On-Line Addition in Real Base *

Christiane Frougny

[1] Université Paris 8
[2] C.N.R.S., L.I.A.F.A.
Case 7014, 2 place Jussieu, 75251 Paris Cedex 05, France
Christiane.Frougny@liafa.jussieu.fr

Abstract. Let β be a real number > 1. Addition and multiplication by a fixed positive integer of real numbers represented in base β are shown to be computable by an on-line algorithm, and thus are continuous functions. When β is a Pisot number, these functions are computable by an on-line finite automaton.

1 Introduction

In computer arithmetic, on-line computation consists of performing arithmetic operations in Most Significant Digit First (MSDF) mode, digit serially after a certain latency delay [7]. This allows the pipelining different operations such as addition, multiplication and division. It is also appropriate for the processing of real numbers having infinite expansions. It is well known that when multiplying two real numbers, only the left part of the result is significant. To be able to perform on-line addition, it is necessary to use a redundant number system (see [15], [7]).

We now take a different point of view. A function is computable by a finite automaton if it needs only a finite auxiliary storage memory, independent of the size of the data. In that setting, one knows that addition of two integers in the classical b-ary system is computable by a finite automaton, but that squaring is not (see [6]). Actually, the natural finite automaton one designs to perform addition is a sequential one, processing numbers in the Least Significant Digit First (LSDF) mode.

On-line finite automata have been introduced by Muller [12]. They are sequential finite automata processing data in MSDF mode. In integral base b on the canonical digit set $\{0, \dots, b-1\}$, addition is not on-line computable, but with a balanced alphabet of signed digits of the form $\{-a, \dots, a\}$ with $b/2 \leq a \leq b-1$, using the algorithms of Avizienis [1] and Chow and Robertson [5], addition is computable by an on-line finite automaton (see [12], [11]). In the same spirit we have shown that, in a complex base of the form $\sqrt[m]{b}$, where b is a relative integer such that $|b| \geq 2$, and with digit set $\{-a, \dots, a\}$ with $|b|/2 \leq a \leq |b|-1$, addition is computable by an on-line finite automaton as well [10].

* To appear in the proceedings of MFCS'99

In this paper we consider a base β which is a real number > 1, generally not an integer. For any $\beta > 1$, by the greedy algorithm of Rényi [14], one can compute a representation in base β of any real number from the interval $[0, 1]$, called its β-expansion, and where the digits are elements of the canonical digit set $A = \{0, \ldots, \lfloor\beta\rfloor\}$. In such a representation, not all the patterns of digits are allowed (see [13] for instance). Recall that a Pisot number (also called a Pisot-Vijayaraghavan number) is an algebraic integer such that all its algebraic conjugates have modulus less than 1. The natural integers and the golden ratio are Pisot numbers. In a previous work, we have shown that addition in real base β is computable by a finite automaton if and only if β is a Pisot number [2], but the automaton is not sequential, that is to say, data are not processed deterministically in MSDF nor in LSDF mode. Note that the result given by this automaton is the greedy β-expansion of the sum of the numbers. On the other hand, it is known that it is not possible to find a sequential finite automaton realizing addition and giving the result under the β-expansion form [9].

We will say that the addition is *nonnormalized* if the result belongs to the canonical digit set A, but *need not be* the greedy β-expansion of the sum of the numbers.

For some Pisot bases of a special kind, nonnormalized addition was known to be computable by an on-line finite automaton, namely for bases $\beta > 1$ where β is the dominant root of an equation of the form

$$X^m - aX^{m-1} - aX^{m-2} - \cdots - aX - b$$

where $a \geq b \geq 1$ are integers, and $m \geq 2$. The most well-known case is the golden ratio $(1 + \sqrt{5})/2$, with $m = 2$, $a = b = 1$ (see [10]).

In this work we generalize this result to any Pisot number.

The paper is organized as follows. First we show that for any real base $\beta > 1$ and any set of nonnegative digits $D \supseteq A = \{0, \ldots, \lfloor\beta\rfloor\}$, the conversion which transforms a β-representation with digits in D onto an equivalent β-representation with digits in A (but not necessarily the β-expansion) is computable by an on-line algorithm (Theorem 1). We then show that such a function is continuous for the product topology on the set $D^{\mathbb{N}}$, and the function induced on real numbers is continuous as well.

When β is a Pisot number, the conversion is computable by an on-line finite automaton (Theorem 2). Note that this result applies to the case where β is an integer and the alphabet A is equal to $\{0, \ldots, \beta\}$. The case $\beta = 2$ and $A = \{0, 1, 2\}$ is the well-known "carry-save" representation used in computer arithmetic.

Nonnormalized addition and multiplication by a fixed positive integer are particular cases of digit set conversions.

In the case where β is a Pisot number, one can define linear numeration systems associated with β, like the Fibonacci numeration system associated with the golden ratio. In these systems, any natural number has a greedy representation by an algorithm of Fraenkel [8]. Then the digit set conversion in a linear numeration system associated with a Pisot number is computable by an on-line finite automaton.

2 Preliminaries

An *alphabet* A is a finite set. A finite sequence of elements of A is called a *word*, and the set of words on A is the free monoid A^*. The *empty word* is denoted by ε. The set of infinite sequences or infinite words on A is denoted by $A^{\mathbb{N}}$. Let v be a word of A^*, denote by v^n the concatenation of v to itself n times, and by v^ω the infinite concatenation $vvv\cdots$.

2.1 Beta-Representations

Let $\beta > 1$ be a real number. A *β-representation* of a number x of $[0,1]$ is an infinite sequence $(d_k)_{k\geq 1}$ such that $\sum_{k\geq 1} d_k\beta^{-k} = x$.

Any real number $x \in [0,1]$ can be represented in base β by the following greedy algorithm [14]:
Denote by $\lfloor . \rfloor$ and by $\{.\}$ the integral part and the fractional part of a number. Let $x_1 = \lfloor \beta x \rfloor$ and let $r_1 = \{\beta x\}$. Then iterate for $k \geq 2$, $x_k = \lfloor \beta r_{k-1} \rfloor$ and $r_k = \{\beta r_{k-1}\}$.
Thus $x = \sum_{k\geq 1} x_k\beta^{-k}$, where the digits x_k are elements of the *canonical* alphabet $A = \{0, \ldots, \lfloor\beta\rfloor\}$ if $\beta \notin \mathbb{N}$, $A = \{0, \ldots, \beta - 1\}$ otherwise. The sequence $(x_k)_{k\geq 1}$ of $A^{\mathbb{N}}$ is called the *β-expansion* of x. When β is not an integer, a number x may have several different β-representations on A: this system is naturally redundant. The β-expansion obtained by the greedy algorithm is the greatest one in the lexicographic ordering. When a β-representation ends with infinitely many zeroes, it is said to be *finite*.

Let D be a digit set. The *numerical value* in base β on D is the function $\pi_\beta : D^{\mathbb{N}} \longrightarrow \mathbb{R}$ such that $\pi_\beta((d_k)_{k\geq 1}) = \sum_{k\geq 1} d_k\beta^{-k}$. Given two digit sets A and D, a *digit set conversion* in base β from D to A is a function $\chi : D^{\mathbb{N}} \longrightarrow A^{\mathbb{N}}$ such that for each sequence $(d_k)_{k\geq 1} \in D^{\mathbb{N}}$ where $x = \pi_\beta((d_k)_{k\geq 1})$ belongs to $[0,1]$, there exists a sequence $(a_k)_{k\geq 1} \in A^{\mathbb{N}}$ such that $x = \pi_\beta((a_k)_{k\geq 1})$. Note that apriori the result of a conversion is not unique, but all the processes we shall be consider later on are deterministic, and thus compute functions.

Remark that the image $\chi((d_k)_{k\geq 1})$ belongs to $A^{\mathbb{N}}$, but need not be the β-expansion of x as computed by the greedy algorithm.

To perform addition in base β the process is the following one: take two numbers $v = \sum_{k\geq 1} v_k\beta^{-k}$ and $y = \sum_{k\geq 1} y_k\beta^{-k}$ with v_k and y_k in A, such that $v + y \in [0,1]$. Set $z_k = v_k + y_k$. Then z_k is an element of $B = \{0, \ldots, 2\lfloor\beta\rfloor\}$, and $v + y = \sum_{k\geq 1} z_k\beta^{-k}$. Addition consists of transforming the representation $(z_k)_{k\geq 1}$ of $v + y$ on B into an equivalent one $(s_k)_{k\geq 1}$, such that $v + y = \sum_{k\geq 1} s_k\beta^{-k}$, with $s_k \in A$. Multiplication by a fixed integer $m \geq 1$ is analogous: multiply by m each digit of the β-expansion. This gives a sequence on the alphabet $\{0, \ldots, m\lfloor\beta\rfloor\}$, to be converted into an equivalent β-representation on A. Nonnormalized addition and multiplication by a fixed positive integer are thus special cases of digit set conversion.

2.2 On-Line Computability

Let X and Y be two finite digit sets, and let φ be a function from $X^{\mathbb{N}}$ to $Y^{\mathbb{N}}$ (for simplicity we consider only one-variable functions, but it is not a restriction). X is the *input* alphabet, and Y is the *output* alphabet. Following [15] and [7], we say that φ is *on-line computable* with delay δ if there exists a natural number δ such that, given $x = (x_k)_{k \geq 1}$, to compute $y = \varphi(x) = (y_k)_{k \geq 1}$, it is necessary and sufficient to have the digits $x_1, \ldots, x_{k+\delta}$ available to generate y_k, for $k \geq 1$. After the delay, one digit of the result is produced upon receiving one digit of X. It is well known that some functions are not on-line computable, like addition in the binary system with canonical digit set $\{0, 1\}$. Addition is considered as a conversion χ from $\{0, 1, 2\}$ to $\{0, 1\}$. Since $\chi(01^n 20^\omega) = 10^\omega$ and $\chi(01^n 0^\omega) = 01^n 0^\omega$ for any $n \geq 1$, one sees that the least significant digits have to be known to be able to output the most significant digit of the result.

2.3 Automata

We refer the reader to [6]. An *automaton over* A, $\mathcal{A} = (Q, A, E, I, T)$, is a directed graph labelled by elements of A. The set of vertices, traditionally called *states*, is denoted by Q, $I \subset Q$ is the set of *initial* states, $T \subset Q$ is the set of *terminal* states and $E \subset Q \times A \times Q$ is the set of labelled *edges*. If $(p, a, q) \in E$, we note $p \xrightarrow{a} q$. The automaton is *finite* if Q is finite. The automaton $\mathcal{A}$ is *deterministic* if E is the graph of a (partial) function from $Q \times A$ into Q, and if there is a unique initial state. A subset H of A^* is said to be *recognizable by a finite automaton* if there exists a finite automaton $\mathcal{A}$ such that H is equal to the set of labels of paths starting in an initial state and ending in a terminal state. A subset K of $A^{\mathbb{N}}$ is said to be *recognizable by a finite automaton* if there exists a finite automaton $\mathcal{A}$ such that K is equal to the set of labels of infinite paths starting in an initial state and going infinitely often through a terminal state (Büchi acceptance condition, see [6]).

In this paper we are interested in 2-tape automata. Let X and Y be two alphabets. A *2-tape automaton* is an automaton over the non-free monoid $X^* \times Y^*$: $\mathcal{A} = (Q, X^* \times Y^*, E, I, T)$ is a directed graph the edges of which are labelled by elements of $X^* \times Y^*$. Words of X^* are referred to as *input words*, as words of Y^* are referred to as *output words*. If $(p, (f, g), q) \in E$, we note $p \xrightarrow{f/g} q$. The automaton is finite if Q and E are finite. The finite 2-tape automata are also known as *transducers*. A relation R of $X^* \times Y^*$ is said to be *computable by a finite 2-tape automaton* if there exists a finite 2-tape automaton $\mathcal{A}$ such that R is equal to the set of labels of paths starting in an initial state and ending in a terminal state. A function is computable by a finite 2-tape automaton if its graph is computable by a finite 2-tape automaton. These definitions extend to relations and functions of infinite words as above.

A *sequential* finite automaton is a finite 2-tape automaton where edges are labelled by elements of $X \times Y^*$, and such that the *underlying input automaton* obtained by taking the projection over X of the label of every edge is deterministic. An *on-line finite automaton* with delay δ, $\mathcal{A} = (Q, X \times (Y \cup \varepsilon), E, \{i_0\}, \omega)$, is a sequential automaton composed of a transient part and of a synchronous part (see [12], [11]). The set of states is equal to $Q = Q_t \cup Q_s$, where Q_t is the set of transient states and Q_s is the set of synchronous states. In the transient part, every path of length δ starting in the initial state i_0 is of the form

$$i_0 \xrightarrow{x_1/\varepsilon} i_1 \xrightarrow{x_2/\varepsilon} \cdots \xrightarrow{x_\delta/\varepsilon} i_\delta$$

where $i_0, \ldots, i_{\delta-1}$ are in Q_t, x_j in X, for $1 \leq j \leq \delta$, and the only edge arriving in a state of Q_t is as above. In the synchronous part, edges are labelled by elements of $X \times Y$. This means that the automaton starts reading words of length $\leq \delta$ outputting nothing, and after that delay, outputs serially one digit for each input digit.

For finite words, there is a *terminal* function $\omega : Q_s \longrightarrow Y^*$, whose value is concatenated to the output word corresponding to a computation in $\mathcal{A}$. The same definition works for functions of infinite words, considering infinite paths in $\mathcal{A}$, but there is no terminal function ω in that case.

3 On-Line Digit Set Conversion in Real Base

Let $A = \{0, \ldots, \lfloor\beta\rfloor\}$ be the canonical alphabet associated with β, and let $D = \{0, \ldots, d\}$ be a digit set containing A, that is, $d \geq \lfloor\beta\rfloor$.

Theorem 1. *There exists a digit set conversion $\chi : D^{\mathbb{N}} \longrightarrow A^{\mathbb{N}}$ in base β which is on-line computable with delay δ, where δ is the smallest positive integer such that*

$$\beta^{\delta+1} + d \leq \beta^\delta(\lfloor\beta\rfloor + 1) \qquad (*)$$

Proof. Clearly a number δ satisfying (*) exists. Let k be the smallest nonnegative integer such that $d/\beta^k(\beta - 1) \leq 1$. In order to avoid overflow, all input words are supposed to begin with a run of k zeroes.

On-line algorithm.

Input: a sequence $(d_j)_{j\geq1} \in D^{\mathbb{N}}$ *such that* $d_1 = \cdots = d_k = 0$.
Output: a sequence $(a_j)_{j\geq1} \in A^{\mathbb{N}}$ *such that* $\sum_{j\geq1} a_j\beta^{-j} = \sum_{j\geq1} d_j\beta^{-j}$.

begin
  $q_0 \leftarrow 0$
  for $i \leftarrow 1$ to δ **do**
    $q_i \leftarrow \beta q_{i-1} + d_i$
  $j \leftarrow 1$
  while $j \geq 1$ **do**
    $z_{\delta+j} \leftarrow \beta q_{\delta+j-1} + d_{\delta+j}$
    if $z_{\delta+j} < \beta^{\delta+1}$
      then $a_j \leftarrow \lfloor z_{\delta+j}/\beta^\delta\rfloor$
      else $a_j \leftarrow \lfloor\beta\rfloor$
    $q_{\delta+j} \leftarrow z_{\delta+j} - \beta^\delta a_j$

$j \leftarrow j+1$

end

Proof of the algorithm.

Claim: For all $j \geq 1$, one has $0 \leq q_j < \beta^\delta$ and $a_j \in A$.

1. For $1 \leq i \leq \delta$, we get

$$q_i = \beta^{i-1} d_1 + \cdots + d_i$$

Then $q_i \leq q_\delta < \beta^\delta$ by hypothesis on the input $(d_j)_{j \geq 1}$.

2. Suppose that for some $j \geq 1$

$$0 \leq q_{\delta+j-1} < \beta^\delta \tag{H}$$

• If $z_{\delta+j} < \beta^{\delta+1}$ then $a_j = \lfloor z_{\delta+j}/\beta^\delta \rfloor$. Thus $0 \leq a_j < \beta$. Then $q_{\delta+j} = \beta^\delta \{z_{\delta+j}/\beta^\delta\} < \beta^\delta$.
• If $z_{\delta+j} \geq \beta^{\delta+1}$ then $a_j = \lfloor \beta \rfloor$ and $q_{\delta+j} = z_{\delta+j} - \lfloor \beta \rfloor \beta^\delta$. Then $q_{\delta+j} \geq \beta^{\delta+1} - \lfloor \beta \rfloor \beta^\delta \geq 0$. On the other hand, $q_{\delta+j} = \beta q_{\delta+j-1} + d_{\delta+j} - \lfloor \beta \rfloor \beta^\delta < \beta^{\delta+1} + d - \lfloor \beta \rfloor \beta^\delta$ by hypothesis (H), thus by condition (*), $q_{\delta+j} < \beta^\delta$. Hence the claim is proved.

We then get, for all $j \geq 1$,

$$\frac{d_1}{\beta} + \cdots + \frac{d_{\delta+j}}{\beta^{\delta+j}} = \frac{a_1}{\beta} + \cdots + \frac{a_j}{\beta^j} + \frac{q_{\delta+j}}{\beta^{\delta+j}}.$$

Since $q_{\delta+j} < \beta^\delta$, when j tends towards infinity, $\sum_{j \geq 1} d_j \beta^{-j} = \sum_{j \geq 1} a_j \beta^{-j}$, with the digits a_j in A, thus $\chi((d_j)_{j \geq 1}) = (a_j)_{j \geq 1}$. □

Remark 1. If δ satisfies (*), then any natural $\gamma > \delta$ satisfies (*) as well.

4 Continuity

Let D be a finite alphabet. One defines a distance ρ on $D^{\mathbb{N}}$ as follows: let $v = (v_i)_{i \geq 1}$ and $w = (w_i)_{i \geq 1}$ be in $D^{\mathbb{N}}$, $\rho(v, w) = 2^{-r}$ where $r = \min\{i \mid v_i \neq w_i\}$. The set $D^{\mathbb{N}}$ is then a compact metric space. This topology is equivalent to the product topology. We first prove a general result, not related to the base.

Proposition 1. *Let D and A be two finite alphabets. Any function $\varphi : D^{\mathbb{N}} \longrightarrow A^{\mathbb{N}}$ which is on-line computable with delay δ is 2^δ-lipschitzian, and is thus uniformously continuous.*

Proof. Suppose that φ is on-line computable with delay δ. Let v and w in $D^{\mathbb{N}}$ such that $\rho(v, w) = 2^{-r}$. Then $v = v_1 \cdots v_{r-1} v_r v_{r+1} \cdots$ and $w = v_1 \cdots v_{r-1} w_r w_{r+1} \cdots$ with $v_r \neq w_r$. Thus $\varphi(v) = y_1 \cdots y_{r-\delta-1} y_{r-\delta} y_{r-\delta+1} \cdots$ and $\varphi(w) = y_1 \cdots y_{r-\delta-1} s_{r-\delta} s_{r-\delta+1} \cdots$. Thus $\rho(\varphi(v), \varphi(w)) \leq 2^\delta \rho(v, w)$. □

As a corollary we get the following result.

Proposition 2. *Let D be a set of nonnegative digits containing A. The digit set conversion $\chi : D^{\mathbb{N}} \longrightarrow A^{\mathbb{N}}$ in base β defined in Theorem 1 is uniformously continuous.*

The results presented below are a straightforward generalization of those proved by Eilenberg [6] in the case where β is an integer.

Proposition 3. *Let D be a finite alphabet of digits. The function numerical value $\pi_\beta : D^{\mathbb{N}} \longrightarrow \mathbb{R}$ is continuous.*

Proof. Let v and w in $D^{\mathbb{N}}$ such that $\rho(v, w) = 2^{-r}$. Then $v = v_1 \cdots v_{r-1} v_r v_{r+1} \cdots$ and $w = v_1 \cdots v_{r-1} w_r w_{r+1} \cdots$ with $v_r \neq w_r$. Let $m(D)$ be the maximum of the absolute values of the elements of D. Thus

$$|\pi_\beta(v) - \pi_\beta(w)| = |\sum_{i \geq r} v_i \beta^{-i} - \sum_{i \geq r} w_i \beta^{-i}| \leq \frac{2\ m(D)}{\beta^{r-1}(\beta - 1)}$$

thus π_β is continuous. □

We now consider functions taking their values in base β into the unit interval $[0, 1]$. Let $A = \{0, \ldots, \lfloor \beta \rfloor\}$ and $D = \{0, \ldots, d\}$, $d \geq \lfloor \beta \rfloor$. A function $\chi : D^{\mathbb{N}} \longrightarrow A^{\mathbb{N}}$ is said to be *β-consistent* if there exists a function $f : [0, 1] \longrightarrow [0, 1]$ such that the diagram

$$\begin{tikzcd} D^{\mathbb{N}} \arrow[r, "\chi"] \arrow[d, "\pi_\beta"'] & A^{\mathbb{N}} \arrow[d, "\pi_\beta"] \\ {[0,1]} \arrow[r, "f"'] & {[0,1]} \end{tikzcd}$$

commutes.

Proposition 4. *Let f and χ as above. If χ is on-line computable then f is continuous.*

Proof. Since χ and π_β are continuous, so is $\pi_\beta \circ f$. Since π_β is surjective and continuous, and $D^{\mathbb{N}}$ and $[0, 1]$ are compact metric spaces, the continuity of f follows. □

5 The Pisot Case

An *algebraic integer* is a root of a monic polynomial with integral coefficients. A *Pisot number* is an algebraic integer > 1 such that all its algebraic conjugates are smaller than 1 in modulus.

Theorem 2. *Let β be a Pisot number, let $A = \{0, \ldots, \lfloor \beta \rfloor\}$, and let $D = \{0, \ldots, d\}$ such that $d \geq \lfloor \beta \rfloor$. There exists an an on-line finite state automaton with delay δ, where δ satisfies (*), which realizes a digit set conversion $\chi : D^{\mathbb{N}} \longrightarrow A^{\mathbb{N}}$ in base β.*

Proof. Let $M(X)$ be the minimal polynomial of β, of degree m, and let $\beta_1 = \beta$, $\beta_2, \ldots, \beta_m$ be its roots. For $2 \le i \le m$, $|\beta_i| < 1$. Recall that $\Lambda = \mathbb{Z}[X]/(M(X))$ is a discrete lattice of rank m. Define

$$\Lambda_\delta = \begin{cases} \Lambda \text{ if } m \le \delta \\ \{q(X) = z_{\delta-1}X^{\delta-1} + \cdots + z_0 + z_{-1}X^{-1} + \cdots + z_{\delta-m}X^{\delta-m} \mid X^{m-\delta}q(X) \in \Lambda\} \\ \text{otherwise} \end{cases}$$

The *norm* of an element q of Λ_δ is taken as $\|q\| = \max_{1 \le i \le m} |q(\beta_i)|$. For $2 \le i \le m$ set

$$\gamma_i = \sup\{|c - a\beta_i^\delta| \mid c \in D,\ a \in A\}.$$

We define an on-line finite state automaton $\mathcal{A} = (Q, D \times (A \cup \varepsilon), E, \{q_0\})$ as follows. The set of synchronous states is equal to

$$Q_s = \{q(X) \in \Lambda_\delta \mid 0 \le q(\beta) < \beta^\delta \text{ and for } 2 \le i \le m,\ |q(\beta_i)| < \frac{\gamma_i}{1 - |\beta_i|}\}.$$

Since for any q in Q_s, $\|q\|$ is bounded, Q_s is a finite set. The set of transient states Q_t is defined by

$$Q_t = \{q_j(X) = d_1X^{j-1} + \cdots + d_j \mod M \mid 1 \le j \le \delta - 1,\ d_1, \ldots, d_j \in D\} \cup \{q_0\}.$$

Note that if $q_j \in Q_t$ then $q_j(\beta) < \beta^\delta$; and, for $2 \le i \le m$, $|q_j(\beta_i)| < d/(1 - |\beta_i|) \le \gamma_i/(1 - |\beta_i|)$. Hence transient states satisfy the same bound inequalities as synchronous states. For $1 \le i \le \delta$, transient edges are defined by

$$q_{i-1}(X) \xrightarrow{d_i/\varepsilon} q_i(X)$$

with $q_0(X) = 0$ and $q_i(X) = Xq_{i-1}(X) + d_i$.

The synchronous edges are defined by: for $j \ge 1$ and $q_{\delta+j-1}(X) \in Q$, define an edge

$$q_{\delta+j-1}(X) \xrightarrow{d_{\delta+j}/a_j} q_{\delta+j}(X)$$

such that

$$Xq_{\delta+j-1}(X) + d_{\delta+j} = X^\delta a_j + q_{\delta+j}(X) \mod M(X)$$

with a_j in A. For the choice of a_j we process as in the on-line algorithm given in Theorem 1, replacing X by β. Hence for all $j \ge 0$, $0 \le q_{\delta+j}(\beta) < \beta^\delta$.

For $2 \le i \le m$, we get

$$|q_{\delta+j}(\beta_i)| = |\beta_i q_{\delta+j-1}(\beta_i) + d_{\delta+j} - a_j\beta_i^\delta| < |\beta_i|\frac{\gamma_i}{1 - |\beta_i|} + \gamma_i = \frac{\gamma_i}{1 - |\beta_i|}.$$

Thus, for all $j \ge 0$, $q_{\delta+j} \in Q_s$. As in Theorem 1, if there is an infinite path in the automaton $\mathcal{A}$ starting in q_0 and labelled by

$$q_0 \xrightarrow{d_1/\varepsilon} q_1 \cdots \xrightarrow{d_\delta/\varepsilon} q_\delta \xrightarrow{d_{\delta+1}/a_1} q_{\delta+1} \xrightarrow{d_{\delta+2}/a_2} q_{\delta+2} \cdots$$

then $\sum_{j \ge 1} d_j\beta^{-j} = \sum_{j \ge 1} a_j\beta^{-j}$, with the digits a_j in A, and $\chi((d_j)_{j\ge1}) = (a_j)_{j\ge1}$. □

Corollary 1. *If β is a Pisot number, nonnormalized addition and multiplication by a fixed positive integer are computable by on-line finite state automata.*

6 Numeration Systems for the Integers

Let us first recall some definitions. Let $U = (u_n)_{n \geq 1}$ be a strictly increasing sequence of integers with $u_0 = 1$. Every positive integer s has a representation in the system U by the following greedy algorithm (see [8]): Let n such that $u_n \leq s < u_{n+1}$; let s_n be the quotient of the Euclidean division of s by u_n, and let r_n be the remainder: $s_n = q(s, u_n)$ and $r_n = r(s, u_n)$. Then iterate $s_k = q(r_{k+1}, u_k)$ and $r_k = r(r_{k+1}, u_k)$ for $n - 1 \leq k \leq 0$. Then $s = s_n u_n + \cdots + s_0 u_0$. The digits s_k are such that $0 \leq s_k < u_{k+1}/u_k$. The word $s_n \cdots s_0$ is the *normal* U_β-representation of s.

Let us denote by $d_\beta(1)$ the β-expansion of 1 computed by the greedy algorithm of Subsect. 2.1. If $d_\beta(1)$ is finite, *i.e.* $d_\beta(1) = a_1 \cdots a_m$, define a sequence $(t_k)_{k \geq 1}$ as $(t_k)_{k \geq 1} = (a_1 \cdots a_{m-1}(a_m - 1))^\omega$. If $d_\beta(1)$ is infinite take $(t_k)_{k \geq 1} = d_\beta(1)$. A sequence $U_\beta = (u_n)_{n \geq 0}$ can be canonically associated with β as follows. Let $u_0 = 1$ and for $n \geq 1$ let

$$u_n = t_1 u_{n-1} + \cdots + t_n u_0 + 1.$$

We then have the following result [4]: the finite factors of β-expansions of real numbers of $[0, 1]$ and the normal U_β-representations of positive integers are the same. In particular, normal U_β-representations of the integers are words on the alphabet $A = \{0, \ldots, \lfloor \beta \rfloor\}$. The system U_β is the *numeration system associated with* β.

Let $\chi : D^* \longrightarrow A^*$. The *prolongation* of χ is a function $\hat{\chi} : D^* 0^\omega \longrightarrow A^* 0^\omega$ defined by : let v and w be in D^* such that $\chi(v) = w$, then $\hat{\chi}(v0^\omega) = w0^\omega$. The function χ will be said to be continuous if its prolongation is continuous. By Theorem 1 and Proposition 2 we have the following result.

Corollary 2. *Let U_β be the numeration system associated with a number $\beta > 1$. Let D be a set of nonnegative digits containing A. Then there exists a digit set conversion $\chi : D^* \longrightarrow A^*$ in the system U_β which is on-line computable and continuous.*

Proof. The algorithm presented in Theorem 1 can be used again. Let $d_1 \cdots d_n \in D^*$ be an input word satisfying $N = d_1 u_{n-1} + \cdots + d_n u_0 < u_n$. Suppose that $n > \delta$. So the word $a_1 \cdots a_{n-\delta}$ is output, and the algorithm stops in a state q_n, the "value" of which is equal to $q_n = N - (a_1 u_{n-1} + \cdots + a_{n-\delta} u_{n-\delta})$. For every state q_n define $\omega(q_n)$ as the normal U_β-representation of q_n. Thus we get that $\chi(d_1 \cdots d_n) = a_1 \cdots a_{n-\delta} \omega(q_n) \in A^*$.

If $n \leq \delta$, then $\chi(d_1 \cdots d_n) = \omega(q_n)$, q_n being the state where the algorithm stops. □

If β is a Pisot number then $d_\beta(1)$ is eventually periodic [3]. In that case, the sequence U_β is linearly recurrent: if $d_\beta(1) = t_1 \cdots t_N (t_{N+1} \cdots t_{N+p})^\omega$, then $U_\beta = (u_n)_{n \geq 0}$ is defined by

$$u_{k+N+p} = t_1 u_{k+N+p-1} + \cdots + t_{N+p} u_k + u_{k+N} - t_1 u_{k+N-1} - \cdots - t_N u_k, \; k \geq 0$$

$$u_0 = 1,\ u_i = t_1 u_{i-1} + \cdots + t_i u_0 + 1,\ 1 \le i \le N + p - 1.$$

If $d_\beta(1)$ is finite, $d_\beta(1) = a_1 \cdots a_m$, define

$$u_{k+m} = a_1 u_{k+m-1} + \cdots + a_m u_k,\ k \ge 0$$

$$u_0 = 1,\ u_i = a_1 u_{i-1} + \cdots + a_i u_0 + 1,\ 1 \le i \le m - 1.$$

As a direct consequence of Theorem 2 we get

Corollary 3. *Let U_β be the linear numeration system associated with a Pisot number β. Let D be a set of nonnegative digits containing A. Then there exists a digit set conversion $\chi : D^* \longrightarrow A^*$ in the system U_β which is computable by an on-line finite state automaton with delay δ, where δ satisfies* (*).

Proof. We can use the construction given in Theorem 2. There is only need to define a terminal function ω such that, for $q \in Q$, $\omega(q(X))$ is equal to the β-expansion of $q(\beta)$. □

7 Examples

We illustrate the previous results on well-known Pisot numbers.

Example 1. Let β be an integer ≥ 2. On the alphabet $A = \{0, \cdots, \beta\}$, the representation of numbers is redundant. Addition on A is computable by an on-line finite state automaton, with delay 2.

For $\beta = 2$, this representation is the well-known carry-save representation. We give below the on-line finite automaton for addition in base $\beta = 2$. Let $\mathcal{A} = (Q, \{0, \ldots, 4\} \times (\{0, 1, 2\} \cup \varepsilon), E, q_0)$, with $Q = Q_t \cup Q_s$. All input words begin with 00. The set of transient states is $Q_t = \{q_0, q_1\}$. The set of synchronous states is

$$Q_s = \{q(X) \in \mathbb{Z}[X]/(X - 2) \mid 0 \le q(2) < 4\} = \{0, 1, 2, 3\}.$$

Transient edges are

$$q_0 \xrightarrow{0/\varepsilon} q_1 \xrightarrow{0/\varepsilon} 0.$$

The transition matrix of the synchronous part of $\mathcal{A}$ is given in the array below: the entry (i, j) contains the label of egdes from state i to state j.

	0	1	2	3
0	0/0, 4/1	1/0	2/0	3/0
1	2/1	3/1	0/0, 4/1	1/0
2	4/2, 0/1	1/1	2/1	3/1
3	2/2	3/2	4/2, 0/1	1/1

Example 2. Let $\beta = (1+\sqrt{5})/2$ be the golden ratio, the associated linear numeration system is the Fibonacci numeration system. Then $A = \{0, 1\}$. Formula (*) gives $\delta = 4$ for addition, with $D = \{0, 1, 2\}$. I have given in [10] an on-line finite automaton for addition with delay 3. The automaton constructed with delay 4 is not minimal in the number of states, but it is equivalent to the automaton with delay 3. In fact, the construction given in Theorem 1 and in Theorem 2 works with delay $\delta' = 3$, the bound on the states becoming

$$0 \leq q(\beta) \leq \mu$$

with here $\mu = \beta + 2$, and Condition (*) being replaced by

$$\beta\mu + d \leq \lfloor\beta\rfloor\beta^{\delta'} + \mu \qquad (**)$$

This gives the on-line finite automaton with delay 3 below.
Let $\mathcal{A} = (Q, \{0, 1, 2\} \times (\{0, 1\} \cup \varepsilon), E, \{q_0\})$. Input words begin with 00. The set of transient states is $\{q_0, q_1, q_2\}$. The elements of A are denoted by words of length 2 [1]. The transient part of $\mathcal{A}$ is of the form
$q_0 \xrightarrow{0/\varepsilon} q_1 \xrightarrow{0/\varepsilon} q_2$ and $q_2 \xrightarrow{0/\varepsilon} 00$; $q_2 \xrightarrow{1/\varepsilon} 01$; $q_2 \xrightarrow{2/\varepsilon} 02$.
In the synchronous part of $\mathcal{A}$ edges are the following ones:

	00	01	02	11	12	10	$\bar{1}2$	03	$1\bar{1}$	20
00	0/0	1/0	2/0							
01				1/0	2/0	0/0				
02	1/1	2/0								0/0
11	0/1	1/1	2/1							
12				1/1	2/1	0/1				
10				0/0	1/0		2/1			
$\bar{1}2$				2/0		1/0			0/0	
03				2/1		1/1			0/1	
$1\bar{1}$		0/0	1/0					2/0		
20		0/1	1/1					2/1		

Example 3. Let $\beta = (3+\sqrt{5})/2$. Here $A = \{0, 1, 2\}$. For addition the delay computed by (*) is 3, which is minimal.

References

1. A. Avizienis, Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on electronic computers* **10** (1961), 389–400.
2. D. Berend and Ch. Frougny, Computability by finite automata and Pisot bases. *Math. Systems Theory* **27** (1994), 274–282.

[1] We denote by $\bar{1}$ the signed digit -1.

3. A. Bertrand, Développements en base de Pisot et répartition modulo 1. *C.R.Acad. Sc.*, Paris **285** (1977), 419–421.
4. A. Bertrand-Mathis, Comment écrire les nombres entiers dans une base qui n'est pas entière. *Acta Math. Acad. Sci. Hungar.* **54** (1989), 237–241.
5. C.Y. Chow and J.E. Robertson, Logical design of a redundant binary adder. *Proc. 4th Symposium on Computer Arithmetic* (1978), 109–115.
6. S. Eilenberg, *Automata, Languages and Machines*, vol. A, Academic Press, 1974.
7. M.D. Ercegovac, On-line arithmetic: An overview. *Real time Signal Processing VII* **SPIE 495** (1984), 86–93.
8. A.S. Fraenkel, Systems of numeration. *Amer. Math. Monthly* **92(2)** (1985), 105–114.
9. Ch. Frougny, Representation of numbers and finite automata. *Math. Systems Theory* **25** (1992), 37–60.
10. Ch. Frougny, On-line finite automata for addition in some numeration systems, *RAIRO Theoretical Informatics and Applications*, to appear, available at `http://www.liafa.jussieu.fr/~cf`.
11. Ch. Frougny et J. Sakarovitch, Synchronisation déterministe des automates à délai borné. *Theoret. Comput. Sci.* **191** (1998), 61–77.
12. J.-M. Muller, Some characterizations of functions computable in on-line arithmetic. *I.E.E.E. Trans. on Computers*, **43** (1994), 752–755.
13. W. Parry, On the β-expansions of real numbers. *Acta Math. Acad. Sci. Hungar.* **11** (1960), 401–416.
14. A. Rényi, Representations for real numbers and their ergodic properties. *Acta Math. Acad. Sci. Hungar.* **8** (1957), 477–493.
15. K.S. Trivedi and M.D. Ercegovac, On-line algorithms for division and multiplication. *I.E.E.E. Trans. on Computers* **C 26** (1977), 681–687.