On the redundancy of real number representation systems

Christophe Mazenc

Laboratoire de l'Informatique du Parallélisme

Ecole Normale Supérieure de Lyon
Institut IMAG
Unité de recherche associée au CNRS n°1398

On the redundancy of real number representation systems

Christophe Mazenc

May 1993

Research Report N$^{\mathrm{O}}$ 93-16

Ecole Normale Supérieure de Lyon
46, Allée d'Italie, 69364 Lyon Cedex 07, France,
Téléphone : + 33 72 72 80 00; Télécopieur : + 33 72 72 80 80;
Adresses électroniques :
lip@frensl61.bitnet; lip@lip.ens−lyon.fr (uucp).

On the redundancy of real number representation systems

Christophe Mazenc

May 1993

Abstract

In this paper, a set of definitions describing general real number representation systems is presented. Our purpose is to find a sufficiently wide model definition including classical systems (signed-digit notation, linear numeration systems, p-adic numbers, symmetric level index arithmetic...) but specific enough to make it possible to build some practical results. We focuse on the redundancy property and the relationships between redundancy, on-line and digit-parallel calculus.

Keywords: Number representation systems, redundancy, on-line arithmetic

Résumé

Dans ce papier, nous présentons un ensemble de définitions décrivant de manière générale tout système de représentation des nombres. Notre but est de définir un modèle suffisamment général pour inclure les systèmes classiques et suffisamment spécifique pour rendre possible la synthèse de propriétés générales. Nous nous intéressons plus particulièrement à la propriété de redondance et explicitons ses liens avec les modes de calcul en ligne et parallèle au niveau du chiffre.

Mots-clés: Systèmes de représentation des nombres, redondance, arithmétique en ligne

On the redundancy of real number representation systems [†]

Christophe Mazenc
LIP
Ecole Normale Supérieure de Lyon
46 Allée d'Italie
69364 Lyon Cedex 07, France

Abstract

A lot of people whose research field has a non empty intersection with computer arithmetic take for granted many basic assumptions such as : "if we want to perform serial addition most significant digits first, we have to use a redundant number system" or "if the system is not redundant, one can not build a fully digit parallel adder". This work gives a fully specified background defining properly what is a redundant number system and gives tools to help people prove what they have ever thought without having it proved yet.

1 Introduction

In this paper, a set of definitions describing general real number representation systems is presented. Our purpose similar to Wiedmer's (see [13]) is to find a sufficiently wide definition including classical systems (signed-digit notation, linear numeration systems, p-adic numbers, symmetric level index arithmetic...) but specific enough to make it possible to build some practical results. We focuse on the redundancy property and the relationships between redundancy, on-line and digit-parallel calculus. The most famous redundant system was introduced by Avizienis using a signed-digit representation (see [2]) and led to the definition of on-line arithmetic. On-line operators compute their result digit by digit, producing a new output digit each time they have received a new digit of every operand, most significant digit first. The operator computes the first output digit after having received the first $\delta + 1$ digits of every operand: δ is called the *delay* of the on-line operator (For more details, the reader can refer to [3, 6]).

2 Getting into the subject

Definition 1 *A system representing a subset of the real numbers is given by a couple (C, F) where C is a* finite alphabet *and F a partially defined mapping: $F : C^{\mathbb{N}} \longrightarrow \mathbb{R}$ ($\mathbb{N}$ is the set of the natural integers and $\mathbb{R}$ the set of the real numbers) such that the image of F is dense in a neighborhood of zero in $\mathbb{R}$. The operators +, -, * working on strings of $C^{\mathbb{N}}$ are defined with respect to the following condition: $F(X \text{ op } Y) = F(X) \text{ op } F(Y)$ each time $F(X) \text{ op } F(Y)$ belongs to $F(C^{\mathbb{N}})$. Letters of the alphabet C are denoted by $c^0, c^1, \ldots$*

[†] This work has been partially supported by the french "réseau doctoral en architecture des ordinateurs".

Definition 2 *For any integer n we denote by $R_n(a_0 a_1 ... a_n)$ the set of real numbers that can be represented by a string of $C^{\mathbb{N}}$ beginning with the $n+1$ letters $a_0, a_1, ..., a_n$. In the following, we focuse on systems where :*

(H1) For all natural integers n, for all infinite string $(a_i)_{i \in \mathbb{N}}$ $R_n(a_0 ... a_n)$ is bounded.

(H2) $\lim_{n \to +\infty} (Sup(R_n(a_0 ... a_n)) - Inf((R_n(a_0 ... a_n)) = 0$.

We call them converging systems. These two restrictions are quite natural since the user expects to get more information on the location of a real number by getting more letters (digits) of one of its representations. It also implies that only a bounded set of reals can be represented. As opposed to converging systems, one can define diverging systems where the $R_n(a_0 ... a_n)$ are not bounded. P_adic number systems are diverging systems (see [8, 1]).

Lemma 1 *In a converging real number system, for all natural integer n, and for all infinite string $(a_i)_{i \in \mathbb{N}}$, $R_n(a_0 ... a_n)$ is a closed set in the usual topology of real numbers.*
In particular: $Sup(R_n(a_0 ... a_n)) = Max(R_n(a_0 ... a_n))$ and will be noted $Max(a_0 ... a_n)$.

Proof. We prove that for any sequence $(x_m)_{m \in \mathbb{N}}$ of $R_n(a_0 ... a_n)$ that goes to the real x when m goes to infinity, the limit x belongs to $R_n(a_0, ..., a_n)$.
For each $m \in \mathbb{N}$ let us define $X_m \in C^{\mathbb{N}}$ so that:

- $X_m = (a_{m,0}, ..., a_{m,n}, ...)$ and $a_{m,0}, ..., a_{m,n} = a_0, ..., a_n$.
- $F(X_m) = x_m$.

Let us consider the set $A = \{a_{m,n+1} | m \in \mathbb{N}\}$. As A is included into C, the cardinal of A is finite and there exists $c^i \in C$ so that $card\{m \in \mathbb{N} | a_{m,n+1} = c^i\} = +\infty$. We set then $a_{n+1} = c^i$ and apply this process iteratively by incrementing n and by only taking into account the terms of the sequence that have $a_0, ..., a_n$ as n first letters.
This is an usual diagonal process that extracts from the sequence $(X_m)_{m \in \mathbb{N}}$ a new sequence $(Y_m)_{m \in \mathbb{N}}$.
The following equalities hold:

- $\forall m \in \mathbb{N}, F(Y_m) \in R_n(a_0 ... a_n)$
- $\lim_{m \to +\infty} F(Y_m) = x$
- $\forall m \in \mathbb{N}, \forall i \leq m, Y_{m,i} = a_i$

Let us call $Y = (a_0, ..., a_n, ...)$ and $y = F(Y)$ the associated real number. We prove in the following that y equals x. We immediatly deduce that $x \in R_n(a_0, ..., a_n)$.
For $n+1$ letters $c_0, ..., c_n$, let us call $diam(c_0, ..., c_n)$ the real number equal to

$$Sup(R_n(c_0 ... c_n)) - Inf(R_n(c_0 ... c_n)).$$

For any $\epsilon \in \mathbb{R}$ with $\epsilon > 0$, let us choose $p \in \mathbb{N}$ so that $\forall n \geq p \; diam(a_0, ..., a_n) \leq \frac{1}{2}\epsilon$. We obtain :

$$|F(Y) - F(Y_q)| \leq \frac{1}{2}\epsilon \tag{1}$$

Now let us choose $q \geq p$ so that:

$$|x - F(Y_q)| \leq \frac{1}{2}\epsilon \tag{2}$$

From (1) and (refequ2) we deduce that:

$$|x - F(Y)| = |x - y| \leq \epsilon$$

As this is true for any ϵ, x equals to y. □

Definition 3 *A number system (F,C) is fully redundant if and only if there exists a non negative integer k that verifies for each real number $A = (a_0, a_1, ...)$ and for each integer $n \geq k$:*

- *If $Sup(a_0...a_n) < Sup(F(C^{\mathbb{N}}))$ then there exists a string $(b_0, ..., b_n)$ different from $(a_0...a_n)$ ($\exists i$ with $0 \leq i \leq n$ and $b_i \neq a_i$) such that $Sup(a_0...a_n) > Inf(b_0...b_n)$ and $Sup(b_0...b_n) > Sup(a_0...a_n)$.*
- *If $Inf(a_0...a_n) > Inf(F(C^{\mathbb{N}}))$ then there exists a string $(c_0, ..., c_n)$ different from $(a_0...a_n)$ such that $Sup(c_0...c_n) > Inf(a_0...a_n)$ and $Inf(a_0...a_n) > Inf(c_0...c_n)$.*

This definition makes the link between redundancy and the overlapping of the representation sets: a real number system is redundant if and only if given a representation $(a_0...)$ of a real number, each prefix of this representation has got a suffix whose combination can be rewritten with a different prefix.

Lemma 2 *If in a real number system, one can add numbers with an online delay δ, then for all integers $n > 0$ there exists $(z_0 z_1...)$ a string representing zero so that $R_n(z_0 z_1...z_n) \neq \{0\}$.*

Proof. Let us suppose that there exists n so that for all string representing zero, $R_n(z_0...z_n) = \{0\}$ and reduce it to the absurd. Let us take x a real number represented by a string X, and $-X$ a string representing its opposite. Let us define X' having the same first $n + 1 + \delta$ digits as $-X$ and anything after. The first $n + 1$ digits of the on-line sum of X and X' must be the first $n + 1$ digits of a string representing zero. As $R_n(z_0...z_n) = \{0\}$ we obtain $X = X'$. As a result, our number system can at most represent only $(card(C))^{n+\delta+1}$ real numbers which is absurd because $F(C^{\mathbb{N}})$ includes a non empty neighborhood of zero. □

Lemma 3 *If in a real number system, one can add numbers with an on-line delay δ, then for all natural integer $n > 0$ there exists $Z = (z_0 z_1...)$ a string representing zero so that $R_n(z_0 z_1...z_n)$ includes a stricly positive real number and a string Z' so that $R_n(z'_0 z'_1...z'_n)$ includes a stricly negative real number.*

Proof. Let us take an integer $n > 0$, from Lemma 2 we produce an infinite string representing zero $Z = (z_0 z_1...)$ so that there exists $Z' \in R_{n+\delta}(z_0 z_1...z_{n+\delta}) \backslash \{0\}$. Let us call Z'' the opposite of Z' and Z''' the string resulting from the on-line addition of Z' and Z''. Let us consider $Z + Z''$, by construction its first $n+1$ digits are the first digits of a string representing zero (They are the same as those of Z''') and the sign of $Z + Z''$ is the opposite of the sign of Z'. □

Lemma 4 *If in a real number system, one can add numbers with an on-line delay δ, then for all integer $n > 0$ there exists a string $Z = (z_0 z_1...)$ representing zero so that $R_n(z_0 z_1...z_n)$ includes a strictly positive real number and a strictly negative number.*

Proof. Let us take an integer $n > 0$, from Lemma 3 we produce two infinite strings representing zero $Z = (z_0 z_1...)$ and $Z' = (z'_0 z'_1...)$ so that there exist $P \in R_{n+\delta}(z_0 z_1...z_{n+\delta})$ with $P > 0$ and $N \in R_{n+\delta}(z'_0 z'_1...z'_{n+\delta})$ with $N < 0$. The first $n + 1$ digits resulting from the on-line addition of $P + Z'$ and of $Z + N$ are the same and are the first $n + 1$ digits of an infinite string representing zero. We immediately see that $P + Z' = P > 0$ and $Z + N = N < 0$. □

Theorem 1 *If in a converging real number system, one can add numbers with an on-line delay δ then this system is fully redundant.*

Proof. Let us take a representable real number $A = (a_0 a_1 ...)$ and an integer $n > k$ such that $Max(a_0...a_n) < Max(F(C^{\mathbb{N}}))$ and $Min(a_0...a_n) > Min(F(C^{\mathbb{N}}))$. Thanks to Lemma 4, let us choose $Z' > 0$ and $Z'' < 0$ with Z' and $Z'' \in R_{n+\delta+i}(z_0...z_{n+\delta})$. As the system is converging, by sufficiently increasing the integer i, we can choose strings Z' and Z'' so that $Max(a_0...a_n) + Z'$ and $Min(a_0...a_n) + Z''$ be representable. Let us call $A_n = Max(a_0...a_n)$. Let us consider the infinite string $(b_0 b_1 ...)$ resulting from the addition of A_n and Z': this real number is strictly greater than A_n therefore the string $(a_0...a_n)$ is different from the string $(b_0...b_n)$. Moreover, when adding serially A_n and Z, we obtain the first $n+1$ digits $(b_0...b_n)$ too, so the infinite string $A_n + Z$ is another representation of A_n. The first $n+1$ digits resulting from the on-line addition of A_n and Z'' are $(b_0...b_n)$ and $A_n + Z'' < A_n$ then:

$$Max(a_0 a_1 ... a_n) > Min(b_0 b_1 ... b_n)$$

We also verify that:

$$Max(b_0 b_1 ... b_n) \geq A_n + Z' > A_n = Max(a_0 a_1 ... a_n)$$

We proceed symmetrically with $B_n = Min(a_0...a_n)$ and the theorem is proven. □

Definition 4 *An operator that performs the exact addition of two real numbers by deducing the i^{th} digit of the result from the values of at most p positions in the representation of the operands and satisfying the following constraint is called a parallel adder:*

Let us denote $Pos(i, A, B)$ the set of the positions in the representation of the operands A and B that have an influence on the i^{th} digit of the writing of the result of the addition of A and B. Then for all i, the set: $\{Pos(i, A, B) \mid A$ and B are representable.$\}$ is bounded by the natural number $f(i)$ ($f() \in \mathbb{N}^{\mathbb{N}}$). If in a number system there exists an online adder, there exists obviously a parallel adder too.

Lemma 5 *If in a real number system, a parallel adder can be built then for all natural integer $n > 0$ there exists a string $(z_0 z_1 ...)$ representing zero so that $R_n(z_0 z_1 ... z_n) \neq \{0\}$.*

Proof. Let suppose that there exists n so that for all string representing zero, $R_n(z_0...z_n) = \{0\}$ and reduce it to the absurd. Let us denote $M_n = Max\{\ Pos(i, A, B) \mid A$ and B be representable and $0 \leq i \leq n\ \}$. By construction, we have $M_n \leq Max\{\ f(i)$ with $0 \leq i \leq n\ \}$. Let us take a representation X of a real number and a representation $-X$ of the opposite. Let us build the infinite string X' obtained by concatenating the first M_n digits of X and anything after. The first $n+1$ digits of the parallel adding of X' and $-X$ are those of a representation of zero. As $R_n(z_0...z_n) = \{0\}$, we get $X' = X$. At most $(card(C))^{M_n}$ different real numbers can be represented in our system what is absurd. □

Lemma 6 *If in a real number system, a parallel adder can be built then for all natural integer $n > 0$ there exists $Z = (z_0 z_1 ...)$ a string representing zero so that $R_n(z_0 z_1 ... z_n)$ includes a stricly positive real number and a string Z' so that $R_n(z'_0 z'_1 ... z'_n)$ includes a stricly negative real number.*

Lemma 7 *If in a real number system, a parallel adder can be built then for all natural integer $n > 0$ there exists $Z = (z_0 z_1 ...)$ a string representing zero so that $R_n(z_0 z_1 ... z_n)$ includes a strictly positive real number and a strictly negative number.*

Proof. The proofs of Lemmas 6,7 are the same as those of of Lemmas 3,4 by replacing everywhere $n + \delta$ by M_n where:

$M_n = Max\{ \ Pos(i, A, B) \mid A$ and B be representable and $0 \leq i \leq n+1 \ \}$. □

Theorem 2 *If in a converging real number system one can build a parallel adder, then this system is fully redundant.*

Proof. Let us take a representable real number $A = (a_0 a_1 ...)$ and an integer $n > k$ so that $Max(a_0 ... a_n) < Max(F(C^{\mathbb{N}}))$ and $Min(a_0 ... a_n) > Min(F(C^{\mathbb{N}}))$. Thanks to Lemma 7, let us choose strings $Z' > 0$ and $Z'' < 0$ with Z' and $Z'' \in R_{M_n+i}(z_0 ... z_{M_n+i})$. As the system is converging, by sufficiently increasing the integer i, we can choose Z' and Z'' so that $Max(a_0 ... a_n) + Z'$ and $Min(a_0 ... a_n) + Z''$ be representable.
The end of the proof is identical to the end of the proof of Theorem 1. □

Theorem 3 *If in a converging real number system, one can multiply numbers with an on-line delay δ in a non-empty interval including 1, then this system is fully redundant.*

Proof. We proceed as for Theorem 1 by proving these successive lemma:

Lemma 8 *If in a number system, one can multiply numbers around one with an online delay δ, then for all integers $n > 0$ there exists a string representing one $(u_0 u_1 ...)$ such that $R_n(u_0 u_1 ... u_n) \neq \{1\}$.*

Lemma 9 *If in a real number system, one can multiply numbers around one with an on-line delay δ, then for all integers $n > 0$ there exists a string representing one $U = (u_0 u_1 ...)$ such that $R_n(u_0 u_1 ... u_n)$ includes a real number strictly greater than one and a string U' so that $R_n(u'_0 u'_1 ... u'_n)$ includes a real number strictly less than one.*

Lemma 10 *If in a real number system, one can multiply numbers around one with an on-line delay δ, then for all integers $n > 0$ there exists a string representing one $U = (u_0 u_1 ...)$ such that $R_n(u_0 u_1 ... u_n)$ includes a real number strictly greater than one and another strictly less than one.*

One can easily deduce the proofs of these three lemmas from the proofs of Lemmas 2,3,4 by replacing everywhere the real number zero by one, the on-line addition by the on-line multiplication and the opposite by the inverse.
By noticing that adding a small positive number to a real number can be viewed as multiplying it by a number a little greater than one, the proof of Theorem 1 stands for these theorems too. □

Definition 5 *In a fully redundant and converging number system, let us call Rd_n and rd_n (the maximal and minimal power representation at rank n) the values:*

- $Rd_n = Max\{ \ Max(a_0 ... a_n) - Min(a_0 ... a_n)$ *for all string* $(a_0 ... a_n) \ \}$
- $rd_n = Min\{ \ Max(a_0 ... a_n) - Min(a_0 ... a_n)$ *for all string* $(a_0 ... a_n) \ \}$

Definition 6 *A syntactically dense real number system satisfies: for each A a representable real number and for each integer n, the sets $R_n(a_0...a_n)$ are closed intervals.*

In such a system, we call $M(a_0...a_n)$ the middle of the interval $R_n(a_0...a_n)$ and $I = F(C^{\mathbb{N}})$ the interval of the representable real numbers.

Definition 7 *In a fully redundant, syntactically dense system, let us take $(a_0...a_n)$ a representation of any representation interval I_A (We will use the abusive notation $I_A = R_n(a_0...a_n)$ in the following) then the writing $I_B = R_n(a_0...a_{n-1}b_n)$ is a right neighbor of I_A if and only if:*

- *$Max(a_0...a_{n-1}b_n) > Max(a_0...a_n)$.*
- *For all writing $I_C = (a_0...a_{n-1}c_n)$ so that $Max(a_0...a_{n-1}c_n) > Max(a_0...a_n)$, we have: $Min(a_0...a_{n-1}b_n) \leq Min(a_0...a_{n-1}c_n)$.*

The same definition holds for left neighbor by replacing maxima by minima and $>$'s by $<$'s and conversely. By extension, a representation interval I_B is a right (respectively left) neighbor of a representation interval I_A if and only if I_B has got a writing that is a right (respectively left) neighbor of a writing of I_A.
By construction, the sets $R_n(a_0...a_n)$ are covering I. A representation interval of rank n (with $n+1$ digits) is principal if and only if it is not included in one of its neighbors.
Two representation intervals are said to be neighbors if the first one is a left neighbor of the second one and the second one a right neighbor of the first one. In the following, we will call redundancy degree at rank n the value d_n equal to the length of the smallest intersection between two neighbors.

Lemma 11 *Let us consider two representation intervals $I_A = R_n(a_0...a_n)$ and $I_B = R_n(a_0...a_{n-1}b_n)$ so that I_B is a right neighbor of I_A and I_A is principal, then I_A is a left neighbor of I_B.*

Proof. By construction, $Min(a_0...a_n) < Min(a_0...a_{n-1}b_n)$ (if not, I_A is included in I_B what denies the fact that I_A is principal).
Let us take $I_C = (a_0...a_{n-1}c_n)$ so that $Min(a_0...a_{n-1}c_n) < Min(a_0...a_{n-1}b_n)$. Let us suppose that $Max(a_0...a_{n-1}c_n) > Max(a_0...a_n)$. This is equivalent to assuming that I_A is not a left neighbor of I_B and let us reduce that hypothesis to the absurd:

- **First case:** $Min(a_0...a_{n-1}c_n) > Min(a_0...a_n)$ and in this case, I_C is a right neighbor of I_A what refutes the fact that I_B is one: absurd.
- **Second case:** $Min(a_0...a_{n-1}c_n) \leq Min(a_0...a_n)$ then I_A is strictly included in I_C what is absurd since I_A is principal.

To conclude, I_A is a left neighbor of I_B. $\square$

Lemma 12 *Let us consider a fully redundant and syntactically dense system where for all natural integer n, the redundancy degree at rank n satisfies $d_n > 0$. Let us take a principal representation interval $I_A = (a_0...a_n)$ so that $Max(a_0...a_n) < Max(a_0...a_{n-1})$ (respectively $Min(a_0...a_n) > Min(a_0...a_{n-1})$), then there exists $I_B = (a_0...a_{n-1}b_n)$ so that the length of the intersection of I_A and I_B is at least d_n and $Max(a_0...a_n) < Max(a_0...a_{n-1}b_n)$ (respectively $Min(a_0...a_n) > Min(a_0...a_{n-1}b_n)$).*

Proof. $I_A = (a_0 ... a_n)$ is principal and $Max(a_0...a_n) < Max(a_0...a_{n-1})$. Let us denote E the set of the representation intervals $R_n(a_0...a_{n-1}c^i)$ so that $Max(a_0...a_{n-1}c^i) > Max(a_0...a_n)$. This set is not empty otherwise $Max(a_0...a_{n-1})$ would be equal to $Max(a_0...a_n)$. Let us choose one of the elements of E with the smallest lower bound: $I_B = R_n(a_0...a_{n-1}c^j)$. The intersection of I_A and I_B is not empty otherwise the set of the intervals $R_n(a_0...a_{n-1}c^i)$ does not cover $R_n(a_0...a_{n-1})$. So by definition, I_B is a right neighbor of I_A, thanks to Lemma 11 I_A is a left neighbor of I_B then by definition of the redundancy degree, their intersection is at least of length d_n. We proceed in a similar fashion for the minima.
It is useful to notice that this result holds when n equals zero too if we decide that $R_n(a_0...a_{-1}) = F(C^{\mathbb{N}}) = I$. □

Theorem 4 *In a fully redundant and syntactically dense system, if for all natural integer $n \geq 0$ $0 < Rd_{n+\delta} \leq \frac{1}{2}d_n$ then it is possible to perform on-line addition with delay δ.*

By applying this theorem to the well-known signed-digit representation systems (see [2]) we obtain an online adder of delay 2 in radix 2 and an online adder of delay 1 in higher radices: these are known to be the best delays. The proof builds an effective online algorithm of addition that can be viewed as the most general online addition algorithm possible. It gives upper bounds for the delay of the online adder in "exotic" number systems (For example radix π with digits in the set $\{-2,-1,0,1,2\}$). In the following, a limited converse proposition will be presented.

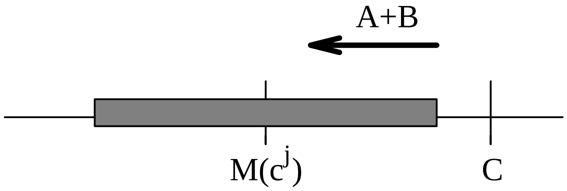
Proof.
We show by induction the following hypothesis $P(n)$:

For all $n \geq 0$, for all representable real numbers A and B whose sum is also representable, we suppose that the first $n+1$ digits of a representation of the sum $c_0, ..., c_n$ have already been produced, when the digits $a_{n+\delta+1}$ and $b_{n+\delta+1}$ are available, it is possible to find a digit c_{n+1} so that

$$A + B \in R_{n+1}(c_0...c_{n+1})$$

First let us prove the hypothesis $P(0)$. Let us define the real number $C = M(a_0...a_\delta) + M(b_0...b_\delta)$. Let us call E the set of letters (digits) c^i of the alphabet so that $C \in R_0(c^i)$.

- **First case:** E is empty. This is possible only if C does not belong to I (since the system is syntactically dense). Let us suppose that $C > Max(I)$ (The other case is similar).
  Let us take c^j so that $Max(c^j) = Max(I)$ and $R_0(c^j)$ is principal.
  If $R_0(c^j) = I$ then $A + B \in R_0(c^j)$ and the new digit to produce is c^j. In the other case, thanks to Lemma 12, there exists a digit c^k so that the length of the interval $R_0(c^j) \cap R_0(c^k)$ be greater or equal to d_0. Let us consider the scheme:



$$As\ |C - (A+B)| \leq \frac{1}{2}(Max(a_0...a_\delta) - Min(a_0...a_\delta) + Max(b_0...b_\delta) - Min(b_0...b_\delta))$$
$$|C - A + B| \leq Rd_p,\ then\ we\ deduce:$$
$$|C - (A+B)| \leq \frac{1}{2}d_0 \qquad (3)$$

$$A\ fortiori,\ |Max(c^j) - (A+B)| < \frac{1}{2} d_0\ and\ A+B \in R_0(c^j)$$

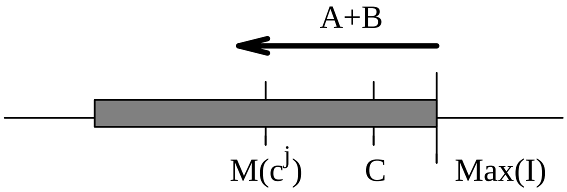
The algorithm produces the digit c^j. QED

- **Second case:** Let us take $R_0(c^j)$ a principal interval so that $C \in R_0(c^j)$. Let us suppose that $Max(c^j) - C \leq C - Min(c^j)$ (In the other case, the proof is identical if we replace each right neighbor by a left neighbor.) then:

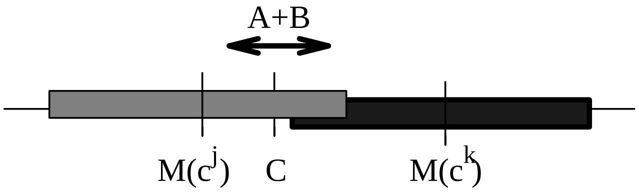
$$C - Min(c^j) \geq \frac{1}{2}\, d_0 \qquad (4)$$

Let us call this step in the proof the step *Look for right neighbor*.

- If $R_0(c^j)$ does not have any right neighbor, we immediatly deduce from Lemma 12 that $Max(c^j) = Max(I)$ and with (3) it is obvious that $A + B \in R_0(c^j)$ QED.

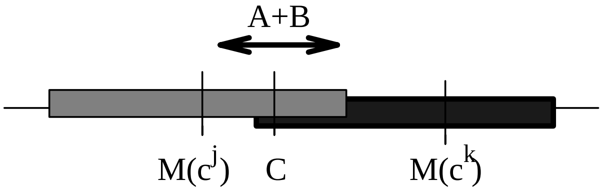


- Let us call $R_0(c^k)$ a principal interval which is a right neighbor of $R_0(c^j)$.
  - If C does not belong to $R_0(c^k)$ the digit c^j is produced:



As the length of the intersection of the intervals $R_0(c^j)$ and $R_0(c^k)$ is greater or equal to d_0, With (4), we deduce that $A + B \in R_0(c^j)$ QED.

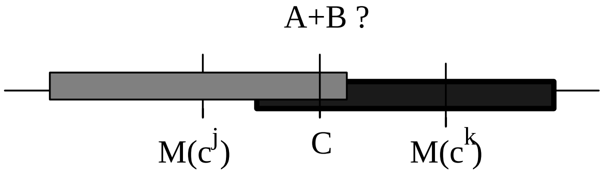
  - If C belongs to $R_0(c^k)$ and $Max(c^j) - C \geq C - Min(c^k)$, the digit c^j is produced.



We immediatly deduce that $Max(c^j) - C \geq \frac{1}{2} d_0$ and with (4) $A + B \in R_0(c^j)$ QED.

  - If C belongs to $R_0(c^k)$ and $C - Min(c^k) > Max(c^j) - C$:

Then we consider the interval $R_0(c^k)$ instead of $R_0(c^j)$ and restart the process at the step *Look for right neighbor*. We easily check that equation (4) is always true after the substitution.



Since the process never enters an infinite loop (The right neighbor substitution can be called at most cardinal($alphabet$) times), we are able to produce in any case a digit c^j so that $A+B \in R_0(c^j)$ with the first $\delta+1$ digits of A and B and the recurrent hypothesis $P(n)$ is proved at step $n=0$.

Now, let us suppose that the hypothesis $P(n)$ is true for an $n \geq 0$. Let us prove $P(n+1)$: with the first $n+\delta+1$ digits of A and B, the first $n+1$ digits of a representation of $A+B$ are already produced: $d_0...d_n$.
Let us consider $C = M(a_0...a_{n+\delta+1}) + M(b_0...b_{n+\delta+1})$. Let us call E the set of letters (digits) c^i of the alphabet such that $C \in R_0(c^i)$.
We develop exactly the same cases as for $n=0$. The proofs are strictly identical except for the interval I which is replaced everywhere by $R_n(d_0...d_n)$.
By recurrence, the hypothesis is proved for all natural integer n. □

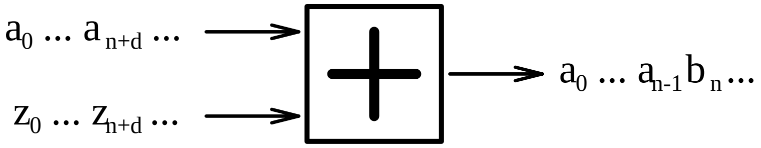
Definition 8 *In a redundant real number system an adder is said to be reversible if and only if for all $A = (a_0...)$ and $B = (b_0...)$ two representations of the same real number, there exists an infinite string $Z_{A,B}$ representing zero so that $A + Z_{A,B} = B$.*

Theorem 5 *In a fully redundant syntactically dense number system, if there exists n so that $rd_{n+\delta} > d_n$, then no algorithm can perform a reversible addition in delay δ.*

Proof. Let us take n so that $rd_{n+\delta} > d_n$. Let us choose $I_A = (a_0...a_n)$ and $I_B = (a_0...a_{n-a}b_n)$ so that:

- I_A be a left neighbor of I_B.
- I_B be a right neighbor of I_A.
- The length of the intersection of I_A and I_B is equal to d_n.

Let us denote $A = Max(I_A) = (a_0...a_n...a_{n+\delta}...) = (a_0...a_{n-1}b_n...)$ and $Z_{A,B} = (z_0z_1...)$. Let us suppose that it is possible to perform addition in delay δ and reduce it to the absurd.



If we add serially $Min(a_0...a_{n+\delta})$ and $Min(z_0...z_{n+\delta})$, we obtain a real number coded by a string beginning with $(a_0...a_{n-1}b_n)$ too. We immediatly deduce that the length of $I_A \cap I_B$ is greater or equal to $rd_{n+\delta}$ which refutes the fact that $rd_{n+\delta} > d_n$. □

Definition 9 *By slightly modifying the previous set of definitions, one can adapt them to describe relative and natural integer number systems: one works only on finite length strings instead. A system would be converging if the sets $R_n(a_0 ... a_n)$ are included in a strictly length decreasing intervals sequence $(I_n)_{n \in \mathbb{N}}$:*

- *for $0 \leq n \leq n_{max}$ $R_n(a_o ... a_n) \subset I_n$ and $R_n(a_o ... a_n) \not\subset I_{n+1}$*
- *$I_0 \supset I_1 \supset ... \supset I_{n_{max}}$ and $I_{n_{max}}$ contains only one integer.*

The redundancy definition does not change and we obtain the same results concerning redundancy and on-line addition with delay δ in relative integer number systems. A relative or natural integer number system is syntactically dense if the $R_n(a_0 ... a_n)$ are intervals of Z or $\mathbb{N}$. Theorems 4 and 5 still holds for relative and natural integers: one only has to be cautious to find the δ correct last digits since the algorithm does not produce them but still prove that they exist.

Lemma 13 *This is to show how to use Theorem 4 in a practical case. For M any large natural integer, we define the encoding function F from $\{0, 1\}^{M+1}$ into $\mathbb{N}$ by:*
$F(a_0 ... a_M) = \sum_{i=0}^{M} a_i U_{M-i}$ where U_i is the Fibonacci sequence with $U_0 = 1$ and $U_1 = 2$. This system is called the Fibonacci Numeration system (see [5]). In this system, it is possible to perform an on-line addition with delay 3.

Proof. The fact that this system is syntactically dense derives from the normalization algorithm that finds for each integer in $\{0, \sum_{i=0}^{M} U_i\}$ a canonical representation by using the Euclidean division. By definition, $R_n(a_0 ... a_n) = [\sum_{i=0}^{n} a_i\, U_{M-i}, \sum_{i=0}^{n} a_i\, U_{M-i} + \sum_{i=0}^{M-n-1} U_i]$.
From that we immediatly deduce that this system is fully redundant and that:

$$Rd_n(a_0 ... a_n) = rd_n(a_0 ... a_n) = \sum_{i=0}^{M-n-1} U_i$$

$$d_n = (\sum_{i=0}^{M-n-1} U_i) - U_{N-n-1} = Rd_{n+1}$$

Thanks to Theorem 4, it is possible to produce the first $M - 2$ digits of a representation of the sum with an on-line adder of delay 3 (when the $M + 1$ digits of both operands have been serially supplied) if for all $n \in \{0, ..., M - 3\}$:

$$Rd_{n+3} \leq \frac{1}{2} d_n \qquad (5)$$

That is to say $\forall n \in \{0, ..., M - 3\}$,

$$Rd_{n+3} \leq \frac{1}{2} Rd_{n+1}$$

That is equivalent to $\forall n \in \{0, ..., M - 3\}$:

$$Rd_{n+1} \leq 2U_{M-n-1}$$

By replacing $n + 1$ by $M - i$, $\forall i \in \{2, ..., M - 1\}$:

$$Rd_{M-i} \leq 2U_i$$

We prove that proposition by induction on i:

First $Rd_{M-2} = U_0 + U_1 \leq 2U_2$

Let suppose the proposition true at step i:
$Rd_{M-(i+1)} = Rd_{M-i} + U_i \leq 2U_i + U_{i-1} + U_{i-2} \leq 2U_i + 2U_{i-1} = 2U_{i+1}$ QED.

At the step M, the on-line adder has got all the digits of the operands then it can produce the three last digits of the sum. As a matter of fact, one can easily adapt the general on-line adding algorithm to find a finite automaton that adds in delay 3 in the Fibonacci Numeration system. □

3 Conclusion

In this paper, we have presented a set of definitions from which one can derive a classification of non-classical number representation systems (see figure 1). Details on the number systems can be found in the corresponding articles [2, 3, 4, 5, 7, 8, 10, 11, 12].

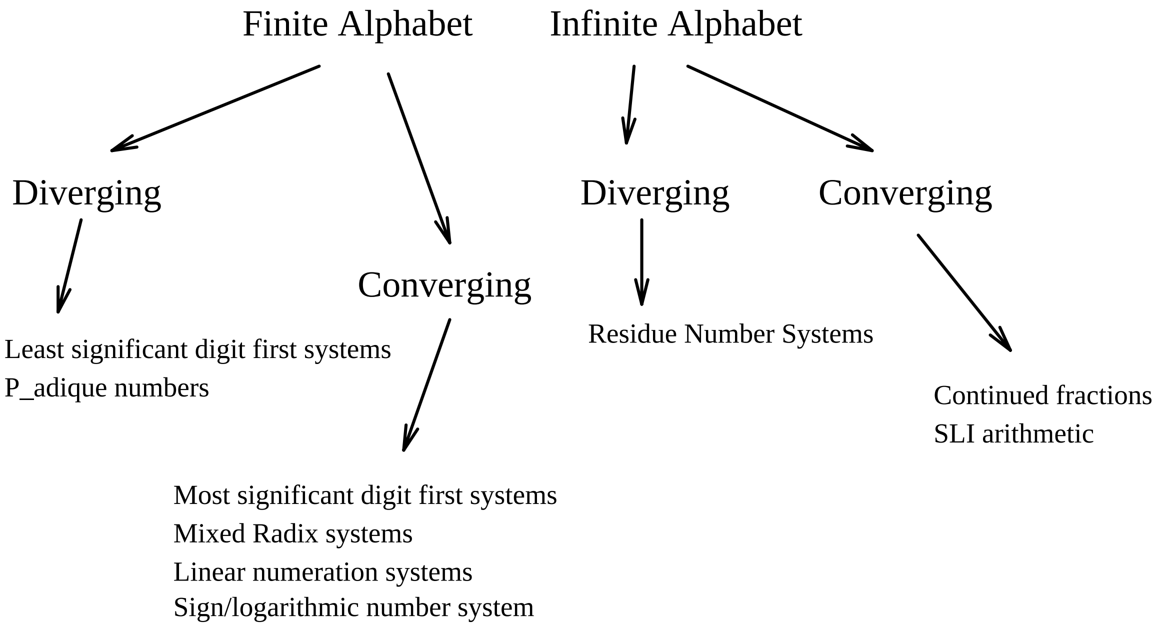


Figure 1: Real number and numeration system classification.

From a general point of view, converging systems are characterized by the fact that the usual arithmetic operations are expensive (At least, addition or multiplication is logarithmic compared to the size of the operands) but they allow serial computations and comparison algorithms are straightforward. On the other hand, in diverging systems the arithmetic operations can be very efficient (even constant time in residue arithmetic) but comparisons are very costly and serial calculus can not be achieved for division. The dream of every computer arithmetician is to break the wall between these two classes and to build an hybrid arithmetic that only keeps the advantages of both sides. The exploration path we have chosen is the one of redundancy. The reasons are multiple: by accepting to lose some information on numbers we gain a freedom degree that allows to build more efficient algorithms at the long run. Thanks to redundancy, it is possible to build systems where numbers flow serially digit by digit which is the only way to handle large numbers without a tremendous hardware cost. Another point is how far we are able to represent “real” real numbers. Most systems only represent and handle a bounded set of rational numbers although the recursive real numbers set a theoretical far-beyond frontier (see [9]). Redundancy by allowing serial calculus on converging systems is the key to “practical exact” real arithmetic. (No hope to obtain a practical exact real arithmetic on a diverging system since one may need an arbitrary large number of digits to only know if the result is close to zero.) We will conclude by giving some conjectures whose confirmation or invalidation would no doubt lead to the design of useful tools and results for computer arithmetic:

- Let us consider an exact real number system with a finite alphabet that can represent real numbers of any magnitude, then no finite automaton can perform serially addition on such a system.
- In no integer number system, one can as well perform addition, multiplication and comparison with finite automata.

References

[1] Y. Amice Les nombres p-adiques. *Presses Universitaires de France*, Col. Sup., 1975.

[2] A. Avizienis Signed-digit number representations for fast parallel arithmetic. *IRE Trans. Electron. Comput. 10*, pp 389–400, Sept 1961.

[3] M.D.Ercegovac and K.S.Trivedi On-line algorithms for division and multiplications *IEEE Transactions on Computers*,Vol. C-26 N. 7, pp. 681-687, July 1977.

[4] A. S. Fraenkel Systems of numeration *The American Mathematician Monthly*, pp 105–114, Feb. 1985.

[5] C. Frougny Representation of numbers in Non-Classical Numeration Systems. *Proc 10th IEEE Symp. on Comp. Arith.*, pp 17–21, June 1991.

[6] A.Guyot,Y.Herreros and J.M.Muller JANUS, An on-line multiplier/divider for manipulating large numbers. 9^{th} *Symposium On Computer Arithmetic*, Santa-Monica, Sept 1989.

[7] D. E. Knuth *The Art of Computer Programming*, vol 2, Seminumerical algorithms, pp 178–364, Addison Wesley, 1981.

[8] E. V. Krishnamurthy, T. M. Rao, K. Subramanian Finite-segment p-adic number systems with applications to exact computations. *Proc. Indian Acad. Sci.*, vol. LXXXI, pp 58–79, 1975.

[9] H. G. Rice Recursive real numbers. *Proc. Americ. Math. Soc.*, vol. 5, no. 5, pp 784–791, 1954.

[10] E. E. Swartzlander and A. G. Alexopoulos The sign/logarithmic number system *IEEE Transactions on Computers*, vol C-24, pp 1238–1242, Dec. 1975.

[11] P. E. Turner Will the "Real" Real Arithmetic Please Stand Up? *Comp. and Math.*, vol 38, Number 4, pp 298–304, April 1991.

[12] J. E. Vuillemin Exact Real Computer Arithmetic with Continued Fractions *IEEE Transactions on Computers*, vol 39, Number 8, pp 1087–1105, Aug. 1990.

[13] E. Wiedmer Computing with infinite objects *Theor. Comput. Sci.*, vol 10, pp. 133–155, 1980.