References

D. Plume.
A Calculator for Exact Real Number Computation. 4th year project. Departments of Computer Science and Artificial Intelligence, University of Edinburgh (1998).

G. D. Plotkin.
LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

M. Tatsuta.
Realizability of Monotone Coinductive Definitions and Its Application to Program Synthesis. *Proceedings of Fourth International Conference on Mathematics of Program Construction*, LNCS 1422 (1998) 338–364.