Numbers as streams of digits

Christiane Frougny
LIAFA, Paris

Representing streams
Lorentz Center
Leiden
10–14 December 2012

We do not bathe twice in the same stream. Heraclitus

Il n’y a que les mots qui comptent, le reste n’est que bavardage.
Only words matter, the rest is just a stream of idle chatter.
Eugène Ionesco

Representation of numbers

Elements of a subset of $\mathbb{C}$ represented by streams of digits

- Positional numeration systems
  - base β in $\mathbb{C}$, $|\beta| > 1$
  - basis $U = (u_n)_{n \geqslant 0}$
- Continued fractions
- Residue number system
- Logarithmic number system
- Abstract numeration systems
- ...

Representations can be finite, or right infinite, or left infinite strings (words) of digits.

Positional numeration systems

Basis $U = (u_n)_{n \geqslant 0}$, $u_n \in \mathbb{C}$

A U-representation of $x \in \mathbb{C}$ on a set D of complex digits can be

- a finite word $d_k \cdots d_0$, with $d_i \in D$, such that $x = \sum_{i=0}^{k} d_i u_i$
- or a right infinite word $d_1 d_2 \cdots$, with $d_i \in D$, such that $x = \sum_{i=1}^{\infty} d_i u_i$
- or a left infinite word $\cdots d_1 d_0$, with $d_i \in D$, $x = \sum_{i=0}^{\infty} d_i u_i$

Most significant digit on the left side.

Representability

Given a base or a basis, an algorithm $\mathcal{A}$ and a set $S \subset \mathbb{C}$:

- Does any element of S has an expansion by algorithm $\mathcal{A}$?
- What is the canonical digit set produced by algorithm $\mathcal{A}$?
- What is the set $\mathcal{A}(S)$ of expansions by $\mathcal{A}$ of the elements of S?

Representability

Given a base or a basis, an algorithm $\mathcal{A}$ and a set $S \subset \mathbb{C}$:

- Does any element of S has an expansion by algorithm $\mathcal{A}$?
- What is the canonical digit set produced by algorithm $\mathcal{A}$?
- What is the set $\mathcal{A}(S)$ of expansions by $\mathcal{A}$ of the elements of S?

Algorithm $\mathcal{A}$:

- greedy algorithm $\mathcal{G}$, produces most significant digit first
- modified Euclidean Division algorithm $\mathcal{D}$, produces least significant digit first.

$U = (u_n)_{n \geqslant 0}$ with $u_n = \beta^n$.

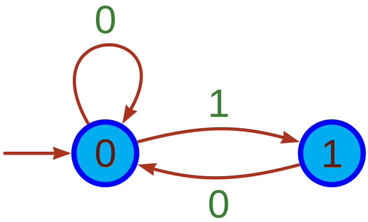
Integer base $\beta > 1$ in $\mathbb{N}$

- By algorithms $\mathcal{G}$ and $\mathcal{D}$, every element of $\mathbb{N}$ has a unique finite expansion $d_k \cdots d_0$, with $d_k \neq 0$, d_i in the canonical digit set $A = \{0, 1, \ldots, \beta - 1\}$, $0 \leqslant i \leqslant k$.
$\mathcal{G}(\mathbb{N}) = \mathcal{D}(\mathbb{N}) = (A \setminus \{0\})A^*$.
- By algorithm $\mathcal{G}$ every element of $[0, 1]$ has a right infinite expansion $(d_i)_{i \geqslant 1}$, $d_i \in A$ (and thus every element of $\mathbb{R}_+$).
$\mathcal{G}([0, 1]) = A^{\mathbb{N}} \setminus A^*(\beta - 1)^\omega$.
- Every element of $\mathbb{Q}_+$ has an eventually periodic right infinite greedy expansion on A.
- By algorithm $\mathcal{D}$ every element of $\mathbb{Q}$ has an eventually periodic left infinite expansion on A (p-adic expansion).

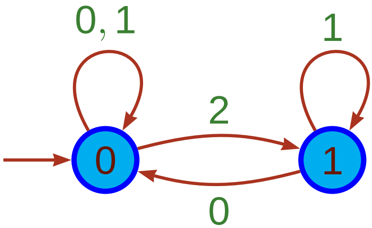
Real base $\beta > 1$

- By algorithm $\mathcal{G}$ every element of $[0, 1]$ has a right infinite expansion $(d_i)_{i \geqslant 1}$, d_i in the canonical digit set $A = \{0, 1, \ldots, \lceil \beta \rceil - 1\}$ (and thus every element of $\mathbb{R}_+$).
- Not every infinite word on A is admissible: $\mathcal{G}([0, 1]) \subsetneq A^{\mathbb{N}}$. When β is a Pisot number, $\mathcal{G}([0, 1])$ with the shift forms a sofic dynamical system, i.e., the set of finite factors of $\mathcal{G}([0, 1])$ is recognizable by a finite automaton (Bertrand).
- When β is a Pisot number, every element of $\mathbb{Q}(\beta) \cap \mathbb{R}_+$ has an eventually periodic expansion (Boyd).
- For some Pisot numbers, for instance the golden mean, every element of $\mathbb{Z}(\beta) \cap \mathbb{R}_+$ has a finite expansion.

Example The golden mean shift: system of finite type. Local automaton.



Example The β-shift for $\beta = \frac{3+\sqrt{5}}{2}$: sofic system not of finite type. Non-local automaton.



Basis $U = (u_n)_{n \geqslant 0}$ with $u_0 = 1$, $u_n \in \mathbb{N}$, strictly increasing: every positive integer has a finite U-expansion by a greedy algorithm (Fraenkel).

Example U is the sequence of Fibonacci numbers with digit set $\{0, 1\}$.

Basis $U = (u_n)_{n \geqslant 0}$ with $u_0 = 1$, $u_n \in \mathbb{N}$, strictly increasing: every positive integer has a finite U-expansion by a greedy algorithm (Fraenkel).

Example U is the sequence of Fibonacci numbers with digit set $\{0, 1\}$.

Example Factorial numeration system: $u_n = n!$, digit set $= \mathbb{N}$.

Basis $U = (u_n)_{n \geqslant 0}$ with $u_0 = 1$, $u_n \in \mathbb{N}$, strictly increasing: every positive integer has a finite U-expansion by a greedy algorithm (Fraenkel).
Example U is the sequence of Fibonacci numbers with digit set $\{0, 1\}$.
Example Factorial numeration system: $u_n = n!$, digit set $= \mathbb{N}$.

Basis $U = (u_n)_{n \geqslant 0}$ with $u_n \in \mathbb{R}_+$, summable and strictly decreasing (Muller), by a greedy algorithm.
Example $u_n = \log(1 + 2^{-n})$, $A = \{0, 1\}$: representation of positive reals by an infinite word on A.

Basis $U = (u_n)_{n \geqslant 0}$ with $u_0 = 1$, $u_n \in \mathbb{N}$, strictly increasing: every positive integer has a finite U-expansion by a greedy algorithm (Fraenkel).
Example U is the sequence of Fibonacci numbers with digit set $\{0, 1\}$.
Example Factorial numeration system: $u_n = n!$, digit set $= \mathbb{N}$.

Basis $U = (u_n)_{n \geqslant 0}$ with $u_n \in \mathbb{R}_+$, summable and strictly decreasing (Muller), by a greedy algorithm.
Example $u_n = \log(1 + 2^{-n})$, $A = \{0, 1\}$: representation of positive reals by an infinite word on A.

Negative base, complex base...

Rational base $\frac{p}{q} > 1$

- By $\mathcal{G}$ every positive real has an aperiodic expansion on $\{0, 1, \ldots, \lceil \frac{p}{q} \rceil - 1\}$.
- By $\mathcal{D}$ every positive integer has a unique finite expansion on $\{0, 1, \ldots, p - 1\}$ (Akiyama, Frougny, Sakarovitch). The set of expansions of elements of $\mathbb{N}$ is not context-free.
- By $\mathcal{D}$ every element of $\mathbb{Q}$ has an eventually periodic left infinite expansion.

Distance on the set of infinite words

Prefix distance on $A^{\mathbb{N}}$:

$$\rho(v, w) = \begin{cases} 2^{-r} & \text{where } r = \min\{i \mid v_i \neq w_i\} \\ 0 & \text{if } v = w \end{cases}$$

$A^{\mathbb{N}}$ is metric compact.

Distance on the set of infinite words

Prefix distance on $A^{\mathbb{N}}$:

$$\rho(v, w) = \begin{cases} 2^{-r} & \text{where } r = \min\{i \mid v_i \neq w_i\} \\ 0 & \text{if } v = w \end{cases}$$

$A^{\mathbb{N}}$ is metric compact.

Addition on $\mathbb{R}$ is continuous, but, addition of expansions base 2 and digit set $\{0, 1\}$ is not continuous for the prefix distance.

$0(1)^k 0^\omega + 0^\omega = 0(1)^k 0^\omega$
$0(1)^k 0^\omega + (0)^k 10^\omega = 10^\omega$

On-line functions

A function $\varphi : A^{\mathbb{N}} \to B^{\mathbb{N}}$ is on-line computable with delay δ if $\exists \delta \in \mathbb{N}$ such that $(b_n)_{n\geqslant 1} = \varphi((a_n)_{n\geqslant 1})$ iff $\forall n \geqslant 1$ there exists $\Phi_n : A^{n+\delta} \to B$ with $b_n = \Phi_n(a_1 \cdots a_{n+\delta})$.

The digit at instant n depends only on the past, and not on the future. On-line arithmetic allows the pipelining of different operations such as addition, multiplication and division, because the processing is Most Significant Digit First. Well adapted to real numbers.

On-line denumerable transducer:

$$\underbrace{q_0 \xrightarrow{a_1|\varepsilon} q_1 \xrightarrow{a_2|\varepsilon} q_2 \cdots \xrightarrow{a_\delta|\varepsilon} q_\delta}_{\text{transient part}} \underbrace{\xrightarrow{a_{\delta+1}|b_1} q_{\delta+1} \xrightarrow{a_{\delta+2}|b_2} q_{\delta+2} \cdots}_{\text{synchronous part}}$$

Proposition

A function on-line computable with delay δ is 2^δ-Lipschitz, and thus uniformly continuous.

Numerical value in base $\beta > 1$:
$\pi_\beta : A^{\mathbb{N}} \to \mathbb{R}$ with $\pi_\beta((a_n)_{n\geqslant 1}) = \sum_{n\geqslant 1} a_n \beta^{-n}$.

If the following diagram commutes, $\varphi_{\mathbb{R}}$ is the real realization of φ in base β

$$\begin{array}{ccc} A^{\mathbb{N}} & \xrightarrow{\varphi} & B^{\mathbb{N}} \\ \downarrow \pi_\beta & & \downarrow \pi_\beta \\ \pi_\beta(A^{\mathbb{N}}) & \xrightarrow{\varphi_{\mathbb{R}}} & \pi_\beta(B^{\mathbb{N}}) \end{array}$$

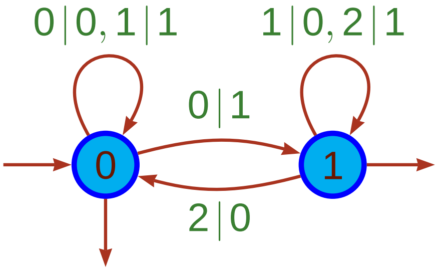
Proposition (Eilenberg)

If φ is continuous than $\varphi_{\mathbb{R}}$ is continuous.

Finite transducers

A function $\varphi : A^{\mathbb{N}} \to B^{\mathbb{N}}$ is computable by a transducer $\mathcal{T} = (Q, A^* \times B^*, E, I, F)$ if the graph of φ is the set of labels of infinite paths starting in I and going infinitely often in F.
$\mathcal{T}$ is finite if E and Q are finite.

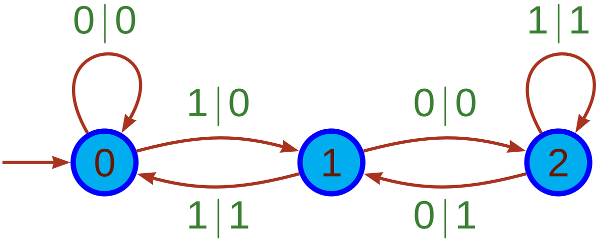
Example Addition of reals base 2 and digit set $\{0, 1\}$ = conversion from $\{0, 1, 2\}$ to $\{0, 1\}$



Sequentiality

$\mathcal{T} = (Q, A^* \times B^*, E, I, F)$ is sequential if $|I| = 1$, $F = Q$, and it is input deterministic.
Processing from left to right: left sequential.

Example Division by 3 in base 2 and digit set $\{0, 1\}$.



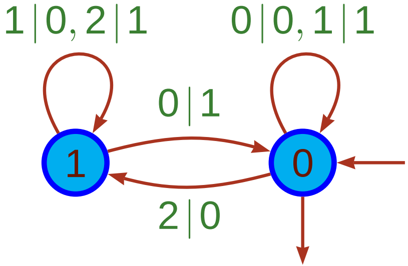
Proposition

A function computable by a finite sequential transducer is uniformly continuous.

Right sequentiality

Processing from right to left

Example Addition in base 2 and digit set $\{0, 1\}$.



Theorem

Any function computable by a finite transducer can be obtained by the composition of a finite right sequential transducer and a finite left sequential transducer.

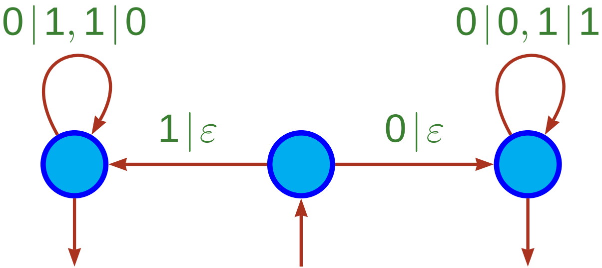
Finite words: Elgot and Mezei
Infinite words: Carton

On-line finite transducer

Particular left sequential finite transducer.

Example Tent function in base 2 and digit set $\{0, 1\}$.

$$f(x) = \begin{cases} 2x & \text{if } 0 \leqslant x \leqslant 1/2 \\ -2x + 2 & \text{if } 1/2 \leqslant x \leqslant 1 \end{cases}$$



$$\frac{3}{4} = .110^{\omega} \mapsto \frac{1}{2} = .01^{\omega}$$

$$\frac{3}{4} = .101^{\omega} \mapsto \frac{1}{2} = .10^{\omega}$$

In positive integer base, addition, multiplication by a fixed integer, division by a fixed integer are computable by an on-line finite transducer.

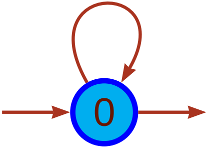
Theorem (Muller)

The real realization of a function computable by an on-line finite transducer (in integer positive base) is a piecewise affine function whose coefficients are rational numbers.

Conversion base 4 $\rightarrow$ base 2 is

1. left sequential and right sequential

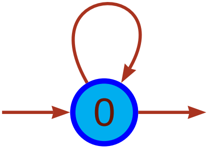
$$0\,|\,00,\ 1\,|\,01,\ 2\,|\,10,\ 3\,|\,11$$



Conversion base 4 $\to$ base 2 is

1. left sequential and right sequential

$$0\,|\,00,\ 1\,|\,01,\ 2\,|\,10,\ 3\,|\,11$$



2. on-line computable with delay 0

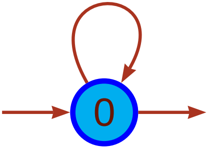
   $a_n \in \{0,\ldots,3\}$, $a_n = a_n^{(1)} a_n^{(2)}$, $a_n^{(1)}, a_n^{(2)}, b_n \in \{0,1\}$,
   $\varphi((a_n)) = b_n$ with $b_{2n-1} = a_n^{(1)}$ and $b_{2n} = a_n^{(2)}$

   Uses a queue: $\varepsilon \xrightarrow{2|1} 0 \xrightarrow{3|0} 11 \xrightarrow{0|1} 100 \xrightarrow{1|1} 0001 \cdots$

Conversion base 4 $\to$ base 2 is

1. left sequential and right sequential

$$0 \mid 00,\ 1 \mid 01,\ 2 \mid 10,\ 3 \mid 11$$



2. on-line computable with delay 0

   $a_n \in \{0, \dots, 3\}$, $a_n = a_n^{(1)} a_n^{(2)}$, $a_n^{(1)}, a_n^{(2)}, b_n \in \{0, 1\}$,
   $\varphi((a_n)) = b_n$ with $b_{2n-1} = a_n^{(1)}$ and $b_{2n} = a_n^{(2)}$

   Uses a queue: $\varepsilon \xrightarrow{2|1} 0 \xrightarrow{3|0} 11 \xrightarrow{0|1} 100 \xrightarrow{1|1} 0001 \cdots$

3. not computable by a finite on-line transducer

Local functions

$\varphi : A^{\mathbb{Z}} \to B^{\mathbb{Z}}$ is a p-local function if $\exists r, t > 0$, and $\exists \Phi : A^p \to B$, with $p = r + t + 1$, such that

$$(b_n)_{n\in\mathbb{Z}} = \varphi((a_n)_{n\in\mathbb{Z}}) \iff \forall n \in \mathbb{Z},\ \ b_n = \Phi(a_{n+t} \cdots a_{n-r}).$$

The image of $(a_n)_{n\in\mathbb{Z}}$ by φ is obtained through a sliding window of length p.
r is the memory and t is the anticipation of φ.
φ is called a sliding block code.

Local functions

$\varphi : A^{\mathbb{Z}} \to B^{\mathbb{Z}}$ is a p-local function if $\exists r, t > 0$, and $\exists \Phi : A^p \to B$, with $p = r + t + 1$, such that

$$(b_n)_{n\in\mathbb{Z}} = \varphi((a_n)_{n\in\mathbb{Z}}) \iff \forall n \in \mathbb{Z}, \;\; b_n = \Phi(a_{n+t} \cdots a_{n-r}).$$

The image of $(a_n)_{n\in\mathbb{Z}}$ by φ is obtained through a sliding window of length p.
r is the memory and t is the anticipation of φ.
φ is called a sliding block code.
Locality ensures robustness: no propagation of errors.

A local function on finite words is computable by a parallel algorithm.

Proposition

A p-local function is computable by a finite on-line transducer with delay $p - 1$. The input automaton is local.

Signed-digit representations

Base 10 and digit-set $\{-5, \ldots, 0, \ldots, 5\}$ Cauchy 1840

Base 10 and digit-set $\{-6, \ldots, 0, \ldots, 6\}$ Avizienis 1961

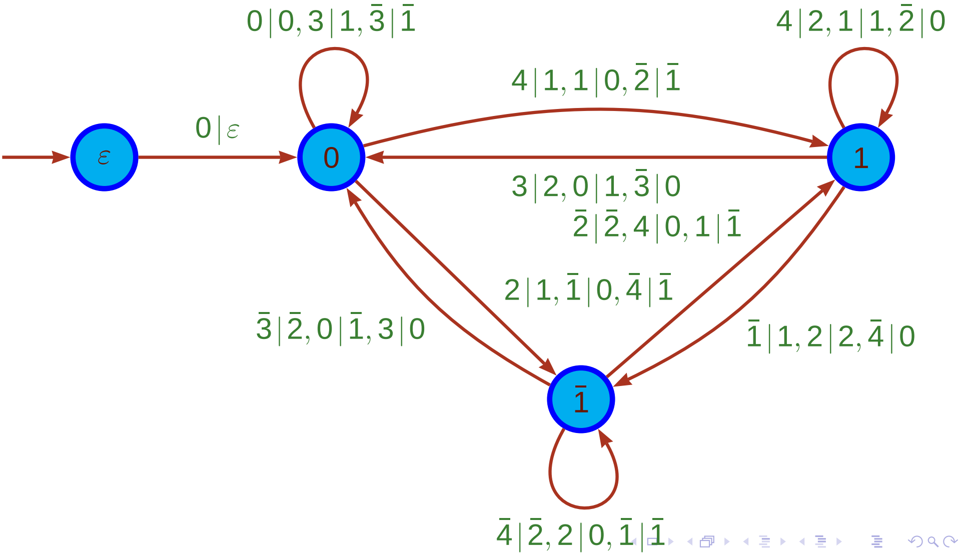
Base 2 and digit-set $\{-1, 0, 1\}$ Chow and Robertson 1978

In integer base b, $b \geqslant 3$, parallel addition on alphabet $\{-a, \ldots, 0, \ldots, a\}$, $b/2 < a \leqslant b - 1$ is possible by Avizienis algorithm. It is a 2-local function.

In integer base $b = 2a$, $b \geqslant 2$, parallel addition on alphabet $\{-a, \ldots, 0, \ldots, a\}$ is possible by Chow and Robertson algorithm. It is a 3-local function.

Redundancy
No propagation of the carry.

On-line finite transducer with delay 1 realizing addition in base 3 on $\{\bar{2}, \ldots, 2\}$: $p \xrightarrow{x|y} q \Leftrightarrow 3p + x = 3y + q$



Parallel addition

Theorem (Frougny, Pelantová, Svobodová)

Let $\beta \in \mathbb{C}$ with $|\beta| > 1$ be an algebraic number. If all its algebraic conjugates have modulus $\neq 1$ one can find an alphabet of contiguous integer digits on which addition can be done in parallel.

Redundancy is necessary.
We have some lower bounds on the minimality of the cardinality of the digit set.
The result is not necessarily admissible.

Example Addition in base the golden mean:

- on the minimal alphabet $\{-1, 0, 1\}$ is a 21-local function
- on $\{-3, \dots, 3\}$ is 13-local
- on $\{-5, \dots, 5\}$ is 9-local.

On-line addition

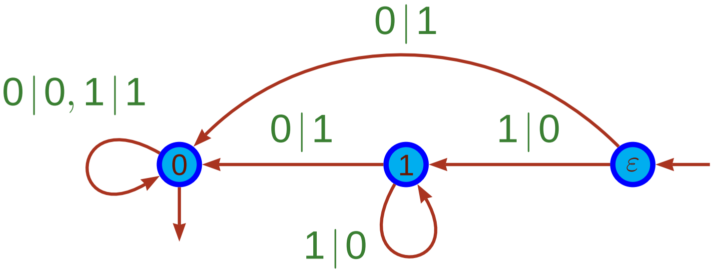
Suitable for real numbers.

- In real base $\pm\beta$, $\beta > 1$, addition is on-line computable on $\{0, \ldots, \lfloor\beta\rfloor\}$ (the result is not admissible).
- If β is a Pisot number, the on-line transducer is finite.
- To get an admissible result, normalization is necessary: If β is a Pisot number, normalization is computable by a finite transducer, which is neither left nor right sequential.

Successor function = addition of 1

In integer base the successor function $\langle n \rangle \mapsto \langle n+1 \rangle$ is realizable by a right sequential letter-to letter finite transducer.

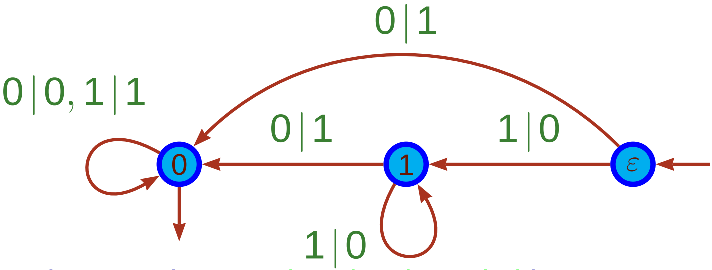
Successor function base 2:



Successor function = addition of 1

In integer base the successor function $\langle n \rangle \mapsto \langle n+1 \rangle$ is realizable by a right sequential letter-to letter finite transducer.

Successor function base 2:



Theorem (Angrand and Sakarovitch)

Let L be a language ordered by the radix order and recognizable by a finite automaton. The successor function on L is realizable by a finite union of right sequential finite transducers with disjoints domains.

Application: L is the set of expansions of $\mathbb{N}$ in a given numeration system.

Multiplication

- Multiplication is not computable by a finite transducer
- Multiplication is on-line computable
  - in positive integer base $b \geqslant 2$ on $\{-a, \ldots, 0, \ldots, a\}$, $b/2 \leqslant a \leqslant b-1$ (Ercegovac and Trivedi)
  - in negative integer base $(-b)$ on $\{-a, \ldots, 0, \ldots, a\}$, $b/2 \leqslant a \leqslant b-1$
  - in real base $\beta > 1$ on $\{0, \ldots, \lfloor \beta \rfloor\}$. (The result is not admissible)
  - in the Knuth number system of base $i\sqrt{b}$, $b \geqslant 2$ integer, on $\{-a, \ldots, 0, \ldots, a\}$, $b/2 \leqslant a \leqslant b-1$
  - in the Penney numeration system of base $-1+i$ on $\{-1, 0, 1\}$ (Surarerks)

Redundancy is necessary

Concluding remarks

Numbers can be seen as streams or flows of digits, either one by one most significant digit first, or looked at through a sliding window.

For application to algorithms, sequentiality and synchronicity are important, as well as finite memory when it is possible.

- On-line functions: most significant digit first, well adapted to real numbers with infinite expansions. Pipelining with addition, multiplication and division...
- Local functions: sliding window, parallel algorithms, adapted to arithmetical circuits. Internal additions in on-line algorithm.

Both need redundancy.

Some questions:

- Find a “good" model for multiplication
- What are the functions computable by an on-line transducer with queue memory?
- Compromise between the size of the digit set and the size of the window for local functions?